

SIMATIC NET

Промышленная связь с программаторами и компьютерами

Руководство

Предисловие, Содержание

Общие сведения

1

Промышленная связь

2

Основы OPC-интерфейса

3

OPC-переменные процесса для
SIMATIC NET

4

Свойства OPC-сервера событий
для SIMATIC NET

5

Использование OPC-сервера

6

Примеры

7

Часто задаваемые вопросы

8

Литература

9

Словарь

Это руководство является частью
комплектов документации с заказными
номерами:

6GK1971-1GA00-0AA1
6GK1971-5DA00-0AA1

11/2003
C79000-G8976-C172
Выпуск 03

Классификация указаний по безопасности

В настоящем руководстве содержатся указания, на которые следует обратить внимание в целях обеспечения собственной безопасности, равно как и безопасности и сохранности оборудования. Эти указания помечаются в руководстве с помощью предупреждающих знаков (треугольник с восклицательным знаком). По степени важности различают следующие указания:



Опасность

Указывает, что несоблюдение надлежащих мер предосторожности **приведет** к смерти, серьезной травме или значительному материальному ущербу.



Внимание

Указывает, что несоблюдение надлежащих мер предосторожности **может привести** к смерти, серьезной травме или значительному материальному ущербу.



Предостережение

Указывает, что несоблюдение надлежащих мер предосторожности может привести к незначительной травме или материальному ущербу.

Предостережение (без треугольника с воскл. знаком)

Указывает, что несоблюдение надлежащих мер предосторожности может привести к материальному ущербу.

Замечание

Указывает, что пренебрежение соответствующей информацией может привести к нежелательному результату или состоянию.

Примечание

Привлекает внимание читателя к особенно важной и полезной информации о продукте, обращении с ним или к определенной части документации.

Торговые знаки

SIMATIC®, SIMATIC HMI®, SIMATIC NET® - зарегистрированные торговые знаки фирмы SIEMENS AG.

Третьи лица, использующие в своих целях любые другие наименования, приводимые в настоящем документе и относящиеся к торговым знакам, могут быть привлечены к ответственности за нарушение прав владельцев торговых знаков.

Указания по технике безопасности для продукта

Прежде чем приступить к использованию продукта, внимательно прочитайте приведенные ниже указания по технике безопасности.

Квалифицированный персонал

К монтажу и эксплуатации данного оборудования может допускаться только **квалифицированный персонал**. В данном руководстве под квалифицированным персоналом понимаются лица, имеющие допуск к выполнению работ по вводу в эксплуатацию, заземлению и маркировке электрических цепей, устройств и систем в соответствии с установленными правилами и стандартами безопасности.

Правила использования аппаратных средств

Обратите внимание на следующее:



Внимание

Правила использования программных средств

Обратите внимание на следующее:



Внимание

Прежде чем приступить к работе

Прежде чем приступить к работе с продуктом, обратите внимание на следующее:

Предостережение

Перед тем как начать работу, ознакомьтесь с инструкциями в соответствующей документации последней версии. Данные для заказа документации можно найти в каталогах либо в региональном представительстве Siemens.

Авторские права © Siemens AG 2003. Все права защищены

Воспроизведение, передача или использование настоящего документа или его части допускается лишь с письменного разрешения. Нарушители будут привлекаться к ответственности за нанесенные убытки. Все права, включая права, вытекающие из патента или регистрации промышленной модели или разработки, защищены.

Отказ от ответственности

Содержание данного руководства было проверено на соответствие описанным в нем техническим и программным продуктам. Поскольку возможные изменения в последних невозможно предвидеть в полном объеме, полное соответствие не может быть гарантировано. Материалы данного руководства регулярно проверяются, а необходимые изменения вносятся в последующие выпуски. Мы рады любым предложениям по улучшению качества наших руководств.

Siemens AG
Automation and Drives
Industrial Communication
Postfach 4848, D-90327 Nürnberg

Siemens Aktiengesellschaft

©Siemens AG 2003
Технические данные могут быть изменены.

C79000-G8976-C172-03

Добро пожаловать в мир SIMATIC NET

SIMATIC NET – Ваш правильный выбор

Вам известны преимущества распределенных систем автоматизации, и вы хотите добиться оптимального использования промышленных коммуникаций. Вам требуется сильный партнер и надежные современные продукты. В таком случае SIMATIC NET – Ваш правильный выбор.

SIMATIC NET – Бесспорное лидерство успешных решений

Теперь, когда решение принято, вы можете положиться на нас. Эта документация будет вашим надежным спутником на пути к успешному применению SIMATIC NET. Она методично и планомерно ознакомит вас со всеми темами и покажет вам, как выполняется инсталляция компонентов, как конфигурируется проект и его отдельные компоненты, как создаются программы пользователя на основе OPC. Вы узнаете о возможностях, которые откроются перед вами, вашими приложениями и перед всей вашей компанией благодаря реализации промышленных коммуникаций на продуктах SIMATIC NET.

SIMATIC NET – Профессиональные решения не только для профессионалов

Не обязательно быть специалистом, чтобы успешно применять продукты SIMATIC NET. Эта документация даст вам нужные знания и поделится с вами "ноу-хау" и компетенцией специалистов.

Вы новичок? Вы можете систематично пополнять багаж своих знаний. Начните с введения в мир промышленных коммуникаций. Из него вы узнаете обо всем, что требуется знать о принципах коммуникаций, составе функций SIMATIC NET, технических возможностях и функциях, необходимых для решения ваших задач автоматизации. Познакомившись кратко с этой информацией, вы перейдете к вопросам инсталляции и отладки. Здесь, разумеется, все фокусируется на OPC-интерфейсе. Прочитайте общее введение в OPC-интерфейс, познакомьтесь с OPC-переменными и выясните, как добиться оптимальных результатов от OPC-сервера SIMATIC NET.

Вы профи? Вы можете совершенствовать свое мастерство. Подробнейшие описания, обширные примеры предоставят вам наикратчайший путь к выполнению инсталляции, конфигурирования и эксплуатации SIMATIC NET.

Вы считаете, что примеры полезны? Предоставленные примеры программ станут вашим базисом и лягут в основу ваших собственных идей.

Содержание

1	Общие сведения	21
1.1	Поиск нужной информации в документации	21
1.1.1	Где можно найти описание используемой продукции?	21
1.1.2	Что нового в документации на продукцию?	22
1.1.3	Какая версия документации является последней?	23
1.1.4	Как работать с документацией	23
1.2	Поддержка и обучение	24
1.3	Нормы и правила	25
1.3.1	Юридические нормы	25
1.3.2	Правила техники безопасности	26
2	Промышленная связь	27
2.1	Введение в промышленную связь	27
2.1.1	Специальные требования, предъявляемые к системам промышленной связи	28
2.1.2	Типы и категории сетей	28
2.1.3	Функции связи в системах автоматизации	29
2.1.4	Термины, используемые в проектировании промышленной связи	29
2.2	Место SIMATIC NET в проектировании систем промышленной связи	32
2.2.1	SIMATIC NET для систем промышленной связи	32
2.2.2	Коммуникационные системы SIMATIC NET	33
2.3	Промышленная связь через PROFIBUS	35
2.3.1	Описание PROFIBUS	35
2.3.2	Протоколы и устройства для SIMATIC NET PROFIBUS	35
2.3.3	Метод доступа к шине PROFIBUS	38
2.3.4	Место PROFIBUS в эталонной модели ISO/OSI	39
2.4	S5–совместимые коммуникации через FDL (протокол SEND/RECEIVE)	41
2.4.1	Типичная конфигурация системы для S5–совместимых коммуникаций	41
2.4.2	Принцип коммуникаций SEND/RECEIVE	42
2.4.3	Конфигурирование SEND/RECEIVE	42
2.4.4	Сервисы SEND/RECEIVE	43
2.5	Протокол DP	45
2.5.1	Протокол DP и его расширения	45
2.5.2	Типичная конфигурация системы DP	48
2.5.3	Принцип организации DP-коммуникаций	48
2.5.4	Конфигурирование системы DP	50
2.5.5	Ведущее устройство DP класса 1 – Принципы коммуникаций и сервисов	51
2.5.6	Ведомые устройства DP – Принципы коммуникаций и сервисов	53
2.5.7	Ведущее устройство DP класса 2 – Принципы коммуникаций и сервисов	53
2.5.8	DPC1 – Принципы коммуникаций и сервисов	55
2.5.9	DPC2 – Принципы коммуникаций и сервисов	57
2.6	Связь приводов с сервером шины PROFIDrive	58

2.7	Протокол FMS	59
2.7.1	Типичная конфигурация системы FMS	59
2.7.2	Принципы FMS-коммуникаций	60
2.7.3	Конфигурирование FMS	61
2.7.4	Сервисы FMS	63
2.7.5	Сервисы FMS для работы с переменными, ПК в качестве FMS-клиента	64
2.7.6	Сервисы FMS для работы с переменными, ПК в качестве FMS-сервера	65
2.8	Протокол S7 для PROFIBUS	66
2.8.1	Типичная конфигурация системы S7	67
2.8.2	Принципы S7-коммуникаций	67
2.8.3	Конфигурирование системы S7	69
2.8.4	S7-сервисы для работы с переменными	71
2.8.5	S7-сервисы для работы с буферами передачи/приема	72
2.8.6	S7-сервисы для управления блоками	73
2.8.7	S7-сервисы для обслуживания событий: тревоги	76
2.8.8	Как программировать тревоги	77
2.8.9	S7-сервисы для обеспечения безопасности	79
2.8.10	Информационные S7-сервисы	79
2.9	Промышленная связь через Industrial Ethernet	80
2.9.1	Описание Industrial Ethernet	80
2.9.2	Протоколы и устройства для SIMATIC NET Industrial Ethernet	81
2.9.3	Методы доступа к шине в Industrial Ethernet	82
2.9.4	Быстрая сеть Ethernet	83
2.9.5	Положение Industrial Ethernet в эталонной модели ISO/OSI	84
2.9.6	Транспортные протоколы для Industrial Ethernet	86
2.10	S5–совместимые коммуникации (SEND/RECEIVE)	87
2.10.1	Типичная конфигурация системы SEND/RECEIVE	87
2.10.2	Принципы коммуникаций SEND/RECEIVE	88
2.10.3	Конфигурация SEND/RECEIVE	90
2.10.4	Сервисы для работы с буферами SEND/RECEIVE	90
2.10.5	Сервисы для работы с переменными SEND/RECEIVE	91
2.11	Протокол S7 для Industrial Ethernet	93
2.11.1	Типичная конфигурация системы S7	93
2.11.2	Принципы S7-коммуникаций	94
2.12	Отказоустойчивые S7-соединения	96
2.12.1	Свойства отказоустойчивых соединений S7	96
2.12.2	Отказоустойчивые S7-соединения с двумя каналами связи	98
2.12.3	Отказоустойчивые S7-соединения с четырьмя каналами связи	99
2.12.4	Что следует помнить при конфигурировании отказоустойчивых соединений	100
2.13	Протокол SNMP	101
2.13.1	Типичная конфигурация системы для применения протокола SNMP	101
2.13.2	Принципы коммуникаций с использованием протокола SNMP	102
2.13.3	Конфигурирование	103
2.13.4	SNMP-прерывание	106
2.14	Сервисы для работы с буферами и OPC-интерфейс	107
2.14.1	Характеристики буфер-ориентированных коммуникаций	107
2.14.2	Назначение буферов данных переменным OPC	108

2.14.3	Применение буфер-ориентированных сервисов	109
2.14.4	Что требуется помнить при использовании буфер-ориентированных сервисов с собственным протоколом TCP/IP	110
3	Основы OPC-интерфейса	111
3.1	Основная терминология	111
3.1.1	Что такое объекты COM?	111
3.1.2	Как выглядят COM-объекты?	113
3.1.3	Что предоставляют COM-интерфейсы?	113
3.1.4	Какие типы интерфейсов существуют?	114
3.1.5	Как .NET-клиент может обратиться к COM-интерфейсу OPC?	115
3.1.6	Что означает управление объектами через интерфейс автоматизации?	116
3.2	Введение в XML и SOAP	116
3.2.1	Web-сервисы	118
3.3	Введение в OPC DX	119
3.4	Введение в OPC	120
3.4.1	Что предоставляет OPC-интерфейс?	121
3.4.2	Что такое OPC-сервер?	123
3.4.3	Что такое OPC-клиент?	124
3.4.4	Каким образом сервер и клиент взаимодействуют друг с другом?	124
3.5	OPC Data Access	125
3.5.1	Что предоставляет модель классов спецификации OPC Data Access?	126
3.5.2	Класс "OPC-сервер"	126
3.5.3	Класс "OPC-группа"	127
3.5.4	Класс "OPC-элемент"	127
3.5.5	Использование параметра "Percent Deadband"	128
3.5.6	Какие интерфейсы определены для OPC Data Access?	129
3.6	Интерфейс OPC DX	130
3.6.1	DX-сервер	130
3.6.2	База данных DX	131
3.7	OPC Alarms & Events	133
3.7.1	Что такое события и сообщения о событиях?	133
3.7.2	Что предоставляет модель классов OPC Alarms & Events?	134
3.7.3	Класс OPC Event Server	135
3.7.4	Класс OPC Event Subscription	135
3.7.5	Класс OPC Event Area Browser	137
3.7.6	Как выполняется прием сообщений?	137
3.7.7	Аварийные сообщения в SIMATIC S7	137
3.7.8	Примеры работы с аварийными событиями	139
3.7.9	Какие интерфейсы используются для Alarms & Events?	140
3.8	Интерфейс OPC XML	140
3.8.1	Web-сервис для OPC XML	142
3.8.2	Простые сервисы для чтения/записи	142
3.9	OPC в SIMATIC NET	144
3.9.1	Что предоставляет OPC-сервер SIMATIC NET?	144
3.9.2	Как применяется OPC-сервер SIMATIC NET в мире автоматизации?	146
3.9.3	Какие преимущества предоставляет семейству SIMATIC NET OPC-сервер?	147

3.9.4	Каков наилучший способ обращения к данным процесса?	148
3.9.5	Использование операций над группами	148
3.9.6	Примеры операций над группами	149
3.9.7	Обращение к OPC-кэшу	149
3.9.8	Примеры сервисов, которые могут использовать кэш	149
3.9.9	Структурирование элементов	150
3.9.10	Пример структурирования элементов	151
3.9.11	Применение сервисов для передачи/приема буферов	151
3.9.12	Пример использования сервисов для передачи/приема буферов	152
3.9.13	Использование соответствующих OPC-методов	152
4	OPC-переменные процесса для SIMATIC NET	157
4.1	Поддерживаемые функции связи	157
4.2	Понятие переменной процесса	158
4.3	Синтаксис идентификаторов (ItemID) переменных процесса	158
4.4	PROFIBUS-DP	159
4.4.1	Идентификатор протокола	160
4.4.2	Имена соединений	160
4.4.3	Высокопроизводительный внутрипроцессный сервер (SIMATIC NET Inproc-Server) для протокола PROFIBUS-DP	161
4.4.4	Мощный OPC-сервер SIMATIC NET для протокола PROFIBUS-DP	162
4.4.5	Сервисы DPC1 и DPC2	164
4.4.6	Переменные процесса для сервисов ведущего устройства класса 1	165
4.4.7	Синтаксис переменных процесса для ведущего устройства класса 1	166
4.4.8	Примеры переменных процесса для ведущего устройства класса 1	168
4.4.9	Сервисы DPC1	169
4.4.10	Синтаксис переменных процесса для сервисов DPC1	169
4.4.11	Примеры переменных процесса для сервисов DPC1	172
4.4.12	Быстрая логика (Fast Logic) для CP 5613 и CP 5614 (только для режима ведущего устройства)	172
4.4.13	Синтаксис управляющих переменных для Fast Logic	173
4.4.14	Специальные информационные переменные протокола DP	175
4.4.15	Синтаксис специальных информационных переменных DP	176
4.4.16	PROFIDrive	181
4.4.17	Protocol ID	181
4.4.18	Имя устройства (Device Name)	181
4.4.19	Обращение к узлу шины (к ведомому устройству)	182
4.4.20	Синтаксис специальных информационных системных переменных	189
4.4.21	Сервисы ведущего устройства DP класса 2	189
4.4.22	Синтаксис сервисов ведущего устройства DP класса 2	190
4.4.23	Имя коммуникационного процессора (Name of the CP)	190
4.4.24	Обращение к узлу шины (к ведущему устройству DP класса 1)	191
4.4.25	Обращение к узлу шины (к ведомому устройству)	194
4.4.26	Элементы для коммуникационного процессора (CP)	201
4.4.27	Примеры специальных информационных DP-переменных	201
4.4.28	Сервисы обслуживания переменных для обращения к локальным данным ведомого устройства	202
4.4.29	Синтаксис переменных процесса для ведомого устройства DP	203
4.4.30	Примеры переменных процесса для ведомого устройства DP	205

4.4.31	Специальные информационные переменные ведомого устройства DP	206
4.4.32	Синтаксис специальных информационных переменных ведомого устройства DP	206
4.5	S7-соединения	207
4.5.1	Мощный OPC-сервер SIMATIC NET для протокола S7	208
4.5.2	Идентификатор протокола	209
4.5.3	Имена соединений	209
4.5.4	Сервисы для работы с переменными	209
4.5.5	Синтаксис переменных процесса для сервисов обслуживания переменных S7	210
4.5.6	Примеры переменных процесса для сервисов обслуживания переменных S7	213
4.5.7	Буфер-ориентированные сервисы	214
4.5.8	Синтаксис переменных процесса для буфер-ориентированных сервисов	214
4.5.9	Примеры переменных процесса для буфер-ориентированных сервисов	216
4.5.10	Специальные информационные переменные протокола S7	217
4.5.11	Синтаксис специальных информационных переменных протокола S7	218
4.5.12	Примеры специальных информационных переменных протокола S7 и возвращаемых значений	220
4.5.13	Сервисы для управления блоками	222
4.5.14	Синтаксис управляющих переменных для сервисов для управления блоками	222
4.5.15	Примеры применения сервисов для управления блоками	226
4.5.16	Пароли	228
4.5.17	Синтаксис управляющих переменных для паролей	229
4.5.18	Пример использования паролей	230
4.5.19	Несконфигурированное S7-соединение	231
4.6	PROFIBUS FMS	236
4.6.1	Идентификатор протокола (FMS)	236
4.6.2	Имена соединений	236
4.6.3	Типы данных	237
4.6.4	Сервисы для работы с переменными	239
4.6.5	Синтаксис переменных процесса для сервисов обслуживания переменных FMS	239
4.6.6	Примеры переменных процесса для сервисов обслуживания FMS-переменных	240
4.6.7	Специальные информационные переменные протокола FMS	241
4.6.8	Синтаксис специальных информационных переменных протокола FMS	241
4.6.9	Примеры специальных информационных переменных протокола FMS и значений, возвращаемых в этих переменных	245
4.6.10	Переменные FMS-сервера	246
4.6.11	Переменные FMS-сервера: ItemID	246
4.7	S5-совместимые коммуникации	247
4.7.1	S5-совместимые коммуникации по сети Industrial Ethernet	247
4.7.2	Мощный OPC-сервер семейства SIMATIC NET для протокола SR	248
4.7.3	Идентификатор протокола (Protocol ID)	249
4.7.4	Имена соединений	250
4.7.5	Сервисы для работы с переменными	250

4.7.6	Сервисы для работы с буферами	253
4.7.7	Специальные информационные переменные для сервисов SEND/RECEIVE	257
4.7.8	S5–совместимые коммуникации по сети PROFIBUS	259
4.7.9	Идентификатор протокола	259
4.7.10	Имена соединений	260
4.7.11	Сервисы для работы с буферами	260
4.7.12	Специальные информационные переменные для FDL	266
4.8	SNMP-коммуникации по сети Industrial Ethernet	269
4.8.1	Идентификатор протокола	270
4.8.2	Типы данных, поддерживаемые протоколом SNMP	270
4.8.3	Переменные процесса для сервисов обслуживания переменных протокола SNMP	271
4.8.4	Специальные информационные переменные протокола SNMP	272
4.8.5	Специальные прерывания протокола SNMP	275
4.9	Ограничение доступа к OPC-переменным	276
6	Использование OPC-сервера	295
6.1	SIMATIC Computing	295
6.1.1	Что такое элементы ActiveX Control?	296
6.1.2	Как управляющие элементы обращаются к данным процесса?	298
6.1.3	Как осуществляется доступ с помощью DCOM?	299
6.1.4	Как используются управляющие элементы SIMATIC NET в инструментах разработки?	300
6.1.5	Конфигурирование элемента управления данными SIMATIC NET OPC Data control	301
6.1.6	Как открыть окно свойств для элемента SIMATIC NET OPC Data control?	302
6.1.7	Выбор OPC-сервера	302
6.1.8	Подключение переменных процесса к элементам визуализации	303
6.1.9	Назначение события переменной процесса	305
6.1.10	Дополнительные задачи	306
6.1.11	Конфигурирование элемента "Кнопка"	308
6.1.12	Свойства элемента "Кнопка"	309
6.1.13	Вызов окна свойств для элемента "Кнопка"	310
6.1.14	Выбор надписи	310
6.1.15	Выбор шрифта надписи	310
6.1.16	Выбор цвета для элемента "Кнопка"	311
6.1.17	Активизация элемента "Кнопка"	311
6.1.18	Назначение элемента "Кнопка" переменной процесса	312
6.1.19	Конфигурирование элемента "Число"	313
6.1.20	Свойства и методы элемента "Число"	313
6.1.21	Вызов окна свойств для элемента "Число"	318
6.1.22	Настройка отображения данных	318
6.1.23	Выбор стиля отображения	320
6.1.24	Выбор надписи	320
6.1.25	Настройка масштаба для отображения значений	322
6.1.26	Выбор шрифта надписи	323
6.1.27	Выбор цвета для элемента "Число"	323
6.1.28	Активизация элемента "Число"	324

6.1.29	Назначение переменной процесса элементу "Число"	324
6.1.30	Конфигурирование элемента "Ползунок"	325
6.1.31	Свойства элемента "Ползунок"	325
6.1.32	Вызов окна свойств для элемента "Ползунок"	327
6.1.33	Настройка отображения данных	327
6.1.34	Активизация элемента "Ползунок"	328
6.1.35	Назначение переменной процесса элементу "Ползунок"	329
6.1.36	Использование других элементов ActiveX	329
6.1.37	Правила работы с контейнерными приложениями	330
6.1.38	Правила работы с элементами ActiveX	330
6.1.39	Каким образом элемент Data Control читает данные?	331
6.1.40	Как данные записываются в элемент Data Control?	331
6.1.41	Свойства и методы управляющих элементов SIMATIC	332
6.1.42	Activated (Активизация)	335
6.1.43	Alignment (Выравнивание)	336
6.1.44	Appearance (Стиль отображения)	337
6.1.45	AutoConnect (Автоматическое соединение)	337
6.1.46	AutoConnectTimeout (Превышение времени при автоматическом соединении)	338
6.1.47	BackColor (Цвет заднего фона)	339
6.1.48	BorderStyle (Стиль границы)	339
6.1.49	Caption <i>n</i> (Надпись <i>n</i>)	340
6.1.50	Caption <i>n</i> Alignment (Выравнивание надписи <i>n</i>)	340
6.1.51	Caption <i>n</i> BackColor (Цвет заднего фона надписи <i>n</i>)	341
6.1.52	Caption <i>n</i> Font (Шрифт надписи <i>n</i>)	342
6.1.53	Caption <i>n</i> ForeColor (Цвет надписи <i>n</i>)	342
6.1.54	Caption <i>n</i> Size (Размер текста <i>n</i>)	343
6.1.55	ConvertedValue (Преобразованное значение)	344
6.1.56	DataType (Тип данных)	345
6.1.57	DefaultDeadband (Зона нечувствительности по умолчанию)	346
6.1.58	DefaultUpdateRate (Скорость обновления по умолчанию)	347
6.1.59	Direction (Положение)	348
6.1.60	Enabled (Разрешено)	348
6.1.61	Factor (Коэффициент)	349
6.1.62	FalseCaption (Текст в состоянии "Ложь")	350
6.1.63	FalseColor (Цвет в состоянии "Ложь")	350
6.1.64	Font (Шрифт)	351
6.1.65	ForeColor (Цвет надписи)	351
6.1.66	LargeChange (Шаг грубой настройки)	352
6.1.67	LayoutType (Расположение)	353
6.1.68	LimitCheck (Проверка граничных значений)	353
6.1.69	Locked (Только чтение)	354
6.1.70	LowerLimit (Нижняя граница)	355
6.1.71	Max (Максимальное значение)	355
6.1.72	Min (Минимальное значение)	356
6.1.73	NodeName (Имя узла)	356
6.1.74	Offset (Смещение)	357
6.1.75	Precision (Точность)	358
6.1.76	Radix (Представление)	358
6.1.77	RangeRawMax (Максимальное входное значение)	359
6.1.78	RangeRawMin (Минимальное входное значение)	360
6.1.79	RangeScaledMax (Максимальное отображаемое значение)	362
6.1.80	RangeScaledMin (Минимальное отображаемое значение)	363

6.1.81	ScaleMode (Режим масштабирования)	364
6.1.82	ServerName (Имя сервера)	365
6.1.83	ShowButtons (Показывать кнопки)	366
6.1.84	ShowErrorBoxes (Показывать сообщения об ошибках)	367
6.1.85	SmallChange (Шаг точной настройки)	368
6.1.86	Ticks (Метки шкалы)	368
6.1.87	TrueCaption (Надпись в состоянии "верно")	369
6.1.88	TrueColor (Цвет в состоянии "верно")	369
6.1.89	UpperLimit (Верхняя граница)	370
6.1.90	Value (Значение)	370
6.1.91	WriteMode (Режим записи)	371
6.1.92	ZeroPad (Заполнение нулями)	372
6.1.93	AddNotification (Добавить уведомление)	372
6.1.94	Connect (Соединить)	373
6.1.95	Disconnect (Отсоединить)	374
6.1.96	GetItemProperties (Прочитать свойства элемента)	374
6.1.97	QueryAvailableItemProperties (Опросить имеющиеся свойства элемента)	375
6.1.98	ReadMultiVariables (Прочитать нескольких переменных)	376
6.1.99	ReadVariable (Прочитать переменную)	377
6.1.100	Write (Записать)	378
6.1.101	WriteMultiVariables (Записать несколько переменных)	378
6.1.102	WriteVariable (Записать переменную)	379
6.1.103	События элементов семейства SIMATIC	379
6.1.104	События элемента Data Control	380
6.1.105	События элемента "Кнопка"	380
6.1.106	События элемента "Число"	380
6.1.107	События элемента "Ползунок"	381
6.1.108	Сообщения об ошибках, предусмотренные для элементов визуализации SIMATIC	381
6.1.109	Сообщения об ошибках элемента Data Control	382
6.1.110	Сообщения об ошибках элемента "Число"	382
6.2	Программирование интерфейса автоматизации	384
6.2.1	Программирование интерфейса автоматизации для доступа к данным	384
6.2.2	Что предоставляет пользователю объектная модель OPC Data Access?	384
6.2.3	Что требуется учитывать при программировании	385
6.2.4	Объекты интерфейса автоматизации для доступа к данным	386
6.2.5	Программирование интерфейса автоматизации для доступа к аварийным сообщениям и событиям	397
6.2.6	Что предоставляет объектная модель OPC Alarms & Events?	397
6.2.7	О чем следует помнить при программировании	399
6.2.8	Объекты интерфейса автоматизации для доступа к аварийным сообщениям и событиям	399
6.3	Программирование пользовательского интерфейса	410
6.3.1	Создание COM-объектов и чтение статуса OPC-сервера	412
6.3.2	Объекты пользовательского интерфейса для доступа к данным	412
6.3.3	Объект OPCServer	412
6.3.4	Объект OPCGroup	417
6.3.5	Объекты пользовательского интерфейса для доступа к аварийным сообщениям и событиям	424

6.3.6	Объект OPCEventServer	424
6.3.7	Объект OPCEventSubscription	428
6.3.8	Объект OPCEventAreaBrowser	430
6.3.9	Интерфейсы клиента для доступа к аварийным сообщениям и событиям	430
6.3.10	Интерфейс IOPCEventSink	431
6.3.11	Интерфейс IOPCShutdown	433
6.3.12	Сообщения об ошибках, предусмотренные для переменных процесса интерфейса OPC DA	433
6.4	Программирование DX-интерфейса	437
6.4.1	Структура базы данных DX	437
6.4.2	Сервисы конфигурирования DX	441
6.5	Программирование XML-интерфейса	444
6.5.1	Описание элементов	444
6.5.2	Основные схемы	445
6.5.3	ItemProperty (Свойство элемента)	446
6.5.4	ItemValue (Значение элемента)	447
6.5.5	OPCError (Ошибка OPC)	448
6.5.6	ReplyBase (Основная информация об ответе)	449
6.5.7	RequestOptions (Опции запроса)	450
6.5.8	Read (Чтение)	451
6.5.9	ReadResponse (Ответ на задание на чтение)	454
6.5.10	Write (Запись)	456
6.5.11	WriteResponse (Ответ на задание на запись)	458
6.5.12	Применение "подписок"	460
6.5.13	Subscribe (Задание на подписку)	462
6.5.14	SubscribeResponse (Ответ на запрос о подписке)	464
6.5.15	SubscriptionPolledRefresh (Обновление элементов подписки)	466
6.5.16	SubscriptionPolledRefreshResponse (Ответ на запрос SubscriptionPolledRefresh)	467
6.5.17	SubscriptionCancel (Отмена подписки)	469
6.5.18	SubscriptionCancelResponse (Ответ на запрос SubscriptionCancel)	469
6.5.19	Browse (Обзор)	470
6.5.20	BrowseResponse (Ответ на запрос Browse)	472
6.5.21	GetProperties (Прочитать свойства)	474
6.5.22	GetPropertiesResponse (Ответ на запрос GetProperties)	476
6.5.23	GetStatus (Прочитать статус)	477
6.5.24	GetStatusResponse (Ответ на запрос GetStatus)	479
Словарь		683

1 Общие сведения

В этой главе будет рассказано о содержании настоящей документации, об ее изменениях и версиях, а также будет пояснено, как ею пользоваться.

Руководствуясь этой Главой, вы сможете быстро и легко находить информацию по нужной вам теме.

В данной главе также поясняется, как можно получить помощь, если у вас имеются технические вопросы.

В конце главы приведены юридические нормы и нормы безопасности, которые распространяются на изделия SIMATIC NET.

1.1 Поиск нужной информации в документации

1.1.1 Где можно найти описание используемой продукции?

В настоящей документации описана следующая продукция:

- DP-5613/Windows V 6.0 и выше
- S7-5613/Windows V 6.0 и выше
- FMS-5613/Windows V 6.0 и выше
- SOFTNET-DP/Windows V 6.0 и выше
- SOFTNET-S7/Windows V 6.0 и выше
- SOFTNET-DP-Slave/Windows V 6.0 и выше
- CP 5613/CP 5614 Software V 6.0 и выше
- S7-1613/Windows V 6.0 и выше
- SOFTNET-S7/Windows V 6.0 и выше
- S7-REDCONNECT/Windows V 6.0 и выше
- PN OPC Server/Windows V 6.0 и выше
- SNMP OPC Server/Windows V 6.0 и выше
- DX OPC Server/Windows V6.2 и выше

1.1.2 Что нового в документации на продукцию?

Ранее документация по SIMATIC NET состояла из отдельных руководств. Чтобы предоставить вам информацию более компактно и облегчить ее использование, мы создали настоящую документацию, объединив в нее несколько руководств. В следующей таблице показано, в каком месте настоящей документации можно найти информацию по темам, которые ранее раскрывались в отдельных руководствах:

Темы предыдущих руководств	Где ее можно найти сейчас	
Интерфейс программирования S7	OPC Process Variables for SIMATIC NET - "S7 Communication"	OPC-переменные процесса для SIMATIC NET - "S7-коммуникации"
Интерфейс программирования SEND/RECEIVE	OPC Process Variables for SIMATIC NET - "S5-compatible Communication over Industrial Ethernet"	OPC-переменные процесса для SIMATIC NET - "S5-совместимые коммуникации через Industrial Ethernet"
Введение в SOFTNET для Industrial Ethernet	"Industrial Communication with Industrial Ethernet"	"Промышленные коммуникации через Industrial Ethernet"
Интерфейс программирования DP	OPC Process Variables for SIMATIC NET - "PROFIBUS -DP"	OPC-переменные процесса для SIMATIC NET - "PROFIBUS -DP"
Интерфейс программирования DPC1	OPC Process Variables for SIMATIC NET - "DPC1 Services"	OPC-переменные процесса для SIMATIC NET - "Сервисы DPC1"
Ведомое устройство DP	"Variable Services for Access to Local Slave Data"	"Сервисы обслуживания переменных для обращения к данным локальных ведомых устройств"
Интерфейс программирования FDL	OPC Process Variables for SIMATIC NET - "S5-compatible Communication over PROFIBUS"	OPC-переменные процесса для SIMATIC NET - "S5-совместимые коммуникации через PROFIBUS"
Интерфейс программирования FMS	OPC Process Variables for SIMATIC NET - "PROFIBUS -FMS"	OPC-переменные процесса для SIMATIC NET - "PROFIBUS -FMS"
COM PROFIBUS	"NCM S7 for PROFIBUS" Manual	Руководство "NCM S7 для PROFIBUS"
COML S7	COML S7 больше не требуется и документация по нему в настоящее время заменена следующей документацией:	
	"NCM S7 for PROFIBUS" или "NCM S7 for Industrial Ethernet" Manual	Руководство "NCM S7 для PROFIBUS" или "NCM S7 для Industrial Ethernet"
Введение в OPC- Сервер	Многие разделы настоящего документа	

1.1.3 Какая версия документации является последней?

В следующей таблице приведен перечень версий настоящей документации:

Версия	Соответствующая версия программного обеспечения	Содержание/Изменения
Декабрь 2003	SIMATIC NET V6.2 OPC DX V1.0 OPC XML V1.0 OPC A&E V1.1	Программное обеспечение SIMATIC NET для промышленных коммуникаций через OPC

1.1.4 Как работать с документацией

Чтобы облегчить работу с документацией, в ней используются различные условные обозначения и символы. С их помощью разграничивается информация различного назначения, а также ускоряется поиск требуемых сведений.

Условные обозначения

В данной документации для разграничения информации различного назначения используется ряд условных обозначений, который позволяет визуально находить информацию определенного типа. В следующей таблице перечислены используемые условные обозначения.

Условное обозначение	Означает:
Шрифт в стиле "печатной машинки"	Программный код.
Курсив	Переменные параметры, которые должны быть заменены текущими значениями.
Жирный	В описаниях синтаксиса: фиксированные элементы синтаксиса. В обычном тексте: выделенные слова позволяют быстро найти нужную информацию.
{ }	В фигурные скобки заключена необязательная информация в описаниях синтаксиса.
	Означает предупреждение.

1.2 Поддержка и обучение

Поддержка

Если у вас есть какие-либо вопросы, касающиеся продукции SIMATIC NET, обращайтесь, пожалуйста, в региональное представительство Siemens.

Необходимые адреса можно найти:

- В нашем каталоге IK PI, который имеется на компакт-диске с документацией по SIMATIC NET "Catalog IK PI Industrial Communication and Field Devices" (дважды щелкните по файлу **START.HTM** в главной папке компакт-диска с документацией по SIMATIC NET -> **English** -> окно выбора под "Industrial Communication SIMATIC NET" -> **Catalogs** -> **Catalog IK PI Industrial Communication and Field Devices**).
- В Интернете
<http://www.ad.siemens.de>
- В интерактивном каталоге CA01
<http://www.siemens.de/automation/ca01>
- На компакт-диске Quick Start

Центр обучения

Познакомиться ближе с системой автоматизации SIMATIC S7 и программируемыми контроллерами можно, пройдя обучение на наших курсах. Обращайтесь, пожалуйста, в свой региональный учебный центр или центральный учебный центр по адресу:

D 90327 Нюрнберг.

Информационная линия по вопросам проведения курсов:

Телефон: 01805 / 23 56 11 (0.12 € в минуту в Германии)

Факс: +49 / (0)1805 / 23 5612

Интернет: <http://www.siemens.de/sitrain>

E-mail: info@sitrain.com

Служба технической поддержки A&D

Информацию о службе технической поддержки A&D можно найти в конце настоящего руководства.

1.3 Нормы и правила

Используя продукцию SIMATIC NET, вы обязаны руководствоваться следующими нормами и правилами:

1.3.1 Юридические нормы

Мы особо подчеркиваем, что содержание настоящего документа не может стать частью или изменить любое предшествующее или существующее соглашение, обязательство или правоотношения. Полный исключительный список обязательств Siemens содержится в Соглашении о покупке. Любые утверждения, содержащиеся в настоящем документе, не создают новых обязательств и не ограничивают уже существующие обязательства.

Кроме того, мы подчеркиваем, что в настоящем документе во избежание его громоздкости не могли быть рассмотрены абсолютно все возможные проблемы, которые могут возникнуть при использовании OPC Server. В случае если вам требуется дополнительная информация или в случае возникновения проблемы, которая недостаточно подробно рассмотрена в настоящем документе, обращайтесь, пожалуйста, в региональное представительство фирмы Siemens.

Отказ от ответственности

Мы тщательно проверили содержание настоящего документа на соответствие описанным в нем аппаратным и программным средствам. Поскольку изменения в последних не могут быть предвидены заранее, полное соответствие не гарантируется. Однако информация, содержащаяся в настоящем документе, непрерывно пересматривается, и все необходимые изменения вносятся в последующие выпуски. Мы с радостью рассмотрим любые предложения по улучшению качества документации.

Технические данные могут быть изменены.

Воспроизведение, передача или использование настоящего документа и его содержания не допускается без письменного разрешения. Нарушители будут привлечены к ответственности за нанесенный ущерб. Все права, включая права, вытекающие из выдачи патента или регистрации модели или конструкции, защищены.

Siemens AG
Bereich Automatisierungs- und Antriebstechnik
Geschäftsgebiet Industrie-Automatisierung
Postfach 4848, D- 90327 Nürnberg
© Siemens AG 2003

1.3.2 Правила техники безопасности

Квалифицированный персонал

К установке и работе с настоящим продуктом может быть допущен только квалифицированный персонал. Под квалифицированным персоналом понимаются лица, имеющие допуск к выполнению работ по вводу в эксплуатацию, заземлению и маркировке электрических цепей, устройств и систем в соответствии с установленными правилами и стандартами безопасности.

Правильное использование аппаратных средств



Внимание

Данное устройство и его компоненты могут применяться только для целей, предусмотренных в соответствующем каталоге или техническом описании. Совместное использование с устройствами или компонентами других производителей допускается только в отношении продуктов, одобренных и рекомендованных фирмой Siemens.

Правильное и безопасное функционирование данного продукта может гарантироваться лишь при условии соблюдения требований к транспортировке, хранению, установке и монтажу, а также при соблюдении рекомендаций по эксплуатации и техническому обслуживанию.

Продукты, описанные в настоящем документе, облегчают доступ к данным процесса и изменение данных процесса. Изменение данных процесса может привести к непредсказуемой реакции процесса, что может явиться причиной нанесения серьезных травм персоналу, в том числе со смертельным исходом, а также повреждения оборудования.

Выполняйте операции с надлежащей осторожностью и не обращайтесь к данным, которые могут вызвать непредсказуемую реакцию со стороны управляемых устройств и оборудования.

2 Промышленная связь

Вы хотите узнать принципы организации связи и познакомиться с набором функций отдельных протоколов, которые используются для организации промышленной связи в SIMATIC NET. Вам интересно узнать о существующих технических возможностях. Прочитав этот раздел до конца, вы будете готовы выбрать функции, наиболее подходящие для решения ваших задач автоматизации.

Интересующие вас вопросы	описаны в Разделе
Вы хотите знать, что понимается под промышленной связью	Введение в промышленную связь
Вы хотите получить общее представление о том, какое место SIMATIC NET занимает в единой концепции промышленной связи фирмы Siemens.	SIMATIC NET для промышленной связи
Вы хотите познакомиться с набором функций различных протоколов для промышленной связи с применением PROFIBUS.	Промышленная связь через PROFIBUS
Вы хотите познакомиться с набором функций различных протоколов для промышленной связи с применением Industrial Ethernet.	Промышленная связь через Industrial Ethernet
Вам нужна информация по использованию сервисов, ориентированных на работу с буферами, через интерфейс OPC.	Буфер-ориентированные сервисы, применяемые в OPC-интерфейсе

2.1 Введение в промышленную связь

Введение

Коммуникационные сети являются основой современных решений в области автоматизации. Благодаря им становятся возможными взаимосвязь и обмен данными между отдельными компонентами и устройствами системы автоматизации.

В данной главе содержатся общие сведения о технологиях и проектировании в области промышленной связи.

2.1.1 Специальные требования, предъявляемые к системам промышленной связи

Промышленные сети

Промышленные сети должны удовлетворять специальным требованиям, которые по степени строгости могут быть как выше, так и ниже требований, предъявляемых к обычным сетям. Как показывает заключительный анализ, успех или неуспех производственного предприятия зависит от функциональности и эффективности системы автоматизации.

- Система промышленной связи должна обеспечивать подключение простейших датчиков и исполнительных механизмов к контроллерам, а также подключение контроллеров друг к другу и к компьютерам.
- Чтобы избежать проблем в протекании технологических процессов и работе цехов, необходимо обеспечить оперативную доставку информации: необходимая информация должна быть там, где она нужна, в нужное время.
- Ряд таких общих для условий промышленного производства факторов, как электромагнитные помехи, высокая степень загрязненности, влажность и механические нагрузки предъявляют повышенные требования к структуре сети и ее компонентам.
- Особенно важным требованием является достоверность данных. Возмущающие факторы, воздействующие на передатчик данных или среду передачи, должны выявляться и сигнализироваться, чтобы пользователь мог полагаться на достоверность и корректность информации.
- Редко одна система автоматизации похожа на другую. Система промышленной связи должна быть гибкой, обеспечивая возможность адаптации к требованиям конкретной задачи.

2.1.2 Типы и категории сетей

Классификация сетей

Чтобы разделить промышленные коммуникационные сети (по-другому, сети промышленной связи) на различные категории, удобно ввести грубую классификацию, основанную на протяженности сетей. Сети можно разбить на следующие группы:

- Всемирная сеть (GAN): сети, охватывающие весь мир, например, Интернет.
- Территориальная сеть (WAN): сети, охватывающие регионы и страны, например, сеть Datex-P почтовой службы Германии или сети Intranet больших предприятий.
- Локальная сеть (LAN): связь в пределах ограниченной территории, например, внутри одного здания. Локальная сеть характеризуется, в общем случае, однородной структурой передачи данных.

К какой категории относятся промышленные сети?

Сети промышленной связи относятся к группе локальных сетей, хотя они допускают и межсетевой обмен (например, через Интернет) и передачу выбранной информации из производственного участка в любую точку мира.

2.1.3 Функции связи в системах автоматизации

Системы промышленной связи могут выполнять различные функции:

- Связь с процессом или полевым уровнем: непосредственное подключение исполнительных механизмов и датчиков к программируемым логическим контроллерам. Обмен данными может быть как циклическим, с использованием образа процесса, так и ациклическим, с применением команд управления.
- Обмен данными: обмен данными между программируемыми контроллерами или между программируемым контроллером и интеллектуальным партнером (например, ПК) с применением протоколов и интерфейсов, используемых в промышленных условиях.
- IT-коммуникации: программируемые контроллеры могут быть интегрированы в мир современных информационных технологий, благодаря чему возможен обмен данными по всему миру с применением таких стандартных инструментов, как электронная почта или веб-браузеры.

2.1.4 Термины, используемые в проектировании промышленной связи

Введение

В этом разделе поясняются некоторые термины, широко применяемые в проектировании систем связи. Этот раздел можно пропустить и обратиться к нему позже, когда соответствующий термин встретится в тексте.

Термины и пояснения

Протокол

Протокол – это набор правил, согласно которым происходит управление передачей данных. Протоколами, например, описывается структура данных, структура пакетов данных и кодировка.

Протоколами также могут описываться механизмы управления, а также требования, предъявляемые к программным и аппаратным средствам.

Соединение

Соединение – это виртуальный канал между двумя коммуникационными партнерами. Сеанс связи с использованием протоколов, ориентированных на соединения, предполагает следующее:

- Установление соединения
- Обмен данными
- Завершение соединения

Связь, ориентированную на установление соединения, можно сравнить с разговором по телефону, когда после набора номера абонента устанавливается соединение, затем происходит обмен данными (разговор), после чего соединение вновь разрывается (абоненты кладут трубки).

Архитектура "клиент/сервер"

Архитектура "клиент/сервер" позволяет нескольким пользователям совместно использовать ресурсы друг друга. Сервер предоставляет свои сервисы клиентам. Клиенты используют сервисы, предоставляемые сервером, отправляя ему задания. Сервер обрабатывает задания и отправляет клиентам результаты. В общем случае, сервер может обслуживать несколько клиентов одновременно. Интерфейс OPC базируется на архитектуре "клиент/сервер".

Принцип "ведущий/ведомый"

Если связь строится по принципу "ведущий/ведомый", в этом случае имеется одна станция, являющаяся ведущей, которая может инициировать обмен данными с ведомыми устройствами по своей инициативе. После этого ведомые устройства отвечают ведущему устройству. Ведомое устройство в своем ответе может передать данные. В отличие от ведущего устройства, ведомое устройство никогда не может активизировать связь по своей инициативе. На принципе "ведущий/ведомый" основан, например, протокол PROFIBUS-DP.

Синхронный/асинхронный обмен (связь)

Понятия "синхронный/асинхронный" описывают принцип выполнения сервиса. Если сервис выполняется синхронно, управление программой возвращается инициатору вызова только после возвращения результатов выполнения сервиса.

Если сервис выполняется асинхронно, управление программой возвращается инициатору вызова сразу же после запроса сервиса. Результат сервиса возвращается инициатору вызова в неопределенное время, другими словами, асинхронно.

Эталонная модель ISO/OSI

Семиуровневая модель "Open Systems Interconnection" (OSI) – это эталонная модель, описывающая передачу данных в сетях, которая получила свое название в честь международной организации по стандартизации (ISO). Модель состоит из семи уровней: два физических (аппаратных) уровня (уровни 1 и 2), два уровня, ориентированных на передачу данных (уровни 3 и 4), и три уровня, ориентированных на прикладные задачи (уровни 5, 6 и 7).

Являясь эталонной моделью, OSI, тем не менее, не признается в качестве стандарта. Однако во многих изделиях в области телекоммуникаций и построения сетей за основу взята модель ISO/OSI.

Различают семь следующих уровней:

Физический уровень (уровень 1): физический уровень описывает передачу битов данных по сети на уровне кабелей и разъемов.

Канальный уровень (уровень 2): на канальном уровне данные группируются в кадры, к ним добавляется информация, необходимая для передачи кадров. Уровень 2 отвечает за транспортировку кадров данных от узла к узлу и за коррекцию ошибок.

Сетевой уровень (уровень 3): сетевой уровень совместно с уровнем 2 управляет маршрутизацией кадров. Он отвечает за адресацию кадров и их маршрутизацию в сети. Примером такого уровня является протокол Internet.

Транспортный уровень (уровень 4): координирует передачу пакетов данных. На этом уровне выполняется проверка, были ли все пакеты приняты полностью. Типичным примером уровня 4 является протокол TCP (Протокол управления передачей).

Сеансовый уровень (уровень 5): на сеансовом уровне устанавливается соединение между процессами, протекающими на различных станциях. Он отвечает за открытие "сессии" и за непрерывный диалог в виде отправки запросов и ответов между отдельными приложениями.

Уровень представления (уровень 6): уровень представления отвечает за преобразование данных в кадры, необходимые для конкретного приложения. На этом уровне также выполняется сжатие текстовых данных и преобразование различных кодов, используемых коммуникационными партнерами.

Прикладной уровень (уровень 7): прикладной уровень отвечает за прикладные задачи, непосредственно доступные пользователю, например, за передачу файлов или почтовые программы. Для пользователя передача данных представляется как последовательность запросов или ответов.

Точка доступа к сервису (SAP)

Точка доступа к сервису (службе) – это точка, в которой уровень эталонной модели ISO/OSI предоставляет свои сервисы (службы) следующему, более старшему уровню.

Обмен информацией между двумя соседними уровнями полностью происходит посредством точек доступа к сервисам. Точка доступа к сервису – это интерфейс между верхним и нижним уровнями. Если SAP находится между уровнем 3 (сетевым уровнем) и уровнем 4 (транспортным уровнем), такую точку доступа к сервисам называют NSAP (точка доступа к сетевым сервисам), а точку SAP между уровнем 4 и уровнем 5 (сеансовым уровнем) называют точкой TSAP (точка доступа к транспортным сервисам).

Как правило, точке доступа к сервисам назначается ряд ресурсов и коммуникационных партнеров, поэтому точки SAP, необходимые для осуществления связи, обозначаются уникальными именами и номерами.

2.2 Место SIMATIC NET в проектировании систем промышленной связи

Введение

В данном разделе описаны возможные способы применения продукции SIMATIC NET.

2.2.1 SIMATIC NET для систем промышленной связи

Характеристики продукции SIMATIC NET

Имя SIMATIC NET объединяет полностью все семейство коммуникационных продуктов и сетей фирмы Siemens. Различные сети удовлетворяют широкому кругу требований к эксплуатационным характеристикам применительно к различным задачам, которые стоят перед проектировщиком систем автоматизации:

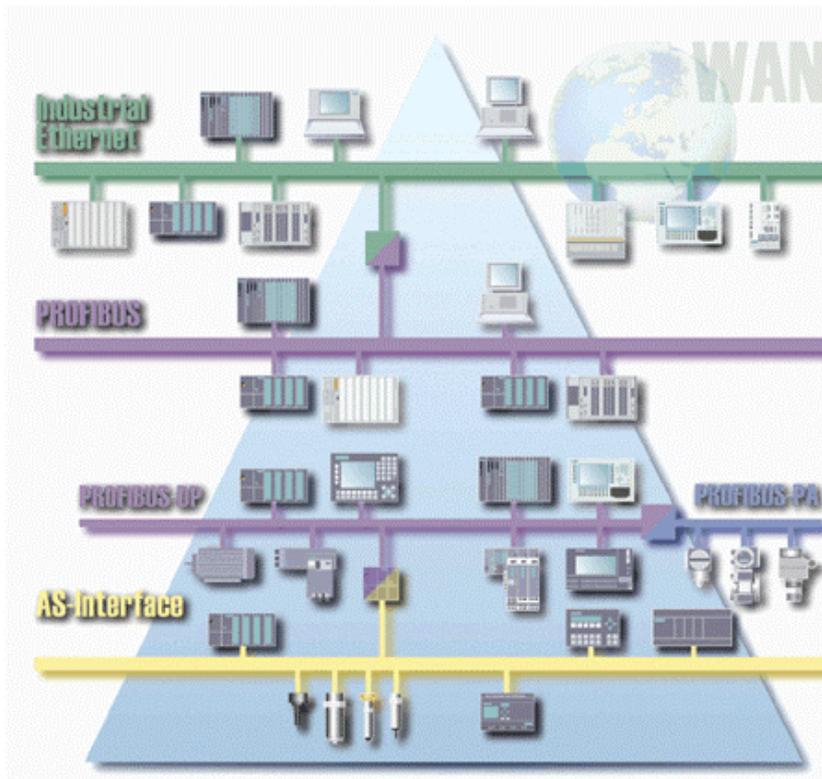
- SIMATIC NET предоставляет решения, отвечающие запросам современной промышленной связи
- SIMATIC NET занимает центральное место в системе автоматизации SIMATIC
- SIMATIC NET обеспечивает однотипный интерфейс между системами и другими компонентами автоматизации
- Связь характеризуется полнотой и стопроцентной интеграцией в систему

Коммуникационные сети семейства SIMATIC NET являются компонентами концепции TIA (Полностью интегрированная автоматизация) от Siemens.

2.2.2 Коммуникационные системы SIMATIC NET

Введение

Шинные системы SIMATIC NET и их функции вписываются в пирамидообразную модель автоматизации. В данном разделе описаны уровни этой модели:



Полевой уровень

На полевом уровне происходит обмен данными между полевыми устройствами, включенными в сеть полевого уровня, и более высоким уровнем, которым является узел, осуществляющий управление. Полевые устройства расположены на уровне отдельного производственного участка, они выполняют измерение, формируют сигналы и передают команды на более высокий уровень в систему управления процессами или общезаводскую систему управления. Объем данных, передаваемых на этом уровне, относительно мал. Для полевого уровня типична иерархическая структура коммуникаций, когда несколько полевых устройств обмениваются данными с одним ведущим устройством.

На полевом уровне происходит сбор сигналов процесса с помощью устройств ввода и вывода, после чего они через коммуникационную систему передаются программируемому контроллеру. Решением от Siemens для связи на полевом уровне и связи с процессом является система PROFIBUS-DP и AS-интерфейс.

Уровень производственного участка

Это тот уровень, на котором сигналы, полученные от процесса с помощью устройств ввода и вывода, поступают на программируемый контроллер через коммуникационную систему. На уровне производственного участка (который по-другому можно назвать уровнем отдельной ячейки автоматизации) связываются между собой программируемые контроллеры, персональные компьютеры (ПК) и устройства операторского управления и визуализации.

На уровне производственного участка программируемые контроллеры обмениваются информацией, которая позволяет им управлять общими задачами, либо к программируемым контроллерам подключаются локальные системы операторского управления и визуализации. Обмен данными на уровне производственного участка (уровне отдельной ячейки автоматизации) может быть реализован с помощью компонентов от Siemens, предусмотренных как для Industrial Ethernet, так и для PROFIBUS.

Верхний уровень управления

На этом уровне решаются и выполняются более широкие задачи, которые влияют на весь производственный или технологический процесс в целом (функции менеджмента). В эти задачи входит хранение значений процесса, оптимизация процессов и аналитические функции, а также функции протоколирования и архивирования. Можно, к примеру, собирать требуемые данные от нескольких производственных участков или отдельных заводов и обрабатывать их централизованно. Таких ячеек может быть не одна тысяча.

На верхнем уровне управления можно решать задачи контроля над всем производственным процессом в целом. Сюда собираются данные со всего завода, а команды (например, рецептура) могут передаваться на нижестоящие производственные ячейки. Для осуществления связи и обмена данными на верхнем уровне управления идеальной коммуникационной системой является Industrial Ethernet от Siemens.

Примечание

Промышленная связь с ПК

В качестве компонента системы промышленной автоматизации персональный компьютер (ПК), с одной стороны, может выполнять задачи управления полевого уровня, а с другой стороны, может служить в качестве станции операторского управления и визуализации на уровне отдельной ячейки автоматизации (производственного участка) или на верхнем уровне управления. В данной документации рассматривается промышленная связь с ПК только по PROFIBUS и Industrial Ethernet.

2.3 Промышленная связь через PROFIBUS

Введение

В этом разделе приводится общее описание шинной системы PROFIBUS.

2.3.1 Описание PROFIBUS

Введение

PROFIBUS – это открытый международный стандарт, который описывает шинную систему, предназначенную для установления связи с процессами и полевыми устройствами на полевом уровне, а также для обмена данными в пределах отдельной ячейки автоматизации. Обмен данными по PROFIBUS осуществляется либо с помощью недорогого медного кабеля (витой пары) или с помощью волоконно-оптического кабеля, не подверженного воздействию электромагнитных помех.

В этом разделе описаны устройства, которые могут участвовать в коммуникациях в системе PROFIBUS. В разделе также описаны механизмы доступа к шине, а также пояснено место системы PROFIBUS в эталонной модели ISO/OSI.

2.3.2 Протоколы и устройства для SIMATIC NET PROFIBUS

Связь с процессом и обмен данными

В семейство SIMATIC NET входят следующие протоколы PROFIBUS и типы устройств:

Для связи с процессом на полевом уровне:

Протокол	Типичный модуль для ПК	Описание	Типичные коммуникационные партнеры от Siemens
Ведущее устройство DP класса 1	CP 5613 CP 5614 CP 5511 CP 5611	Основное устройство для чтения входных данных и передачи выходных данных от/на ведомые устройства DP. Ведущее устройство с расширением DPV1 (опция) также может обращаться к данным ведомых устройств DPV1 в асинхронном режиме.	ET 200B ET 200X ET 200L ET 200LS CP 5614
Ведущее устройство DP класса 2	CP 5611 CP 5511 CP 5613	Устройства для диагностики и отладки системы DP. Ведущее устройство с расширением DPV1 (опция) также может обращаться к данным ведомых устройств DPV1.	CP 5613 CP 342–5 CP 343–5 CP 443–5 И ведомые устройства, перечисленные выше
Ведомое устройство DP	CP 5611 CP 5511 CP 5614	Модуль распределенного ввода/вывода, например, для опроса и преобразований сигналов процесса и передачи выходных сигналов.	CP 5613 CP 5614 CP 5511 CP 5611 CP 342–5 CP 443–5

Для обмена данными:

Протокол	Типичный модуль для ПК	Описание	Типичные коммуникационные партнеры от Siemens
FMS	CP 5613 CP 5614	Открытый протокол для подключения программируемых контроллеров различных производителей в единую сеть на уровне отдельной ячейки автоматизации с несколькими узлами	CP 5613 CP 343–5 CP 443–5 CP 5431 FMS/DP
Протокол S7	CP 5613 CP 5614 CP 5511 CP 5611	Встроенные оптимизированные функции связи систем SIMATIC S7/C7 для широкого круга задач	CP 5611 CP 5613 CP 342–5, 343–5 CP 443–5
S5-совместимые коммуникации	CP 5613 CP 5614 CP 5511 CP 5611	Простые коммуникационные сервисы на базе PROFIBUS FDL для обмена данными с устройствами S5 и S7.	CP 5431 CP 5434 CP 342–4, 343–5 CP 443–5

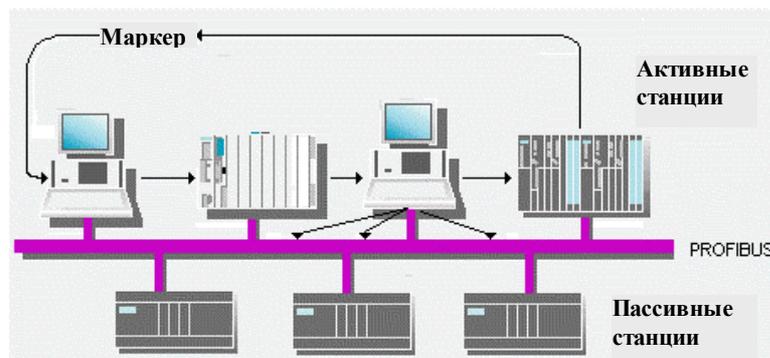
2.3.3 Метод доступа к шине PROFIBUS

Введение

Спецификация PROFIBUS является достаточно гибкой, позволяя применять различные протоколы, оптимально подходящие для решения определенных задач в различных областях применения. Единое управление доступом к шине обеспечивается на канальном уровне FDL (уровень 2 эталонной модели OSI/ISO).

Передача маркера

В PROFIBUS для управления доступом к шине используется технология передачи маркера. Это означает, что только та станция, которая обладает в данный момент маркером, может передавать данные в сеть. Спустя некоторое фиксированное время маркер передается следующей станции. По завершению цикла маркер вновь поступает на первую станцию.



Активные и пассивные узлы

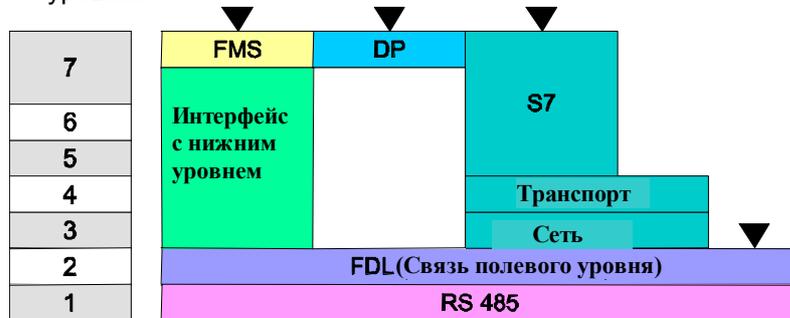
В сети PROFIBUS различают два основных типа узлов:

- Активные узлы управляют коммуникациями в шине. Каждый активный узел в пределах одного цикла только один раз получает маркер и может обмениваться данными с активными и пассивными узлами. Как только время удержания маркера истекло, узел передает маркер следующему ведущему устройству. Ведущие устройства DP и S7-серверы, к примеру, являются активными станциями.
- Пассивные узлы не могут быть инициаторами связи. Они не принимают маркер и отвечают только на запросы со стороны другой станции. Типичным примером пассивной станции является ведомое устройство DP.

2.3.4 Место PROFIBUS в эталонной модели ISO/OSI

Введение

PROFIBUS базируется на эталонной модели ISO/OSI, но не охватывает все ее уровни:



▼ Интерфейс пользователя

Уровень 1

На физическом уровне (уровне 1), на котором устанавливаются электрические и механические характеристики шины, для PROFIBUS используется широко распространенный интерфейс RS–485. Стандартом PROFIBUS определен выбор параметров шины, которые можно варьировать для выполнения специальных требований шинной системы.

Электрические характеристики:

Максимальное количество узлов	126
Стандартные скорости передачи	9.6 / 19.2 / 45.45 / 93.75 / 187.5 / 500 / 1500 / 3000 / 6000 / 12000 кбит/с
Длина кабеля	Зависит от скорости передачи и среды связи, находится в пределах от 200 м до 90 км
Топология	Шина Звезда Кольцо (только для оптических сетей)
Кабель ЛВС	Экранированная витая пара Волоконно-оптический кабель
Согласующая нагрузка	На обоих концах сегмента электрической шины должны быть подключены согласующие резисторы.

Механические характеристики (электрическая сеть):

Подсоединение	9–контактное гнездо типа D–sub на интерфейсном устройстве, с определенной разводкой контактов
---------------	---

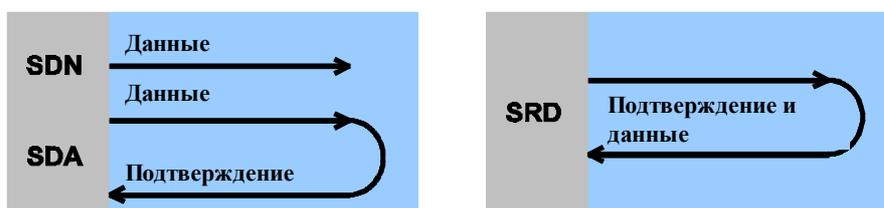
Уровень 2

На канальном уровне (уровне 2) описан упомянутый ранее метод доступа к шине и основные сервисы для передачи данных. Этот уровень известен также как уровень связи полевого уровня (FDL). Он обеспечивает универсальность протокола для доступа к шине, описывающего базовые характеристики коммуникаций для любых прикладных задач.

Сервисы FDL можно разделить на управляющие сервисы и производственные сервисы. С помощью управляющих сервисов можно настраивать локальные параметры, необходимые для использования производственных сервисов. Производственные сервисы отвечают за обмен данными между различными узлами PROFIBUS.

Для передачи данных предусмотрены следующие сервисы:

- SDN (Передача данных без подтверждения)
- SDA (Передача данных с подтверждением)
- SRD (Передача и запрос данных с ответом)



Уровни 3 ... 7

Уровни 3 ... 6 в неявном виде реализованы протоколами FMS и DP, описанными в стандарте PROFIBUS. В протоколе DP уровни 3 ... 6 полностью отсутствуют. Адаптация осуществляется посредством интерфейса нижнего уровня (LLI), назначенного уровню 7, на котором реализуются такие подфункции, как установление соединения, завершение соединения и контроль.

S7-коммуникации – это расширение PROFIBUS, делающее эту шину однородным компонентом мира SIMATIC, посредством которого также реализуются элементы сетевого уровня (уровня 3) и транспортного уровня (уровня 4).

2.4 S5–совместимые коммуникации через FDL (протокол SEND/RECEIVE)

Введение

S5–совместимые коммуникации через PROFIBUS позволяют реализовать простой обмен данными в сети PROFIBUS между контроллерами SIMATIC S5, контроллерами SIMATIC S7, персональными компьютерами и рабочими станциями.

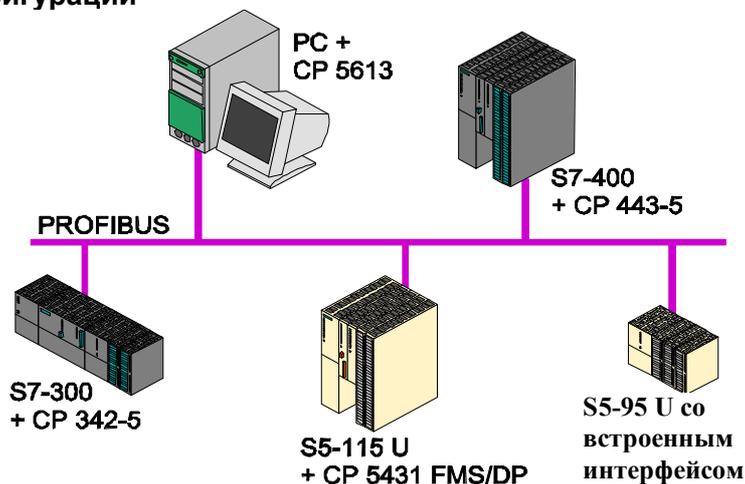
В этом разделе описан принцип организации коммуникаций, а также сервисы протокола SEND/RECEIVE. В разделе также поясняется структура и конфигурация.

2.4.1 Типичная конфигурация системы для S5–совместимых коммуникаций

Введение

Для реализации обмена данными с использованием S5–совместимых коммуникаций предусмотрены специальные коммуникационные модули для контроллеров семейств SIMATIC S5, SIMATIC 505 и SIMATIC S7, а также для ПК и рабочих станций.

Пример конфигурации



2.4.2 Принцип коммуникаций SEND/RECEIVE

Общее описание сервисов

S5–совместимые коммуникации базируются на простой передаче данных в кадре FDL. Во время передачи данных приемник предоставляет передатчику буфер приема, куда передатчик передает содержимое своего буфера данных. Для S5–совместимых коммуникаций используются сервисы канального уровня PROFIBUS, т.е., непосредственно сервисы уровня полевой связи (FDL).

Примечание

OPC-элементы, которые используются в ПК в случае S5-совместимых коммуникаций по PROFIBUS, снабжаются префиксом "FDL".

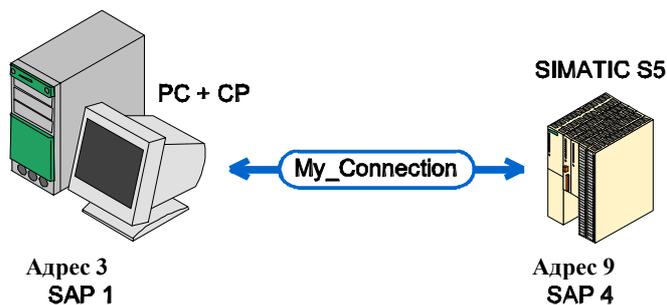
S5–совместимые коммуникации возможны только между активными узлами PROFIBUS. Длина данных пользователя ограничена длиной кадров FDL и не может превышать 246 байтов в одном кадре. Для обмена данными используются сервисы SDA (Передача данных с подтверждением) и SDN (Передача данных без подтверждения).

Для коммуникаций не требуется установление соединения. Передатчик и приемник назначаются друг другу путем указания адреса партнера и точки доступа к сервисам (SAP). Путем выбора определенных точек SAP можно передавать кадры нескольким узлам одновременно (групповая передача/широковещание).

2.4.3 Конфигурирование SEND/RECEIVE

Адрес и точка доступа к сервисам

Прежде чем приступить к использованию S5–совместимых коммуникаций по PROFIBUS, следует сконфигурировать взаимосвязи между коммуникационными партнерами. Коммуникационные взаимосвязи конфигурируются путем указания имени, адреса партнера и точки доступа к сервисам.

Пример**2.4.4 Сервисы SEND/RECEIVE****Введение**

В этом разделе приводится обзор коммуникационных сервисов, которые можно использовать для S5–совместимых коммуникаций.

Функции

S5–совместимые коммуникации по PROFIBUS предоставляют следующие функции:

- Сервисы для передачи буферов данных (SEND / RECEIVE)

Для ПК также предусмотрены следующие информационные сервисы:

- Получение параметров шины и адреса локальной станции
- Получение списка станций в шине
- Идентификация локальной станции и партнерской станции

Примечание

В отличие от S5-совместимых коммуникаций через Industrial Ethernet, в состав сервисов для работы с буферами в PROFIBUS не входят сервисы "Fetch" ("Считать") и "Write" ("Записать") для работы с переменными.

Описание сервисов

S5–совместимые коммуникации предоставляют сервисы для передачи блоков данных с помощью сервисов FDL SDN и SDA. Передача пакета от передатчика к получателю должна инициироваться передатчиком явным образом. Если передающим узлом является, например, контроллер SIMATIC S7, он инициирует связь путем вызова блока "AG_SEND", а если передатчиком является ПК с OPC, он вызывает метод для записи переменной "SEND".

Приемник не может инициировать передачу данных со стороны передатчика.

Размер передаваемого буфера данных ограничивается длиной кадра PROFIBUS, которая составляет 244 байта (или 246 байтов в случае использования точек SAP, принимаемых по умолчанию).

В случае использования точки SAP для группового вещания (63) и адреса для группового вещания (127) один кадр можно одновременно передать нескольким станциям.

Поскольку данным протоколом не предусматривается установление соединения и, следовательно, отсутствует контроль соединения, приемник не может обнаружить неисправность партнерского устройства. Однако в контроллерах и ПЛК можно запрограммировать циклическую передачу заданий и предусмотреть контроль приема этих заданий партнером, реализовав, таким образом, контроль.

Примечание

В случае использования OPC-сервера буферы приема создаются путем инициализации переменной "RECEIVE", а буферы передачи создаются путем инициализации переменной "SEND".

В случае использования OPC-сервера имеется возможность "привязки" блока данных или его отдельных частей к отдельным переменным OPC, а также назначения этим частям определенных типов данных. Такая структуризация позволяет пользователю OPC более просто использовать буфер-ориентированные сервисы.

Преимущества / недостатки

S5–совместимые коммуникации через FDL обладают следующими преимуществами и недостатками:

Преимущества

- Большие объемы блоков данных, вплоть до 244 или 246 байтов.
- Сеть не нагружена, когда не передаются данные.
- Возможно "широковещание" кадров нескольким узлам.
- На ПК с OPC можно структурировать обращение к блокам данных.
- Возможна связь с устройствами SIMATIC S5 и SIMATIC S7.
- Возможна связь между двумя ПК/PG.

Недостатки

- Приемник не может инициировать передачу данных. Он должен ожидать, пока передатчик не передаст данные.
- Протокол не предусматривает для принимающего узла возможность контроля неисправности партнера или разрыва сети.
- Отсутствует маршрутизация (перенаправление задания в другие сети).

2.5 Протокол DP

Введение

В этом разделе описан принцип организации связи и сервисы протокола DP. В этом разделе также описана структура и конфигурация.

2.5.1 Протокол DP и его расширения

Описание протокола DP

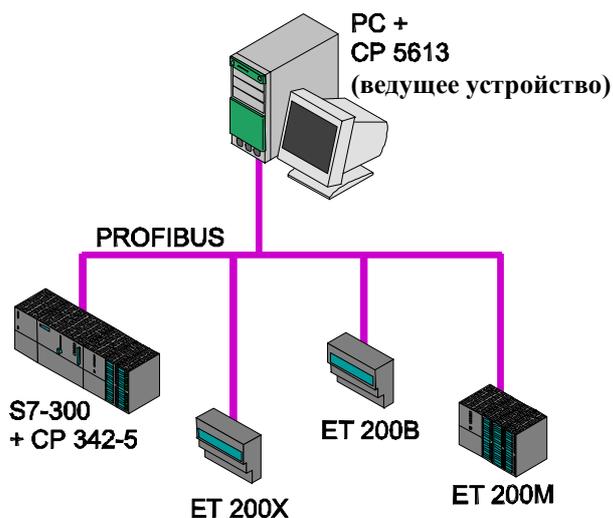
Система распределенного ввода/вывода (DP = децентрализованная периферия) позволяет создать распределенную структуру, состоящую из большого количества модулей ввода и вывода аналоговых и дискретных сигналов, расположенных непосредственно по месту технологического процесса.

Отдельные устройства ввода/вывода, объединенные по PROFIBUS, могут находиться на больших расстояниях друг от друга. Локальные станции распределенного ввода/вывода собирают входные сигналы, которые затем циклически считываются центральным контроллером. И наоборот, центральный контроллер циклически передает выходные данные станциям распределенного ввода/вывода.

DP-протокол предназначен для прикладных задач, критичных ко времени. Простой оптимизированный протокол передачи данных, высокие скорости передачи и использование принципа "ведущий-ведомый" позволяют достичь коротких длительностей цикла.

Являясь открытым протоколом, DP базируется на стандарте, описывающем коммуникации полевого уровня (IEC 61158).

Пример конфигурации



Характеристики протокола DP

Система распределенного ввода/вывода обладает следующими базовыми характеристиками:

- Централизованное управление осуществляется ведущим устройством
- Высокая пропускная способность с использованием простого протокола передачи
- Циклическая передача образа процесса в направлениях ввода и вывода
- Обнаружение ошибок с применением online-диагностики
- Базируется на PROFIBUS FDL, что позволяет работать совместно с другими устройствами (ведущими и ведомыми устройствами) в одной шине.

DPV1

Стандарт DPV1 является расширением DP-коммуникаций. Ведомые устройства, поддерживающие DPV1, обладают дополнительной областью памяти, в которой могут храниться специальные данные ведомых устройств. С использованием функций DPV1 "записи" данных можно читать или записывать, что расширяет функциональность системы.

Характеристики протокола DP и его расширений

Для протокола DP предусмотрено несколько расширений, которые позволяют реализовать специальные функции. В следующей таблице предусмотрен обзор различных ведущих устройств DP и их расширений.

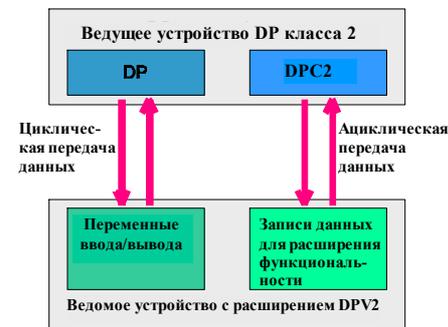
Описание протоколов

Протоколы	Описание
Ведущее устройство DP класса 1	Предоставляет сервисы для настройки параметров ведомых устройств и циклического обмена данными.
Ведущее устройство DP класса 2	Дополнительные функции диагностики. Ведущее устройство класса 2 может опрашивать состояние ведущего устройства класса 1 или ведомого устройства, не мешая работе сети.
DPC1	Расширение DPV1 для ведущих устройств класса 1. Оно позволяет реализовать ведущее устройство C1, а также циклически записывать и читать дополнительные области данных ведомого устройства DPV1.
DPC2	Расширение DPV1 для ведущих устройств класса 2. Позволяет реализовать ведущее устройство C2, а также циклически записывать и читать дополнительные области данных ведомого устройства DPV1.

Ведущее устройство DP класса 1 и DPC1:



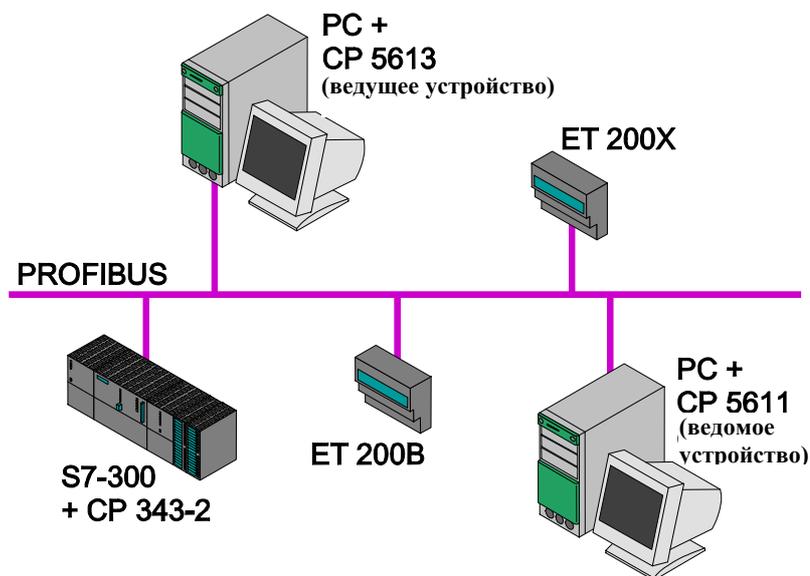
Ведущее устройство DP класса 2 и DPC2:



2.5.2 Типичная конфигурация системы DP

Пример

На следующем рисунке показан типичный пример конфигурации системы:



2.5.3 Принцип организации DP-коммуникаций

Коммуникационные партнеры

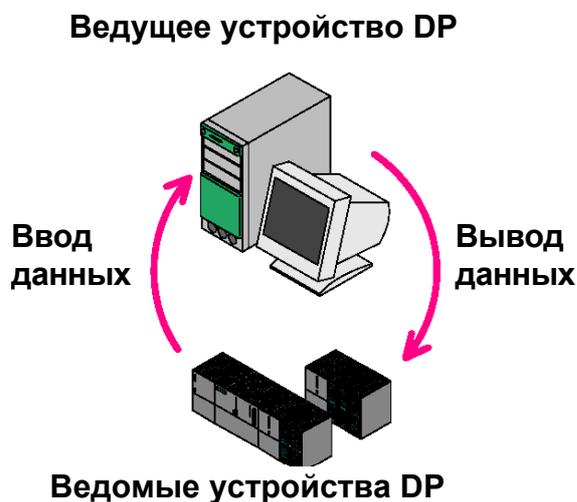
В системе распределенного ввода/вывода различают три типа коммуникационных партнеров:

Свойства коммуникационных партнеров DP

Коммуникационные партнеры DP	Описание
Ведомые устройства DP	Пассивные узлы шины, как правило, устройства ввода/вывода.
Ведущее устройство DP класса 1	Активный узел, центральный компонент для управления ведомыми устройствами DP.
Ведущее устройство DP класса 2	Активный узел, который можно использовать для отладки и диагностики одновременно с ведущим устройством класса 1.

Опрос

В общем случае связь между ведущим устройством DP и станциями распределенного ввода/вывода имеет форму опроса. Опрос означает, что ведущее устройство DP циклически передает кадры опроса назначенным ему ведомым устройствам DP. Каждому ведомому устройству DP передается отдельный кадр опроса.



Кадр опроса содержит текущие выходные данные, которые ведомое устройство DP должно вывести на свои выходы. Если у ведомого устройства DP отсутствуют выходы, вместо этих кадров передается "пустой кадр".

Адресуемое ведомое устройство DP должно подтвердить, что оно приняло кадр опроса, вернув в ответ кадр подтверждения. В кадре подтверждения содержатся текущие входные данные, присутствующие на входах ведомого устройства DP. Если у ведомого устройства DP отсутствуют входы, вместо них передается "пустой кадр".

Все работоспособные ведомые устройства DP адресуются в одном цикле опроса. Следующий цикл опроса начинается сразу же после того, как произошло обращение к последнему ведомому устройству. Такой способ обеспечивает актуальность данных. В каждом цикле опроса ведущее устройство DP пытается включить в цикл неработоспособные ведомые устройства.

Примечание

OPC-элементы, которые используются в ПК для DP-коммуникаций через PROFIBUS, снабжаются префиксом "DP".

Отсутствие управления передачей данных в DP

Протокол DP оптимально подходит для быстрого обмена данными между ведущим устройством и ведомым устройством, и в нем не предусмотрено управление потоком данных между ведущим и ведомым устройствами. Ведущее и ведомое устройства обмениваются данными ввода и вывода циклически. При написании DP-приложения необходимо помнить о следующем:

- Если данные вывода ведущего устройства изменяются в пределах одного цикла несколько раз, некоторые значения могут быть не переданы ведомым устройствам.
- Если данные ввода ведущего устройства не читаются в каждом цикле, некоторые изменения значений, произошедшие на ведомом устройстве, могут быть пропущены.

2.5.4 Конфигурирование системы DP

Введение

Ведущее устройство DP может участвовать в производительном обмене данными с ведомыми устройствами DP только в том случае, если ранее оно настроило параметры этих ведомых устройств и сконфигурировало их.

Ведущее устройство конфигурирует ведомые устройства и задает их параметры в следующих случаях:

- На этапе запуска ведущего устройства DP.
- После временного выхода из строя ведомого устройства DP во время работы.

Требуемая конфигурация и параметры указываются и сохраняются на этапе конфигурирования сети.

Кадр назначения параметров

Кадр назначения параметров устанавливает глобальные рабочие параметры ведомого устройства DP (например, время для сторожевого таймера).

Кадр конфигурации

Кадр конфигурации передается после успешной установки параметров ведомого устройства DP. Он содержит текущую конфигурацию ведомого устройства DP. Конфигурация включает в себя количество и тип портов ввода/вывода (входов/выходов).

Ведомое устройство DP сравнивает принятый кадр конфигурации со значениями, которые оно получило само на этапе запуска. Если значения совпадают, ведомое устройство DP подтверждает конфигурацию и переходит в режим работы.

2.5.5 Ведущее устройство DP класса 1 – Принципы коммуникаций и сервисов

Режимы работы ведущего устройства DP

Ведущее устройство DP управляет состоянием системы DP. В каждом режиме ведущее устройство DP характеризуется определенным типом взаимодействия с ведомыми устройствами DP:

Режим	Назначение
OFFLINE	Не происходит никаких DP-коммуникаций между ведущим устройством DP и ведомыми устройствами DP. Это исходный режим DP-ведущего.
STOP	В этом режиме также не происходит DP-коммуникаций между ведущим устройством DP и ведомыми устройствами DP. В отличие от режима OFFLINE, в этом режиме станция диагностики DP (ведущее устройство DP класса 2) может читать диагностическую информацию из ведущего устройства DP.
CLEAR	Всем ведомым устройствам DP, введенным в базу данных и активизированным, назначаются параметры и конфигурация. После этого в режиме CLEAR на все выходы всех ведомых устройств передается значение 0h, т.е., выходы отключаются. Входные данные ведомых устройств являются известными, и их можно читать.
OPERATE	В этом режиме устанавливается циклический обмен данными с ведомыми устройствами DP. Этот режим также называют рабочим (производственным) режимом. В этом режиме ведомые устройства DP опрашиваются один за другим ведущим устройством DP.

Примечание

Начиная с текущего режима, режимы должны следовать друг за другом в выбранном порядке (по возрастанию или по убыванию) OFFLINE – STOP – CLEAR – OPERATE!

Если используются стандартные параметры, OPC-сервер SIMATIC NET управляет сменой режимов ведущего устройства DP автоматически.

Описание сервисов

Обращение к переменным процесса из прикладной задачи пользователя, работающей на ведущем устройстве класса 1, происходит не напрямую, а посредством образа процесса в коммуникационном модуле. Образ процесса содержит входные данные, прочитанные из всех ведомых устройств в последнем цикле, и выходные данные, которые должны быть записаны в следующем цикле.

Образ процесса состоит из трех областей данных для каждого ведомого устройства DP:

- Входные данные (данные ввода) от ведомого устройства DP
- Выходные данные (данные вывода) на ведомое устройство DP
- Данные диагностики от ведомого устройства DP

Приложение, работающее на ПК, может использовать следующие сервисы через ведущее устройство DP класса 1:

Сервисы для работы с переменными для образа процесса ведущего устройства DP

Предусмотрены следующие информационные сервисы:

- Режим ведущего устройства DP и ведомых устройств DP
- Сообщения о событиях от ведущего устройства DP
- Контроль работы (активности) модулем DP
- Тип ведомого устройства DP

Преимущества / недостатки

Использование ведущих устройств класса 1 обладает следующими преимуществами и недостатками:

Преимущества

- Быстрый доступ к циклическим данным
- Задания, поступающие от приложений, могут выполняться необычайно быстро, поскольку данные могут быть получены непосредственно из образа процесса и не связаны явным образом с коммуникациями.
- Простые (а значит, недорогие) коммуникационные модули.

Недостатки

- Высокая нагрузка на шину вследствие циклического обмена данными ввода/вывода.
- Синхронное обращение к переменным процесса с использованием OPC в пределах цикла DP невозможно.

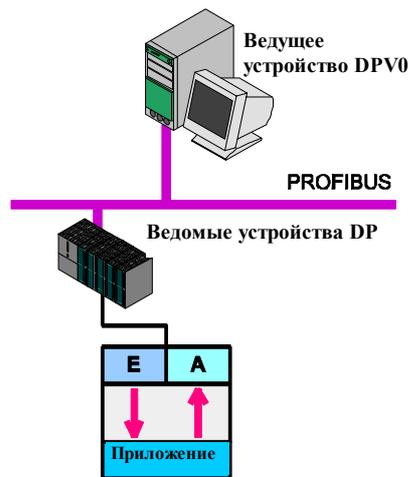
2.5.6 Ведомые устройства DP – Принципы коммуникаций и сервисов

Принципы коммуникаций

Ведомое устройство DP обеспечивает обмен данными. Оно предоставляет входные данные, которые могут быть прочитаны ведущим устройством DP в следующем цикле опроса, и оно читает и использует выходные данные, передаваемые ведущим устройством DP. Ведомое устройство DP также может предоставлять данные диагностики, которые читаются ведущим устройством DP в следующем цикле.

В рамках DP-коммуникаций ведомые устройства DP считаются модульными устройствами. Каждое ведомое устройство может быть составлено из нескольких подмодулей, каждый из которых имеет свои собственные области ввода и вывода.

Ведомое устройство, поддерживающее расширение DPV1, может содержать дополнительные записи данных для каждого модуля. Эти записи содержат специальные данные ведомых устройств, которые могут быть прочитаны и записаны ведущим устройством DPV1. Каждая запись может содержать до 240 байтов полезных данных.



Конфигурирование ведомого устройства DP с помощью STEP 7

2.5.7 Ведущее устройство DP класса 2 – Принципы коммуникаций и сервисов

Принцип коммуникаций

Помимо устройств, являющихся ведущими устройствами DP класса 1, в системе DP также могут находиться устройства, являющиеся ведущими устройствами DP класса 2. Устройства этого класса используются во время отладки системы, ее конфигурирования и диагностики. Они, как правило, служат для программирования, конфигурирования или в качестве устройств операторского управления.

Ведущее устройство класса 2 можно, например, включить в сеть PROFIBUS для целей диагностики. В этом случае оно опрашивает состояние ведомых устройств и ведущих устройств класса 1 в любое время, не мешая работе сети. Ведущее устройство класса 2 может также изменить адрес ведомого устройства, если последнее это позволяет.

Входы и выходы, а также диагностические данные ведомого устройства могут быть только прочитаны; в выходы запись осуществлять нельзя.

Ведущее устройство класса 2 выполняет следующие основные функции:

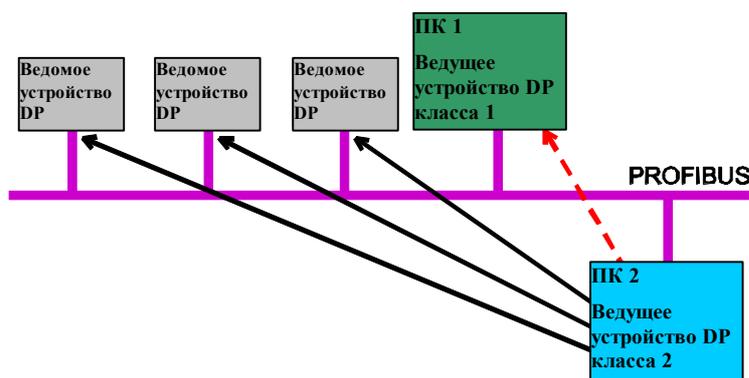
- Чтение данных ведомого устройства
- Чтение данных ведущего устройства класса 1

Поскольку эти сервисы выполняются вместе с циклическими сервисами, ведущее устройство класса 2 увеличивает нагрузку на сеть.

Описание сервисов

Практически так же, как и ведущее устройство класса 1, ведущее устройство класса 2 может также обращаться к циклическим данным ввода и вывода и связывать информацию с переменными. В то же время, данные могут быть только прочитаны, но не записаны.

Другие переменные позволяют обращаться к диагностическим данным ведущего устройства класса 1 и ведомых устройств. Записывая значения в специальную переменную, ведущее устройство класса 2 также может устанавливать адрес ведомого устройства.



Ведущее устройство DP класса 2 обеспечивает следующие типы коммуникаций:

- ← Диагностика, чтение входов, чтение выходов, чтение конфигурации, установка адреса ведомого устройства
- ← - Диагностика ведущего устройства

Преимущества / недостатки

Использование ПК в качестве ведущего класса 2 обладает следующими преимуществами и недостатками:

Преимущества

- Воздействие на работу сети сводится к минимуму.
- Можно изменять конфигурацию ведомого устройства (устанавливать адрес ведомого устройства).

Недостатки

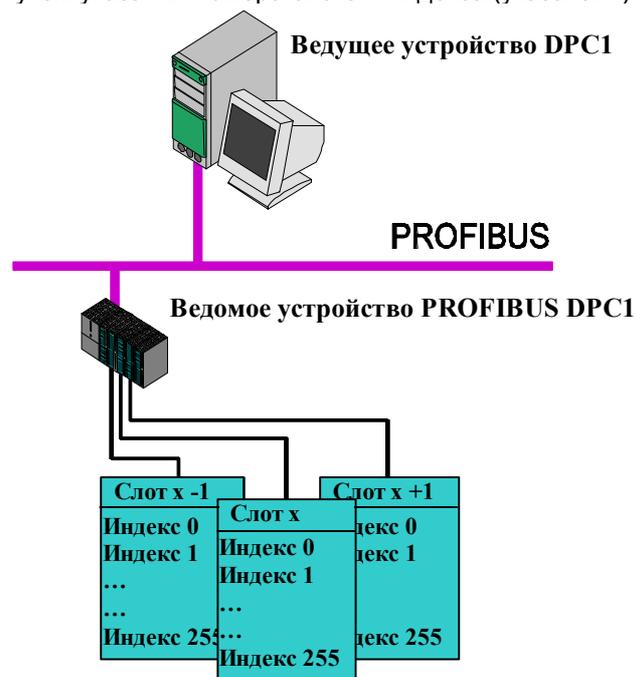
- Входы, выходы и диагностические данные ведомых устройств могут быть только прочитаны.
- Синхронный доступ к переменным процесса с помощью OPC в цикле DP невозможен.

2.5.8 DPC1 – Принципы коммуникаций и сервисов

Принципы коммуникаций

Спецификация DPV1 является расширением DP-коммуникаций для ведущих устройств класса 1 и класса 2. Специальные функции для ведущих устройств, принадлежащих классу 1, в SIMATIC NET известны как сервисы DPC1.

Сервисы DPC1 позволяют опрашивать данные ведомых устройств ациклически, дополнительно к циклическому опросу ведущим устройством DP. Каждое ведомое устройство DPV1 имеет дополнительную область данных, которая может читаться и записываться ведущим устройством DPV1. Эта область данных зависит от конкретного ведомого устройства и может содержать, например, значения параметров или аварийные сообщения. Обращение к отдельным "записям" данных в дополнительной области данных производится путем указания номера слота и индекса (указателя).



Описание сервисов

Существуют следующие сервисы DPC1:

- Ациклическое чтение и запись "записей" данных
- Обслуживание аварий

Поскольку сервисы DPV1 выполняются с меньшим приоритетом по отношению к сервисам циклического опроса данных, пропускная способность в DPV1-коммуникациях меньше. Время "кругового обхода" также, в общем случае, повышается, поскольку на сеть оказывается дополнительная нагрузка.

Для использования функций DPC1 не требуется устанавливать коммуникационные соединения с ведомыми устройствами, поскольку цикл опроса ведущего устройства уже инициирован, как неявное соединение. Поскольку параметры ведомого устройства с функциями DPC1 уже настроены и сконфигурированы, к нему можно обращаться посредством функций DPC1.

Обращаться можно только к целым блокам данных слота DPV1. В ответ на запрос на чтение возвращается содержание полностью всей "записи" данных, идентифицируемой номером слота и индексом (указателем), а запрос на запись приводит к перезаписи всей "записи" данных целиком.

Примечание

При использовании OPC-сервера имеется возможность структурирования "записи" данных. "Запись" данных можно разбить на отдельные части, назначить этим частям тип данных и использовать их в качестве переменных. Тем не менее, с точки зрения сети "запись" данных всегда читается и записывается как единое целое.

Преимущества / недостатки

Применение сервисов DPC1 обладает следующими преимуществами и недостатками:

Преимущества

- Возможен асинхронный доступ к ведомым устройствам.
- Могут передаваться большие блоки данных.
- В OPC возможен структурированный доступ к полям данных.
- Возможно параллельное использование совместно с сервисами обслуживания переменных ведущего устройства класса 1.

Недостатки

- Дополнительная нагрузка на сеть.
- Меньшая пропускная способность по сравнению с циклической передачей данных.

2.5.9 DPC2 – Принципы коммуникаций и сервисов

Принципы коммуникаций

Спецификация DPV1 является расширением DP-коммуникаций для ведущих устройств класса 1 и для ведущих устройств класса 2.

Функции, предназначенные специально для ведущих устройств класса 2, в SIMATIC NET известны как сервисы DPC2. Сервисы DPC2 позволяют ведущему устройству класса 2 опрашивать данные ведомых устройств ациклично, дополнительно к циклическому опросу ведущим устройством DP. Каждое ведомое устройство DPV1 имеет дополнительную область данных, которую может читать и записывать ведущее устройство DPV1. Эта область данных зависит от конкретного ведомого устройства и может содержать, например, значения параметров или аварийные сообщения. К отдельным "записям" данных дополнительной области данных можно обращаться, указывая номер слота и индекс (указатель).

Основная разница между коммуникациями DPC2 и обычными коммуникациями "ведущий-ведомый" состоит в том, что сначала должно быть установлено соединение. Далее это соединение поддерживается до тех пор, пока оно не будет разорвано вмешательством со стороны или ведущим устройством. После того, как это соединение установлено, ведущее устройство может обмениваться данными с ведомым устройством.

Наиболее важные функции DPC2 таковы:

- Установление и прерывание коммуникационного соединения
- Чтение данных ведомых устройств

Описание сервисов

Применяя DPC2, пользователь получает в свое распоряжение следующие сервисы:

Сервисы для работы с буферами через DPC2

Поскольку сервисы DPV1 выполняются с меньшим приоритетом по отношению к сервисам циклического опроса данных, пропускная способность в DPV1-коммуникациях меньше. Время "кругового опроса" также, в общем случае, увеличивается, поскольку сеть испытывает дополнительную нагрузку.

Точно так же, как и ведущее устройство DP класса 1 (DPC1), ведущее устройство класса 2 может обращаться только к целым блокам слота DPV1. В ответ на задание на чтение возвращается полностью все содержание "записи" данных, идентифицируемое номером слота и индексом (указателем), а задание на запись приводит к перезаписи полностью всей "записи" данных.

Примечание

При использовании OPC-сервера имеется возможность структурирования "записи" данных. "Запись" данных можно разбить на отдельные части, назначить этим частям тип данных и использовать их в качестве переменных. Тем не менее, с точки зрения сети "запись" данных всегда читается и записывается как единое целое.

Преимущества / недостатки

Применение сервисов DPC2 обладает следующими преимуществами и недостатками:

Преимущества

- Возможен асинхронный доступ к ведомым устройствам.
- Передаются большие блоки данных.
- Возможна структуризация доступа к полям данных.
- Возможно параллельное использование совместно с ведущим устройством класса 1.

Недостатки

- Дополнительная нагрузка на сеть.
- Меньшая пропускная способность.

2.6 Связь приводов с сервером шины PROFIDrive

Функции сервера шины

Сервер шины SIMATIC NET OPC PROFIDrive связывает элементы, соответствующие параметрам сервера общего профиля OPC, с сервисами SIMATIC NET DPC2. Другими словами, обмен данными между сервером шины PROFIDrive и приводами происходит с использованием протокола DPC2.

Связь между сервером шины PROFIDrive и сервером профиля, а также между сервером профиля и клиентом происходит через интерфейсы OPC, причем, клиент OPC PROFIDrive обращается только к серверу профиля OPC. Сервер профиля OPC (OPC Profile Server) не входит в состав программных пакетов диска SIMATIC NET CD.



2.7 Протокол FMS

Введение

В этом разделе поясняются принципы коммуникаций и сервисов протокола FMS. В разделе также описывается структура и конфигурация.

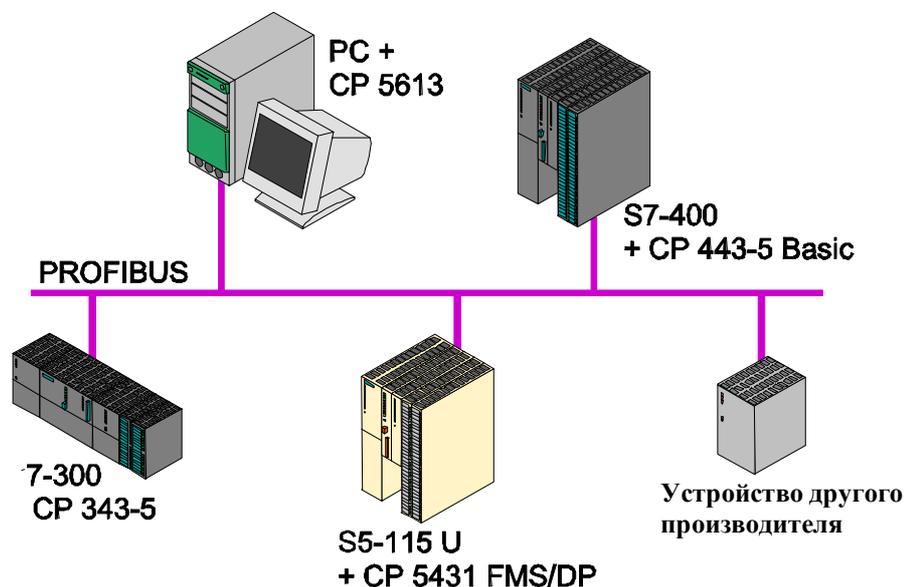
2.7.1 Типичная конфигурация системы FMS

Введение

Спецификация сообщений сети полевого уровня (FMS) описывает коммуникационный протокол, который определен стандартом EN 50170 на уровне 7 эталонной модели ISO/OSI.

Таким образом, FMS идеально подходит для связи между гетерогенными системами автоматизации различных производителей. Из-за своей сложности и, как следствие, ограниченной скорости обработки, он подходит только для применения на уровне отдельной ячейки автоматизации в сетях, состоящих из нескольких узлов.

Пример конфигурации



2.7.2 Принципы FMS-коммуникаций

Объекты коммуникаций

Цель модели FMS-коммуникаций состоит в том, чтобы объединить распределенные прикладные процессы в единый процесс. В FMS используется объектно-ориентированное представление: коммуникационным объектам сопоставляются такие объекты реального процесса, как, например, значение температуры, измеренное датчиком температуры. Спецификация FMS описывает структуру коммуникационных объектов и сервисов, которые могут с ними использоваться.

Доступ к коммуникационным объектам можно получить разными способами:

- Индекс (указатель) объекта: наиболее эффективным способом обращения является индекс объекта. Индекс – это адрес объекта в формате 16–битового счетчика.
- Символьное обращение: обращение к объекту происходит с использованием уникального символьного имени. Это более удобное решение для пользователя, поскольку на объект можно сослаться, к примеру, используя краткое описание его содержимого.

Коммуникационный объект описывается следующими атрибутами:

- индекс (указатель)
- символьное имя
- тип данных
- длина
- права доступа/пароль

Каждое FMS-устройство содержит словарь объектов (OD), в котором перечислены объекты, предоставляемые этим устройством. Протокол FMS предусматривает чтение словаря объектов партнерским устройством, чтобы последнее могло получить информацию о доступных объектах. В зависимости от конфигурации, словари объектов могут быть очень большими. Если словарь объектов загружается всякий раз, когда стартует клиент, это может существенно задержать установление соединения. Во избежание этой проблемы клиент FMS может загрузить словарь объектов партнера в какое-нибудь другое время и сохранить его у себя, например, в файл. Эта функция поддерживается сервером SIMATIC NET OPC.

Примечание

OPC-элементы, используемые на ПК в случае FMS-коммуникаций через PROFIBUS, помечаются префиксом "FMS".

Коммуникационный контакт

За несколькими исключениями (групповое вещание/широковещание), обмен данными всегда происходит между двумя коммуникационными партнерами. Один партнер берет на себя роль клиента, а второй партнер становится сервером.

Следовательно, коммуникации осуществляются между клиентом и сервером. Понятие "коммуникационный контакт" (Communication Relations), применяемое в протоколе FMS, соответствует по смыслу термину "соединение". Каждый узел может поддерживать несколько коммуникационных контактов с одним и тем же партнером или с различными партнерами одновременно.

Коммуникационный контакт описывается рядом параметров в коммуникационной базе. В состав параметров, описывающих коммуникационный контакт, входит адрес партнерской станции, точка доступа к сервисам, тип соединения и сервисы, поддерживаемые соединением, а также размер буферов приема и передачи.

Тип соединения

Тип соединения описывает тип узла, участвующего в коммуникациях (ведущее или ведомое устройство) и тип передачи на уровне 2 (циклическая или ациклическая передача). Типичной конфигурацией на уровне отдельной ячейки автоматизации являются ациклические коммуникации "ведущее устройство – ведущее устройство".

Виртуальные полевые устройства

Виртуальное полевое устройство (VFD) – это абстрактный (виртуальный) объект, который служит для описания данных и поведения устройства с точки зрения его коммуникационного партнера. VFD включает в себя все коммуникационные объекты (например, коммуникационный контакт) вместе с их свойствами и статусами. Физическому устройству может быть назначено несколько VFD. VFD может включать в себя коммуникационные контакты с несколькими физическими устройствами.

2.7.3 Конфигурирование FMS

Введение

Для осуществления коммуникацией с использованием протокола FMS через PROFIBUS на этапе конфигурирования заранее должны быть описаны коммуникационные объекты и коммуникационные соединения.

Указанные на этапе конфигурирования параметры используются коммуникационными компонентами в режиме выполнения, и их больше не требуется указывать в программе пользователя. В случае физического переконфигурирования системы изменения должны быть внесены только в конфигурацию, программы менять не требуется.

Объекты и соединения

На этапе конфигурирования FMS-коммуникаций указываются следующие элементы:

Виртуальные полевые устройства (VFD)

- Номер и имя виртуального полевого устройства
- Сведения об изготовителе
- Версия

Соединения FMS

- Название соединения
- Адреса партнерских станций
- Точки доступа к сервисам уровня FDL (точки LSAP)
- Тип коммуникаций/тип соединения
- Сервисы данного соединения
- Защита доступа

Объекты коммуникаций (коммуникационные объекты)

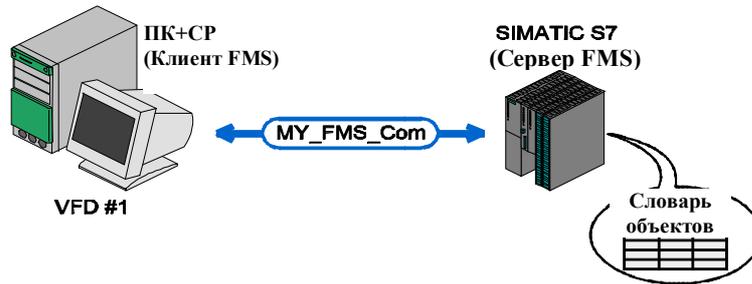
Словарь объектов FMS-сервера описывает объекты, к которым может обращаться OPC-клиент. Коммуникационный объект характеризуется следующими свойствами:

- индекс (указатель)
- символьное имя (не обязательно)
- тип данных
- длина
- права доступа

Если FMS-сервером является устройство S7, коммуникационные объекты указываются в символьной таблице путем определения дополнительных свойств.

Если ПК с OPC-сервером используется в качестве FMS-сервера, коммуникационные объекты должны быть описаны внутри описания VFD.

Пример



2.7.4 Сервисы FMS

Производственные и управляющие сервисы

Поскольку к системе FMS предъявляется широкий набор требований, спецификация FMS определяет разнообразные сервисы, из которых только небольшая часть является обязательной.

Различают производственные (рабочие) и управляющие сервисы:

Производственные (рабочие) сервисы

- READ (чтение коммуникационного объекта)
- WRITE (запись коммуникационного объекта)

Управляющие сервисы

- INITIATE (запуск установления соединения)
- ABORT (прерывание соединения)
- STATUS (чтение статуса партнера)
- IDENTIFY (идентификация коммуникационного партнера)
- GET-OD (чтение словаря объектов партнера)

Сервисы FMS для SIMATIC NET

- Сервисы для работы с переменными, ПК в качестве FMS-клиента
- Сервисы для работы с переменными, ПК в качестве FMS-сервера
- Идентификация и опрос атрибутов партнерского устройства
- Запрос статуса партнерского устройства
- Получение обзора узлов, участвующих в обмене данными через PROFIBUS

2.7.5 Сервисы FMS для работы с переменными, ПК в качестве FMS-клиента

Описание сервисов

Клиент (в данном случае – ПК) может обращаться к объектам, которые предоставлены FMS-сервером в качестве переменных. У клиента имеется два способа обращения к объекту:

- индекс (указатель) коммуникационного объекта
- символьное имя коммуникационного объекта

Клиент может получить список объектов, предоставляемых сервером, а также свойства переменной, например, сведения о правах доступа и типе данных, из словаря объектов сервера.

Когда начинаются коммуникации, клиент (ПК) может загрузить этот словарь объектов, чтобы получить всю информацию об объектах, доступных для коммуникаций. На стороне клиента переменные конфигурировать не требуется. Однако на стороне сервера должен быть сконфигурирован словарь объектов.

После того, как клиент получил словарь объектов, он может записывать или читать переменные (в зависимости от прав доступа, предоставляемых сервером).

Чтобы начать обмен данными с сервером, следует установить соединение между двумя узлами. Это, как правило, инициируется клиентом. Соединение устанавливается в соответствии с параметрами, указанными в списке коммуникационных контактов, и подтверждается сервером. Начиная с этого момента, могут использоваться другие сервисы FMS. Соединение может быть разорвано любым из партнеров, в любое время.

Преимущества / недостатки

Вид коммуникаций, когда ПК обращается к FMS-объектам, обладает следующими преимуществами и недостатками:

Преимущества

- Являясь открытым протоколом, FMS может применяться для обращения к широкому кругу FMS-устройств.
- Сервер предоставляет только данные, сконфигурированные пользователем, неавторизованный доступ невозможен.
- Тип данных для объекта или переменной устанавливается на стороне сервера.
- Клиент может прочитать список переменных сервера, поэтому переменные не требуется конфигурировать на клиенте.
- Переменным могут быть даны символьные имена, и клиент может прочитать символьные имена из словаря объектов сервера.

Недостатки

- Интенсивность потока данных при таком протоколе меньше, по сравнению с протоколом DP, вследствие более дружественной поддержки пользователя.
- Для контроля переменных OPC-серверу необходимо их опрашивать через коммуникационную систему. В результате сеть постоянно нагружена.

2.7.6 Сервисы FMS для работы с переменными, ПК в качестве FMS-сервера

Описание сервисов

Как участник FMS-соединения, ПК также может использоваться в качестве FMS-сервера и предоставлять свои переменные. Другие партнерские устройства, например программируемые контроллеры, могут читать и записывать переменные, имеющиеся на ПК. Прикладная задача, работающая на ПК, также использует эти переменные, в основном, таким же образом, как и переменные партнерского устройства: операция записи изменяет значение локальной переменной. Когда партнер читает переменную, считывается ранее установленное значение.

Доступ к переменным осуществляется

- с помощью индекса (указателя) коммуникационного объекта или
- с помощью символьного имени коммуникационного объекта.

О том, какие данные должны быть доступны тому или иному клиенту, решение принимает пользователь на этапе конфигурирования. Должен быть создан словарь объектов, который содержит информацию о переменных (тип, начальное значение, права доступа). Эта информация размещается в словарь объектов FMS-сервера (в данном случае, ПК).

Преимущества / недостатки

Предоставление переменных персональным компьютером, выступающим в роли FMS-сервера, обладает следующими преимуществами и недостатками:

Преимущества

- Являясь открытым протоколом, FMS может использоваться для обращения к широкому кругу FMS-устройств.
- Отсутствует коммуникационная нагрузка, когда партнер не читает или не записывает данные.
- Если переменные сервера записываются партнерским устройством, измененные значения поступают в интерфейс OPC (если эти переменные контролируются).

Недостатки

- В контроллере должна присутствовать программа для чтения или записи переменных, что налагает определенные требования к CPU контроллера.
- В ПК должен быть сконфигурирован словарь объектов.
- Интенсивность потока данных при использовании этого протокола меньше, по сравнению с протоколом DP.

2.8 Протокол S7 для PROFIBUS

Введение

В этом разделе описаны принципы коммуникаций и сервисов протокола S7. В разделе также описана структура и конфигурация.

Коммуникационные сервисы

Протоколом S7 поддерживаются следующие коммуникационные сервисы:

Сервис	Описание
Информационные сервисы	Сведения о состоянии соединения. Отображение статуса устройства/пользователя коммуникационного партнера.
Сервисы для работы с переменными	Функции для чтения и записи одной или нескольких переменных.
Сервисы для работы с буферами передачи/приема	Управляемая программно передача больших блоков данных.
Сервисы для управления блоками	Эти сервисы обеспечивают загрузку, считывание, удаление и связывание блоков в программной последовательности ПЛК во время работы. Благодаря этому обеспечивается динамическое изменение программных последовательностей и параметров.
Сервисы для обработки событий	Эти сервисы предназначены для приема и обработки сообщений от программируемых контроллеров SIMATIC S7, например, тревог ALARM.
Сервисы для обеспечения безопасности	Установка паролей для доступа к объектам данных SIMATIC S7.

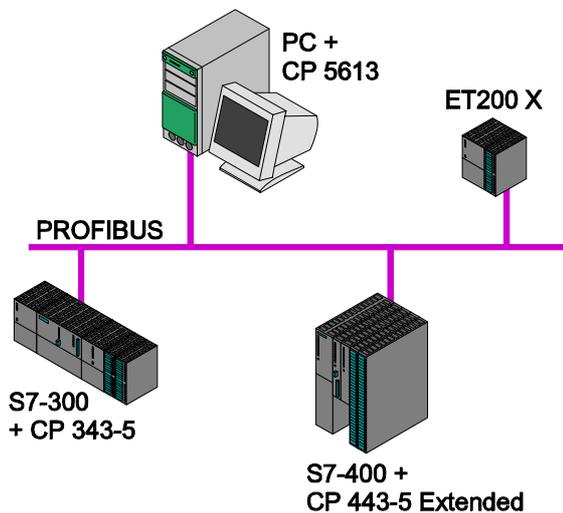
2.8.1 Типичная конфигурация системы S7

Введение

Протокол S7 служит для связи с программируемыми контроллерами (ПЛК) SIMATIC S7/M7. Он обладает следующими основными свойствами:

- Оптимизирован для коммуникаций SIMATIC
- Элементы протокола адаптированы к требованиям SIMATIC
- Большая скорость по сравнению с другими протоколами систем автоматизации, предназначенными для передачи данных.
- Доступен для шинных систем на верхнем уровне управления и уровне отдельной ячейки автоматизации, какими являются Industrial Ethernet и полевой уровень с PROFIBUS.

Пример конфигурации



2.8.2 Принципы S7-коммуникаций

ПК в роли клиента или сервера

Протокол S7 предоставляет простые и мощные коммуникационные сервисы, построенные по принципу "клиент-сервер". Обмен данными происходит между прикладной задачей автоматизации, работающей на станции SIMATIC PC (клиент) и другой прикладной задачей автоматизации (сервер). Данные предоставляются программируемым контроллером, поэтому такие станции являются серверами. Данные запрашиваются задачей, работающей на ПК, следовательно, ПК является клиентом. На этапе установления соединения, а также в случае сервисов для работы с буферами передачи/приема, ПК также может выступать в роли сервера.

Протокол S7 в сети PROFIBUS

Реализация протокола S7 в сети PROFIBUS базируется на сервисах FDL (уровень 2) и расширяет протокол специальными элементами от Siemens. Коммуникации, следовательно, предоставляют преимущества протокола, ориентированного на установление соединений, например, возможность проверки существования станции с контролем соединения.

S7-соединения

В рамках протокола S7 соединение описывается адресами партнеров в сети PROFIBUS и сведениями об адресе (например, точками доступа к сервисам). Эти свойства протокола S7 указываются в конфигурации.

На ПК S7-соединение описывается путем указания имени соединения, например, "S7_connection_1".

S7-соединения, которые определены и на стороне клиента, и на стороне сервера, называют "двухсторонними" соединениями. Соединения, описанные только на стороне клиента, например, соединение между ПК и станцией S7–300, известны как "односторонние".

В программируемом контроллере двухстороннее соединение идентифицируется локальным идентификатором (local ID), например, значением "1". Указывая локальный идентификатор, программы S7 могут использовать коммуникационное соединение с помощью функциональных блоков, таких как BRCV/BSEND.

Примечание

OPC-элементы, используемые на ПК для S7-коммуникаций в сети PROFIBUS, снабжаются префиксом "S7".

Установление соединения

На этапе установления соединения коммуникационные партнеры автоматически обмениваются между собой наиболее важными характеристиками коммуникационного тракта. Будут использоваться те параметры, которые могут быть обеспечены обоими коммуникационными партнерами.

Согласовываются следующие параметры:

- Размер пакетов данных на транспортном уровне (размер PDU)
- Количество разрешений на передачу (количество ресурсов для передачи и приема, используемых одновременно)

2.8.3 Конфигурирование системы S7

Введение

Для реализации связи с применением протокола S7 в сети PROFIBUS на этапе конфигурирования следует заранее описать коммуникационные соединения.

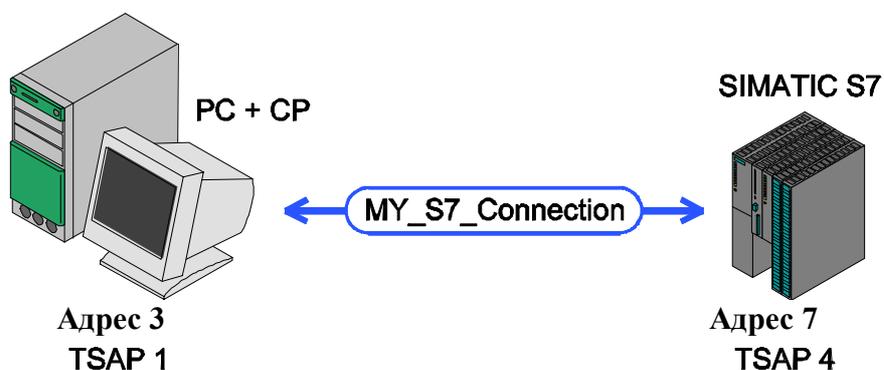
Указанные на этапе конфигурирования данные используются коммуникационными партнерами в режиме выполнения, и их не требуется больше указывать в программе пользователя. В случае физического переконфигурирования системы изменения должны быть внесены только в конфигурацию, программы менять не потребуется.

Адрес и точка доступа к сервисам

В конфигурации для коммуникационного S7-соединения определяются следующие параметры:

- имя соединения
- адрес партнера
- точка доступа к сервисам транспортного уровня (TSAP) для локальной станции
- точка доступа к сервисам транспортного уровня (TSAP) для удаленной станции
- способ установления соединения

Пример



Структура точки доступа к сервисам

Точка доступа к сервисам для S7-коммуникаций (TSAP) состоит из 2 байтов:

Байт 1

Номер коммуникационного ресурса, зарезервированного для этого соединения. Номер ресурса соединения для локальной и удаленной станций назначается в программе конфигурирования (STEP 7). Программа конфигурирования обеспечивает, чтобы в программируемом контроллере не было зарезервировано больше ресурсов, чем их реально существует.

Некоторые номера зарезервированы за специальными типами соединений и не контролируются:

- 01: PG-соединение (зарезервировано для использования программатора)
- 02: OS-соединение для устройств операторского управления и визуализации, для которых требуется постоянное соединение
- 03: Прочие соединения

Байт 2

Определяет положение CPU в программируемом контроллере:

- Биты 7 ... 5: Стойка (корзина)
- Биты 4 ... 0: Слот (гнездо)

(Особый случай: если обращение к устройству, включенному в сеть, происходит напрямую, байт 2 имеет нулевое значение)

Примечание

В случае использования OPC-сервера в конфигурации можно установить дополнительные свойства соединения:

- Время установления и разрыва соединения OPC-сервером.
 - Использование и конфигурирование сервисов обслуживания аварий для соединения.
 - Прочие детализируемые параметры
-

2.8.4 S7-сервисы для работы с переменными

Описание сервисов

Протокол S7 обеспечивает простой доступ к переменным S7 программируемого контроллера.

Переменные S7 – это, как правило, отдельные части S7-объекта, начинающиеся с определенного адреса памяти объекта, которым назначен тип данных. Например, переменная с адресом "DB 5,W10" обращается к байтам 10 и 11 блока данных 5 и содержит значение слова. Можно также составить массив переменных (несколько элементов одного типа в одной переменной).

Большинство S7-устройств поддерживает следующие объекты S7:

- блоки данных
- экземплярные блоки данных
- входы/выходы
- периферийные входы/выходы
- биты памяти
- таймеры
- счетчики

Преимущества / недостатки

Использование сервисов для работы с переменными обладает следующими преимуществами и недостатками:

Преимущества

- Очень простой доступ к партнерскому устройству без создания на нем программы.
- Оптимизированное чтение и запись нескольких переменных и длинных массивов переменных.
- В случае использования OPC-сервера отдельные переменные в целях безопасности можно защитить, назначив для них права доступа.
- В случае OPC возможно использовать символы из STEP 7.
- В случае использования OPC отсутствует ограничение размера переменных (в частности, массивов), связанное с размером PDU.
- Согласованность (целостность) данных в пределах всей длины PDU в случае использования новых CPU.

Недостатки

- Чтобы контролировать изменения переменных, необходимо циклически обращаться к партнерскому устройству.
- Обращение через короткие промежутки времени означает высокую нагрузку на сеть.
- Не гарантируется согласованность данных (данные считываются за несколько циклов программы ПЛК), если размер переменной превышает размер PDU.

2.8.5 S7-сервисы для работы с буферами передачи/приема

Описание сервисов

Буфер-ориентированные сервисы обеспечивают управляемую программно передачу больших блоков данных. Независимо от размера PDU, может быть передано до 65534 байтов "полезных" данных.

Чтобы был возможен обмен данными между ПЛК SIMATIC S7 и SIMATIC PC, на обеих сторонах должно быть сконфигурировано соединение. В отличие от сервисов для работы с переменными, конфигурация соединения должна быть загружена в ПЛК.

Чтобы можно было принимать блоки данных, должны быть предоставлены ресурсы для приема. Программа ПЛК содержит для этих целей функциональный блок "BRCV".

Примечание

Если используется OPC-сервер, ресурсы предоставляются путем создания переменной "RECEIVE".

Когда используется OPC-сервер, имеется возможность привязки блока данных или его частей к отдельным OPC-переменным, а также назначения этим частям определенных типов данных. Такая структуризация облегчает пользователю применение буфер-ориентированных сервисов.

Преимущества / недостатки

Применение буфер-ориентированных сервисов обладает следующими преимуществами и недостатками:

Преимущества

- Могут также передаваться большие (макс. 64 кбайт) блоки данных.
- Станция SIMATIC PC может быть как клиентом, так и сервером. Другими словами, в случае применения сервисов для работы с буферами передачи/приема данные также могут передаваться между двумя ПК с использованием протокола S7.
- Имеется возможность определения структуры блоков данных для OPC-элементов.
- Все переменные OPC, описанные внутри буфера приема, принимают сообщение об изменении, когда поступает блок данных.
- Целостность (согласованность) данных в пределах блока данных (все данные считываются за один цикл программы программируемого контроллера).
- Нагрузка на сеть из-за опроса отсутствует, если не передаются данные.

Недостатки

- На ПЛК должны быть запрограммированы блоки передачи и приема.
- Приемная сторона не может запросить данные, а должна ожидать, пока они не будут переданы.
- Сервисы для передачи/приема буферов доступны не для всех программируемых контроллеров S7.

2.8.6 S7-сервисы для управления блоками

Имеющиеся сервисы для управления блоками

Блок представляет собой загружаемую область программируемого контроллера.

В рамках протокола S7, в случае использования OPC-сервера, доступны следующие сервисы для управления блоками, которые запускаются программатором или ПК (SIMATIC PG или PC):

- Загрузка данных из PG/PC в CPU SIMATIC.
- Считывание данных из CPU SIMATIC в PG/PC.
- Связывание блоков в программную последовательность CPU SIMATIC.
- Удаление блоков.
- Сжатие памяти программируемого контроллера.

Типы блоков

Сервисы для управления блоками можно использовать для следующих типов блоков:

- организационные блоки (OB)
- функциональные блоки (FB)
- функции (FC)
- блоки данных (DB)

Управление блоками

Блок может быть считан из CPU SIMATIC в ПК с помощью S7-OPC приложения, или загружен из ПК в CPU SIMATIC. В ПК блоки хранятся в виде файлов.

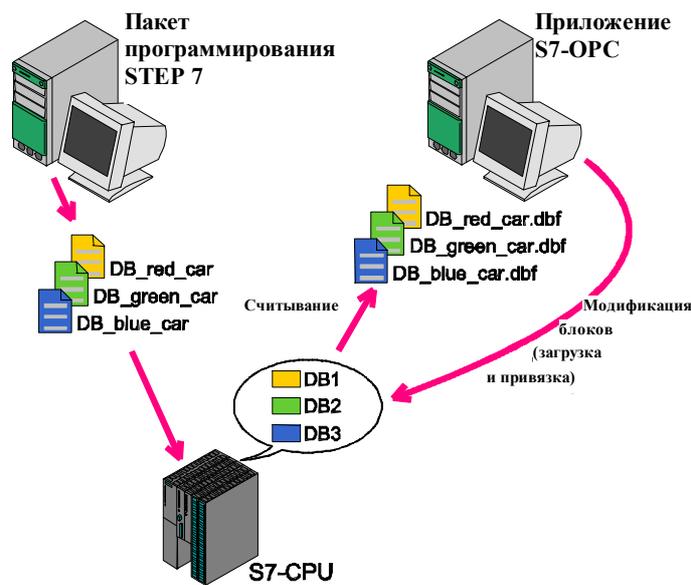
Имя блока в CPU S7 должно быть уникальным. Максимальный объем данных ограничивается в зависимости от конкретного CPU. По этой причине блоки делятся на отдельные сегменты, которые передаются последовательно друг за другом.

Связывание блоков

Блок, передаваемый программируемому контроллеру, размещается в буфер. Это означает, что для S7-программы блок еще не доступен. Хотя блок уже "виден" в списке блоков данных, который можно просмотреть с помощью online-функций STEP 7, блок открыть нельзя. Это становится возможным только после того, как блок включается в список активных блоков.

Пример применения сервисов для управления блоками

Блоки, запрограммированные или созданные с помощью STEP 7, загружаются в программируемый контроллер с помощью программатора на этапе ввода системы в эксплуатацию. Эти блоки (в нашем примере блоки DB_red_car.dbf, DB_green_car.dbf,...) сохраняются в память программы как DB1, DB2, Приложение S7-OPC может считать эти блоки во время работы и сохранить их локально в виде файлов DB_red_car.dbf, DB_green_car.dbf,.... Таким образом, ПК-контроллер может загружать и удалять блоки и влиять на выполнение программы, заменяя, например, блок DB1 (DB_red_car.dbf) другим блоком данных (DB_blue_car.dbf), который становится блоком DB1.



Примечание

Сервисы для управления блоками назначаются специальным переменным OPC. Запись значений в эти переменные приводит к запуску сервиса. В качестве параметров сервиса используются записанные значения.

2.8.7 S7-сервисы для обслуживания событий: тревоги

Описание сервисов

Для сигнализации событий могут применяться "тревоги". Тревоги должны быть запрограммированы, поскольку они инициируются S7-блоками, а не изменением состояния процесса.

Совместно с тревогой может быть передано 10 сопутствующих значений. Тревоги являются исключительными событиями, которые запускаются партнерским устройством. Тревоги размещаются в буфер и не могут быть утрачены.

В случае запрограммированных сообщений пользователь должен включить в программу S7 блок ALARM, ALARM_8, ALARM_8P или NOTIFY S7. Блок следит за изменением восьми сигналов, после чего передает кадр, содержащий значение времени и дополнительные значения.

Примечание

S7-тревоги могут обрабатываться только OPC-сервером, поддерживающим спецификацию Alarms & Events (Тревоги и события). Возможность назначения тревог переменным (OPC Date Access) не планируется.

Программирование тревог

Прежде чем использовать тревоги в соответствии с приведенным выше описанием, их необходимо запрограммировать. Используя сконфигурированный номер сообщения, можно обращаться к принятым тревогам с помощью OPC Alarms & Events.

Преимущества / недостатки

Преимущества

- Тревоги размещаются в буфер и не могут быть утрачены.
- Совместно с тревогой может быть передано 10 сопутствующих значений. Размер сопутствующих значений ограничивается только размером PDU.

Недостатки

- Для формирования тревог в контроллере должна быть создана программа.
- Для сопутствующих значений поддерживается только ограниченное количество типов данных.
- Могут использоваться только для OPC Alarms & Events.

2.8.8 Как программировать тревоги

Введение

S7–OPC-сервер может принимать тревоги, передаваемые программой S7. Тревоги инициируются блоком S7. В следующем разделе поясняется, каким образом можно программировать такие тревоги. Более подробные сведения можно найти в руководстве пользователя по системе STEP 7.

Последовательность действий

Шаг	Шаги
1	Выберите подходящий блок сообщения.
2	Выберите функциональный блок, которому должно быть назначено сообщение, и откройте его.
3	Заполните таблицу объявления переменных.
4	Запрограммируйте вызов блока сообщения в разделе объявления функционального блока.
5	Создайте тип сообщения, используя текстовые строки, атрибуты и средства отображения, в FB.
6	Назначьте экземплярные блоки данных функциональному блоку (FB) и измените их в соответствии с вашими требованиями.

The screenshot shows the SIMATIC Manager interface. At the top, a table lists variable declarations:

Address	Declaration	Name	Type	Initial value	Comment
0.0	in	MessageNumber	DWORD	DW#16#1	MessageNumber
4.0	in	Alarm_in	BOOL	FALSE	
	out				
	in_out				
6.0	stat	Alarmbaustein	"ALARM"		Declaration
	temp				

Below the table, the 'Network 1: Alarm call' is shown. It includes a comment field and a ladder logic diagram for the 'ALARM' block. The diagram shows the following connections:

- EN
- EN_R
- #Alarm_in — SIG
- ID
- #MessageNumber — EV_ID
- SEVERITY
- P#DB1.DBX 0.0 BYTE 20 — SD_1
- SD_2
- SD_3
- SD_4
- SD_5 — DONE
- SD_6 — ERROR
- SD_7 — STATUS
- SD_8 — ACK_DM
- SD_9 — ACK_UP
- SD_10 — ENO

2.8.9 S7-сервисы для обеспечения безопасности

Описание сервисов

Доступ для записи и чтения к CPU ПЛК может быть ограничен паролем. Пароль обладает бóльшим приоритетом по отношению к ключу-переключателю модуля CPU.

С помощью сервисов обеспечения безопасности пароль для соединения может быть передан с целью разрешения доступа и отмены уровня защиты. Для программируемого контроллера S7 с помощью пакета конфигурирования STEP 7 может быть активизировано три уровня защиты для сервисов управления блоками:

- Защита переключателем-ключом
- Защита от записи
- Защита от записи и чтения

Передав правильный пароль, можно отменить все перечисленные уровни защиты для соединения, через которое передается пароль.

Примечание

Для соединений, сконфигурированных на обоих концах, все сервисы могут выполняться даже, когда ключ находится в положении RUN (выполнение).

2.8.10 Информационные S7-сервисы

Описание сервисов

Протокол S7 предоставляет сервисы, с помощью которых

- могут опрашиваться атрибуты виртуального полевого устройства (VFD),
- может быть прочитан статус виртуального полевого устройства.

Примечание

OPC-сервер предоставляет не только информационные переменные для этих сервисов, но также переменные, предназначенные для указания статуса соединения и для регистрации сервисов обслуживания событий.

2.9 Промышленная связь через Industrial Ethernet

Введение

В этом разделе приводятся общие сведения о системе связи Industrial Ethernet.

2.9.1 Описание Industrial Ethernet

Введение

Industrial Ethernet – это мощная коммуникационная сеть, соответствующая международному стандарту IEEE 802.3 (Ethernet), специально предназначенная для работы в промышленных условиях.

Она обладает следующими основными свойствами:

- Объединяет в единую сеть различные уровни предприятия, например, офис и производственный цех.
- Надежность конструкции и электромагнитная устойчивость.
- Высокая скорость передачи даже при большом количестве узлов благодаря общедоступным сетевым компонентам, соответствующим стандарту Fast Ethernet (быстрая сеть Ethernet) и поддерживающим скорость 100 Мбит/с.
- Поддержка различных сред передачи (например, промышленная витая пара (ИТР), волоконно-оптический кабель).
- Возможность расширения системы с использованием технологии коммутации.
- Высокая степень надежности сетевых топологий с резервированием.

2.9.2 Протоколы и устройства для SIMATIC NET Industrial Ethernet

Обмен данными:

Область применения Industrial Ethernet ограничивается обменом данными. Для этого имеется целый ряд устройств и протоколов:

Протокол	Описание	Типичный участник связи от Siemens
S7-коммуникации	Встроенные, оптимизированные функции связи систем SIMATIC S7/M7 для широкого круга задач (TCP/IP с RFC 1006 или ISO).	Любой Ethernet-адаптер для ПК CP 1512, CP 1612 CP 1613 CP 343-1 CP 443-1 CP 443-1 IT
S5-совместимые коммуникации (SEND/RECEIVE)	Коммуникационные сервисы на базе транспортного протокола (TCP/IP с RFC 1006 или ISO) для обмена данными с устройствами S5 и S7.	Любой Ethernet-адаптер для ПК CP 1512, CP 1612 CP 1613 CP 343-1 (TCP) CP 443-1 CP 443-1 IT
Собственный протокол TCP/IP	Простые коммуникационные сервисы на базе TCP/IP для обмена данными с любым устройством, которое поддерживает протокол TCP/IP.	Любой Ethernet-адаптер для ПК CP 1512, CP 1612 CP 1613 CP 443-1

2.9.3 Методы доступа к шине в Industrial Ethernet

Обнаружение коллизий методом CSMA/CD

В Ethernet используется механизм доступа к шине CSMA/CD (коллективный доступ с контролем несущей и обнаружением коллизий). Такой метод предполагает, что станция, которая намеревается передавать данные, сначала "слушает", не передаются ли по каналу связи (по кабелю общей шины) данные другими станциями (контроль несущей), и передает данные, если кабель "свободен". Если кабель в настоящий момент используется другой станцией, узел отзывает свой запрос на передачу и предпринимает попытку передачи данных позже (коллективный доступ).

Если попытка обращения к шине завершается неудачей, передающий узел непрерывно "слушает линию" и приступает к передаче сразу же по завершению текущего сеанса связи. Это может привести к конфликту (коллизии) двух пакетов данных и их обоюдному разрушению. Для обнаружения такого конфликта (коллизии) узлы принимают сигналы при передаче данных. Если переданные данные отличаются от принятых данных, это является признаком коллизии, и в этом случае станция отзывает свой запрос и выдерживает паузу, длительность которой случайна, прежде чем вновь приступить к передаче. Количество коллизий ограничивается в случае применения технологии коммутации и дуплексного режима.

Такой метод доступа к среде передачи работает корректно лишь в том случае, когда время, необходимое для передачи пакета данных, в два раза превышает время распространения сигнала. Следовательно, протяженность сети определяется скоростью передачи и размером пакета данных.

Для реализации более протяженных сетей можно использовать коммутаторы и дуплексный режим передачи.

При использовании такого метода доступа к шине совершенно очевидно, что эффективная скорость передачи данных зависит от нагрузки на шину. Если нагрузка на шину слишком велика, в ней будут часто происходить коллизии. В результате, процент данных, передаваемых корректно, будет снижен.

Режимы

Предусмотрено два режима:

Полудуплексный: узел может либо передавать, либо принимать, и не может делать обе эти операции одновременно.

Дуплексный режим: узел может передавать и принимать в одно и то же время. Основным требованием к режиму дуплексной связи является наличие среды передачи с отдельными каналами для передачи и приема (например, волоконно-оптический кабель или ИТР).

2.9.4 Быстрая сеть Ethernet

Свойства технологии Fast Ethernet (быстрая сеть Ethernet)

Быстрая сеть Ethernet – это дальнейшее развитие технологии Ethernet. Стандарт IEEE 802.3u (Fast Ethernet), по существу, базируется на классическом стандарте Ethernet, за тем лишь исключением, что скорость передачи в нем увеличена в 10 раз и составляет 100 Мбит/с.

Ethernet и Fast Ethernet обладают следующими общими свойствами:

- формат данных
- управление доступом к каналу связи методом CSMA/CD

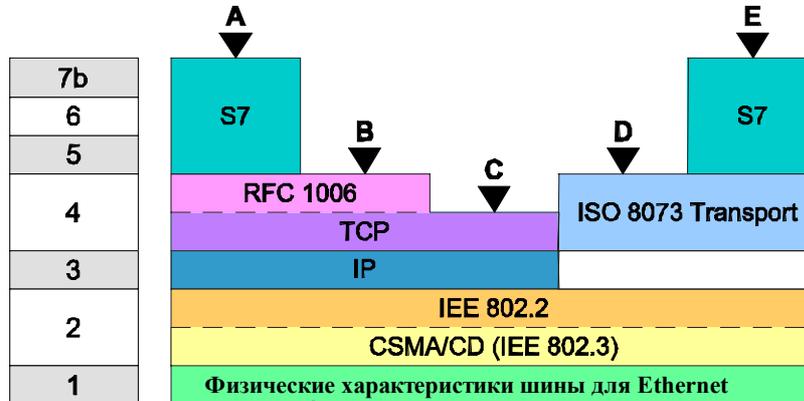
Они отличаются друг от друга в следующем:

- протяженность сети
- правила конструирования сетей
- автоматическое определение скорости передачи
- поддержка дуплексного режима.

2.9.5 Положение Industrial Ethernet в эталонной модели ISO/OSI

Введение

Базируясь на нескольких уровнях эталонной модели, Industrial Ethernet предоставляет несколько пользовательских интерфейсов:



▼ Интерфейс пользователя

Символ	Протокол	Описание
A, E	S7-коммуникации	Единый интерфейс пользователя для TCP/IP (A) и ISO (E) с применением функций S7
B, D	S5–совместимые коммуникации (SEND/RECEIVE)	Коммуникационные сервисы, базирующиеся на транспортном интерфейсе ISO для обмена данными с устройствами S5 и S7. В случае TCP/IP требуется адаптер (RFC 1006). Благодаря этому унифицируется интерфейс пользователя SEND/RECEIVE для TCP/IP (B) и ISO (D).
C	Собственный протокол TCP/IP	Простые коммуникационные сервисы на базе TCP/IP (C) для обмена данными с любым устройством, которое поддерживает протокол TCP/IP.

Примечание

Работающий на ПК OPC-сервер приводит все перечисленные пользовательские интерфейсы к единому (унифицированному) OPC-интерфейсу.

Уровень 1**Электрические характеристики**

Максимальное количество узлов	Не ограничивается, не более 1024 на один сегмент сети.
Скорость передачи	10 Мбит/с или 100 Мбит/с.
Протяженность сети	Электрическая сеть: 1.5 км Оптическая сеть, 10 Мбит/с: 4.5 км Коммутируемая сеть: в принципе не ограничена
Топология	Шина Дерево Резервированное кольцо Звезда
Среда передачи данных	(Промышленная) витая пара (ИТР) Триаксиальный кабель Волоконно-оптический (ВО) кабель

Механические характеристики

Подключение	15-контактное гнездо D-sub для подключения к AUI или промышленной витой паре Разъем RJ-45 для 10BaseT, 100BaseTX (витая пара)
-------------	--

2.9.6 Транспортные протоколы для Industrial Ethernet

Транспортный протокол ISO

Международный стандарт (ISO 8073, класс 4) описывает сервисы, предназначенные для передачи данных через соединения. Транспортный сервис ISO соответствует уровню 4 эталонной модели ISO.

Благодаря возможности "расщепления" данных на блоки (данные пользователя могут быть разбиты на несколько кадров на транспортном уровне ISO), транспортный сервис ISO может передавать большие объемы данных. Транспортный сервис ISO позволяет осуществлять обмен данными с любым коммуникационным партнером (например, ПЛК SIMATIC S5 или ПК), который поддерживает передачу и прием данных в соответствии с правилами транспортного протокола ISO.

Надежность передачи данных, например, в случае проблем с кабелем

Благодаря автоматическому повтору передачи данных, предусмотренному транспортным протоколом ISO, и дополнительным механизмам проверки блоков (например, проверка кода CRC на уровне 2) достигается высокая степень надежности передачи данных. Для подтверждения приема данных транспортный сервис ISO коммуникационного партнера высылает подтверждение.

ISO на TCP (RFC1006)

Сервис "ISO на TCP" соответствует стандарту TCP/IP (Протокол управления передачей данных через межсетевой протокол (IP)) с расширением RFC 1006 в соответствии с уровнем 4 эталонной модели ISO. Расширение RFC 1006 необходимо, поскольку протокол TCP не предусматривает разбиение передаваемого потока данных на отдельные сообщения. Эта проблема решается протоколом ISO уровня 4 путем применения кода EOM (Завершение сообщения). Код EOM позволяет передавать сообщения в виде отдельных блоков данных. TCP/IP не распознает этот код. Расширение протокола необходимо для получения возможности передачи сообщений. Расширение RFC 1006 описывает, каким образом сервисы уровня 4 ISO "вписываются" в протокол TCP. RFC 1006 – это официальный стандарт, который применяется многими производителями оборудования.

Сервис "ISO на TCP" обеспечивает связь с любым коммуникационным партнером (например, персональным компьютером или системами других производителей), который поддерживает передачу и прием данных в соответствии с протоколом "ISO на TCP".

Надежность передачи данных, например, в случае проблем с кабелем

Благодаря автоматическому повтору передачи данных и дополнительным механизмам проверки блоков (проверка кода CRC на уровне 2) достигается высокая степень надежности передачи данных. Факт получения данных подтверждается коммуникационным партнером путем передачи подтверждения.

Собственный протокол TCP/IP (без RFC 1006)

Стандарт TCP/IP (Протокол управления передачей данных через межсетевой протокол) без расширения RFC 1006 позволяет обратиться к контроллеру S7 любому коммуникационному партнеру, использующему TCP/IP. Поскольку транспортный уровень протокола TCP/IP не предполагает фрагментирование потока данных, задача разбиения данных на блоки лежит на пользователе. Обоим участникам коммуникационного соединения должен сообщаться размер передаваемого пакета, чтобы пакет был извлечен из потока данных без ошибок.

Надежность передачи данных, например, в случае проблем с кабелем

Благодаря механизмам проверки блоков (проверка кода CRC на уровне 2) достигается высокая надежность передаваемых данных. Факт приема данных подтверждается коммуникационным партнером путем передачи подтверждения. Надежная связь возможна лишь в том случае, если размеры буфера согласовываются между коммуникационными партнерами.

Примечание

Коммуникационные функции на базе RFC 1006, ISO и TCP в OPC-сервере унифицированы и предоставляются посредством функций "SEND/RECEIVE".

2.10 S5–совместимые коммуникации (SEND/RECEIVE)

Введение

S5–совместимые коммуникации для Industrial Ethernet обеспечивают обмен данными по сети Ethernet между контроллерами SIMATIC S5, контроллерами SIMATIC S7, персональными компьютерами и рабочими станциями.

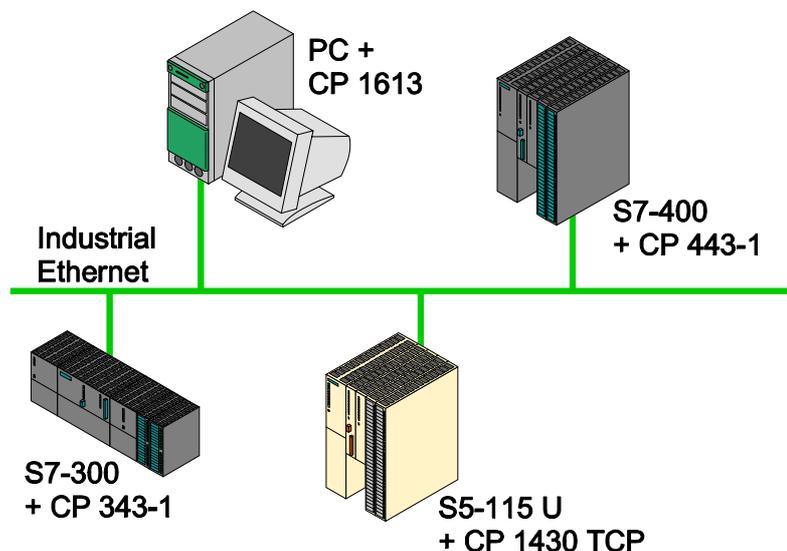
В этом разделе описаны принципы коммуникаций и сервисов протокола SEND/RECEIVE. Также рассматривается структура и конфигурация.

2.10.1 Типичная конфигурация системы SEND/RECEIVE

Введение

Для обмена данными с использованием S5–совместимых коммуникаций фирма Siemens предлагает коммуникационные модули для контроллеров семейства SIMATIC S5, SIMATIC 505 и SIMATIC S7, а также для персональных компьютеров и рабочих станций.

Пример конфигурации



2.10.2 Принципы коммуникаций SEND/RECEIVE

Транспортные протоколы

В отличие от S5–совместимых коммуникаций, осуществляемых через PROFIBUS, S5–совместимые коммуникации в Industrial Ethernet базируются на транспортном уровне эталонной модели ISO/OSI. Благодаря этому пользователь получает такие функции транспортного уровня, как соединения, управление потоком и фрагментирование данных.

S5–совместимые коммуникации могут применяться в Industrial Ethernet с использованием следующих транспортных протоколов:

- ISO
- TCP/IP с RFC 1006
- TCP/IP без RFC 1006

S5–совместимые коммуникации ориентированы на установление соединения. Данные могут передаваться только после установления транспортного соединения с партнерским устройством. Одновременно можно использовать несколько соединений с партнерским устройством.

Установление соединения инициируется одним из коммуникационных партнеров. Партнер А, являющийся инициатором установления соединения, отправляет запрос на установление соединения другому устройству В. Такое соединение, с точки зрения партнера А, является активным. Если соединение с точки зрения партнера А является пассивным, партнер А ожидает поступления запроса на установление соединения от устройства В. Сторона, отвечающая за установление соединения, указывается в конфигурации соединения.

Типы блоков/типы заданий

Для S5–совместимых коммуникаций через Industrial Ethernet используются блоки обработки или функциональные блоки программируемого контроллера. Для этих целей предусмотрены следующие типы блоков или типы заданий:

Название блока/задания	Описание
SEND (без функции WRITE)	Передача буфера данных партнеру
RECEIVE	Прием буфера данных от партнера
SEND (с функцией WRITE)	Запись в объект/переменную партнера
FETCH	Чтение объекта/переменной партнера

Примечание

Во время связи с устройствами S5 в пределах соединения можно использовать задания только одного типа. Одновременное использование соединения для заданий SEND и RECEIVE или FETCH приведет к тому, что контроллер SIMATIC S5 разорвет соединение.

Помимо сервисов для передачи буферов данных (SEND/RECEIVE), S5–совместимые коммуникации также предоставляют сервисы "Fetch" (Прочитать) и "Write" (Записать) через Industrial Ethernet, благодаря которым возможен прямой доступ к объектам коммуникационного партнера.

С помощью SEND и RECEIVE осуществляется обмен данными между блоком SEND и блоком RECEIVE. Связь инициируется всегда блоком SEND. Для двунаправленного обмена данными на обеих сторонах должны вызываться и блок SEND, и блок RECEIVE.

Если в качестве задания выбран тип FETCH (соответствует функции READ в S5), в этом случае в задании на передачу можно предусмотреть параметры источника данных, которые будут использоваться коммуникационным процессором S5 для непосредственного чтения данных из CPU и их отсылки запросившей стороне. С помощью заданий такого типа возможно прямое обращение к таким объектам, как входы, выходы, блоки данных, биты памяти (флаги) и счетчики.

2.10.3 Конфигурация SEND/RECEIVE

Параметры соединения

Прежде чем приступить к использованию S5–совместимых коммуникаций через Industrial Ethernet, необходимо сконфигурировать коммуникационные соединения. Для каждого соединения устанавливаются следующие параметры:

- Имя соединения
- Тип соединения (тип задания)
- Адреса партнерских станций
- Идентификатор точек доступа к сервисам на транспортном уровне (точки TSAP)
- Управление установлением соединения (активное/пассивное соединение)

2.10.4 Сервисы для работы с буферами SEND/RECEIVE

Описание сервисов

В рамках S5–совместимых коммуникаций предоставляются сервисы для передачи буферов данных. Передача пакета данных от передающего узла принимающему узлу должна инициироваться явным образом передающим узлом. Например, в ПЛК SIMATIC S7 это осуществляется путем вызова блока "AG_SEND" .

Максимальный размер буфера данных составляет 4 кбайт.

Принимающий узел не может инициировать передачу данных от передающего узла. Принимающий узел, тем не менее, должен быть готов к приему пакетов, предоставив соответствующие ресурсы для приема данных.

Примечание

Передача пакета персональным компьютером инициируется путем записи в переменную "SEND". Для приема блоков данных в соответствующем соединении должна быть сконфигурирована переменная "RECEIVE".

В случае использования OPC-сервера имеется возможность структурирования блока данных. Блок данных можно разбить на отдельные части, назначив каждой из них свой тип данных и используя их в качестве переменных. Тем не менее, с точки зрения сети всегда происходит чтение или запись целого блока данных.

Преимущества / недостатки

Коммуникации с использованием сервисов для работы с буферами, предусмотренные протоколом SEND/RECEIVE, обладают следующими достоинствами и недостатками:

Достоинства

- OPC-сервер позволяет передавать большие блоки данных размером до 64 кбайт.
- Отсутствует нагрузка на сеть, поскольку передача данных не происходит, если она не инициируется из программы пользователя (отсутствует опрос)
- Применение OPC-сервера позволяет структурировать адресацию к блокам данных.
- Возможна связь с устройствами S5 и S7, а также с ПК.

Недостатки

- Принимающая сторона не может инициировать передачу данных. Она должна ожидать, пока данные не будут переданы передающей стороной.
- Данные должны размещаться в буфере или быть скопированы в буфер программой пользователя на партнерском устройстве.

2.10.5 Сервисы для работы с переменными SEND/RECEIVE

Описание сервисов

Применение S5–совместимых коммуникаций через Industrial Ethernet обеспечивает простой доступ к символьным переменным программируемого контроллера S5 или S7. В этом случае используются сервисы "FETCH" (Прочитать) и "WRITE with SEND function" (Записать с помощью функции SEND (Передать)).

Когда выполняется сервис "FETCH" , на партнерское устройство отправляется задание, которое в качестве входных параметров содержит сведения о запрашиваемых переменных. Сторона, принявшая задание, анализирует входные параметры и возвращает текущее содержимое запрошенной переменной, а также подтверждение задания.

В составе задания на сервис "WRITE with SEND function" передается не только входной параметр, но также значение, которое должно быть записано в переменную адресуемого устройства. Принимающая сторона анализирует входной параметр и записывает значение в указанную переменную. После этого возвращается подтверждение выполнения задания.

В общем случае, переменные являются частями объекта S7 или S5, располагаясь в памяти объекта, начиная с указанного адреса. Переменным назначается определенный тип данных. Например, переменная с адресом "DB 5,B10" соответствует байту 10 блока данных 5, и ее содержимое эквивалентно слову. Также имеется возможность формирования массива переменных (несколько элементов одинакового типа в одной переменной).

Большинством устройств S5 и S7 поддерживаются следующие типы объектов:

- блоки данных
- входы/выходы
- периферийные входы/выходы
- биты памяти
- таймеры
- счетчики
- системная область
- расширенные входы/выходы

Преимущества / недостатки

Коммуникации с использованием сервисов для работы с переменными, предусмотренных протоколом SEND/RECEIVE, обладают следующими достоинствами и недостатками:

Достоинства

- Гибкий доступ к данным
- Возможна автоматическая оптимизация OPC- сервером

Недостатки

- Меньшая интенсивность потока данных по сравнению с сервисами для работы с буферами (SEND/RECEIVE).
- Для контроля изменений переменной необходимо циклически опрашивать партнерское устройство.

2.11 Протокол S7 для Industrial Ethernet

Введение

Протокол S7 служит для связи с программируемыми контроллерами SIMATIC S7/M7.

В этом разделе описываются принципы коммуникаций и сервисы протокола S7. В разделе также поясняется структура и конфигурация системы.

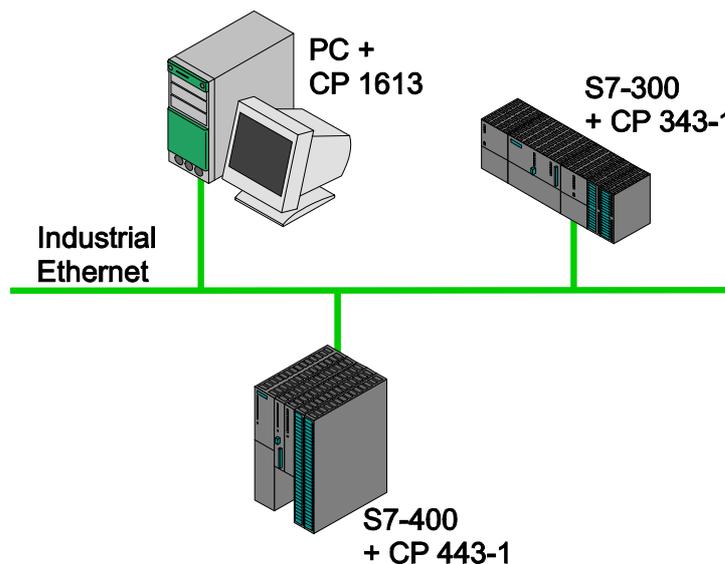
2.11.1 Типичная конфигурация системы S7

Введение

Протокол S7 служит для обмена данными с программируемыми контроллерами (ПЛК) SIMATIC S7/M7. Он обладает следующими основными свойствами:

- Оптимально подходит для коммуникаций в среде SIMATIC
- Элементы протокола адаптированы к требованиям SIMATIC
- Более высокая скорость связи по сравнению с другими протоколами, предназначенными для систем автоматизации.
- Доступен для шинных систем верхнего уровня управления и уровня отдельной ячейки автоматизации, например, Industrial Ethernet, а также для полевого уровня на базе PROFIBUS.

Пример



2.11.2 Принципы S7-коммуникаций

Характеристики протокола S7

Протокол S7 предназначен для обмена данными с программируемыми контроллерами (ПЛК) SIMATIC S7/M7. Он обладает следующими основными свойствами:

- Оптимально подходит для коммуникаций в среде SIMATIC.
- Элементы протокола адаптированы к требованиям SIMATIC.
- Большая скорость связи по сравнению с другими протоколами, предназначенными для систем автоматизации.
- Подходит для шинных систем верхнего уровня управления и уровня отдельной ячейки автоматизации, например, Industrial Ethernet, а также для полевого уровня на базе MPI/PROFIBUS.

Протокол S7 предоставляет простые и производительные коммуникационные сервисы на основе модели "клиент – сервер". Обмен данными происходит между прикладной задачей автоматизации, работающей на станции SIMATIC PC (клиент) и программируемыми контроллерами SIMATIC (сервер). Данные предоставляются программируемым контроллером, следовательно, эти станции являются серверами. Данные запрашиваются приложением, работающим на ПК, следовательно, ПК является клиентом. Во время установления соединения, а также при работе с сервисами для буфера передачи/приема ПК может также выступать в роли сервера.

Протокол S7 в сетях Industrial Ethernet

Реализация протокола S7 в сети Industrial Ethernet базируется на ISO-совместимом интерфейсе уровня 4 (транспортный уровень) эталонной модели ISO/OSI, дополняя уровни 5-7 элементами специальными элементами Siemens. Если в качестве транспортного протокола используется TCP/IP, необходимо наличие адаптера (RFC 1006).

S7-коммуникации предоставляют преимущества протокола, ориентированного на установление соединения, например, проверку существования станции и мониторинг соединения.

S7-соединения

Для соединения, предназначенного для протокола S7, указываются MAC-адрес (Ethernet-адрес) или TCP/IP-адреса партнеров, а также информация об адресах (например, точки доступа к сервисам). Эти свойства указываются для протокола S7 в конфигурации.

Аналогично протоколу FMS, протоколом S7 описываются "Виртуальные полевые устройства" (VFD). Вид VFD имеют только ПК-станции. В пределах VFD формируются группы соединений, которые могут использоваться только одним приложением. В рамках протокола S7 роль VFD выполняет приложение. Чтобы можно было управлять несколькими приложениями на одной ПК-станции с использованием S7-коммуникаций, необходимо наличие нескольких VFD.

В пределах ПК S7-соединение идентифицируется именем соединения, например, "S7_connection_1".

Примечание

Следует помнить, что OPC-сервер является S7-приложением. Если несколько OPC-клиентов обращаются к OPC-серверу, необходимо наличие только одного VFD!

S7-соединение, описанное как на стороне клиента, так и на стороне сервера, называется "двухсторонним" соединением. Соединение, описанное только на стороне клиента, например, соединение между ПК и станцией S7-300, называют "односторонним".

В программируемом контроллере S7-соединение идентифицируется локальным идентификатором (local ID), например, значением "1". Указывая локальный идентификатор, программы S7 могут использовать коммуникационное соединение посредством таких функциональных блоков, как BRCV/BSEND.

В процессе установления соединения коммуникационные партнеры автоматически согласовывают между собой наиболее важные характеристики канала связи. Используются те параметры, которые поддерживаются обоими коммуникационными партнерами.

Согласовываются следующие параметры:

- Размер пакетов данных на транспортном уровне (размер PDU)
- Количество запросов на передачу и прием (количество одновременно используемых ресурсов для передачи и приема данных)

2.12 Отказоустойчивые S7-соединения

Введение

Надежность коммуникаций можно повысить путем резервирования канала связи, путем дублирования компонентов (например, CPU, CP или сети) или путем дублирования всех сетевых компонентов.

В этом разделе описаны типы отказоустойчивых соединений, а также указаны нюансы, о которых следует помнить при конфигурировании таких соединений.

2.12.1 Свойства отказоустойчивых S7-соединений

Принцип работы

Стандартные S7-соединения

Стандартное S7-соединение устанавливается через нерезервированный канал связи (CP – сеть – CP).

Отказоустойчивые соединения

Отказоустойчивое соединение состоит, по меньшей мере, из двух резервных каналов связи.

Механизмы контроля и синхронизации обеспечивают автоматическое включение пассивного (резервного) канала связи в случае сбоя активного (резервированного) канала связи. Само соединение при этом не разрывается.

Обнаружение обрыва кабеля

Время, за которое происходит обнаружение обрыва кабеля, не превышает одну секунду при активном трафике данных в этом канале связи.

После того, как обрыв кабеля обнаружен, вступают в работу те части протокола, которые специально предназначены для отказоустойчивой связи. В зависимости от количества соединений, являющихся рабочими и используемыми на ПК, эта операция может занять больше одной секунды.

Повышение степени резервирования

Степень резервирования отказоустойчивого соединения можно повысить, увеличив количество коммуникационных процессоров (CP) и количество используемых сетей.

Можно сконфигурировать следующие резервированные соединения S7-REDCONNECT:

- Отказоустойчивые соединения с двумя каналами
- Отказоустойчивые соединения с четырьмя каналами
("Enable max. CP redundancy (with 4 connection paths)" = "Задействовать макс. резервирование CP (с 4 каналами связи)").

2.12.2 Отказоустойчивые S7-соединения с двумя каналами связи

Свойства

Отказоустойчивое S7-соединение использует два канала связи. Сбой одного из компонентов приводит к автоматическому переключению на резервный канал связи.

Пример

Система с двумя каналами связи может быть реализована, например, на базе следующих компонентов:

- Станция SIMATIC H, две стойки с CP в каждой
- Две сети
- ПК-станция с двумя CP

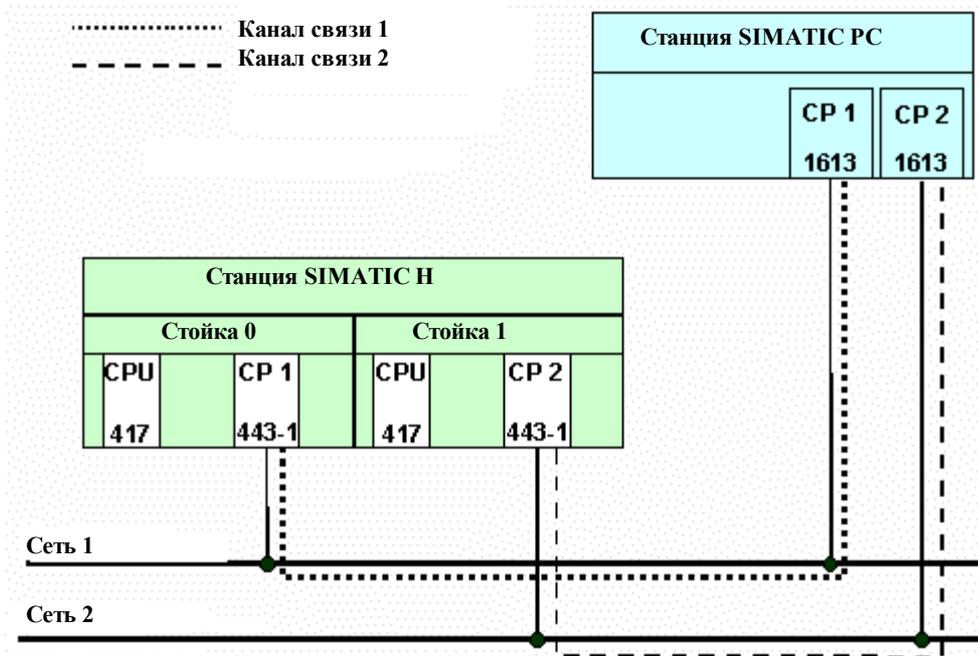


Рисунок 2–1 Пример системы с дублированием каналов связи между станцией SIMATIC S7 и ПК-станцией

2.12.3 Отказоустойчивые S7-соединения с четырьмя каналами связи

Свойства

Отказоустойчивое S7-соединение использует четыре канала связи. Предположим, что вышли из строя стойка 0 и сеть 2. Поскольку имеется четыре канала связи, автоматически включатся в работу стойка 1 и сеть 1 (см. пример ниже).

Пример

Отказоустойчивую систему с повышенным резервированием на базе четырех каналов связи можно создать, например, на базе следующих компонентов:

- Станция SIMATIC H, две стойки с двумя CP в каждой
- Две сети
- ПК-станция с двумя CP

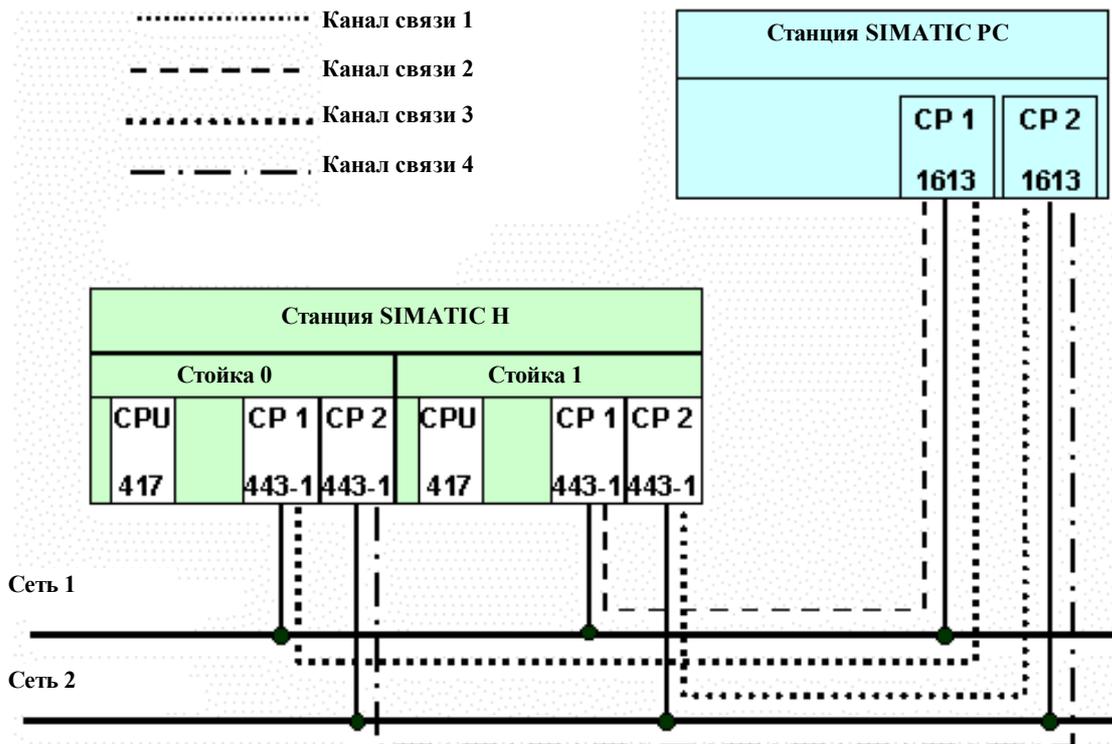


Рисунок 2-2 Пример резервирования с четырьмя каналами связи между станцией SIMATIC S7 и ПК-станцией

2.12.4 Что следует помнить при конфигурировании отказоустойчивых соединений

Требования

Чтобы ПК мог использовать соединение с отказоустойчивой системой SIMATIC S7, его необходимо снабдить специальными данными, например, адресами узлов шины, участвующих в соединении, данными о применяемых сетевых адаптерах и т.п.

Эта информация формируется автоматически, если выполняются три следующих требования:

- В STEP 7 (версия 5 или выше) сконфигурирован проект S7.
- ПК включен в проект в качестве "SIMATIC PC Station".
- Коммуникационные партнеры находятся в пределах одной и той же подсети.

Конфигурирование соединения с четырьмя каналами связи

Если сеть конфигурируется с помощью STEP 7 и необходимо повысить степень резервирования за счет использования четырех каналов связи, в этом случае в "NetPro" (Properties - Fault-Tolerant S7 Connection) (Свойства – Отказоустойчивое S7-соединение) необходимо установить флажок "Enable max. CP redundancy (with 4 connection paths)" (Задействовать макс. степень резервирования CP (с четырьмя каналами связи)).

Конфигурирование CP 443–1

В случае конфигурирования CP 443–1 с помощью STEP 7 необходимо помнить о следующем:



Предостережение

Если в станцию SIMATIC S7 установлен коммуникационный процессор CP 443-1, должен быть установлен флажок "Activate fast switchover of the connections" (Активизировать быстрое переключение соединений) (Properties CP 443-1 dialog > Options > Ethernet Profile for Fault-Tolerant Connections > Activate fast switchover of the connections) (Диалоговое окно "Свойства CP 443-1" > Дополнительные возможности > Профиль Ethernet для отказоустойчивых соединений > Активизировать быстрое переключение соединений).

2.13 Протокол SNMP

Введение

SNMP (Простой протокол управления сетью) – это открытый протокол для управления сетями, базирующийся на UDP (Протокол датаграмм пользователя). С помощью SNMP данные со всей сети можно собирать и анализировать на центральном узле. Этот протокол также позволяет настраивать параметры устройств. OPC SNMP-сервер в составе SIMATIC NET поддерживает протокол SNMPv1 (Протокол SNMP версии 1.0).

Информация о свойствах SNMP-совместимых устройств хранится в файлах MIB (MIB = База управляющей информации). Сервер OPC не может обрабатывать эти данные непосредственно, поэтому необходимо наличие компилятора базы MIB (MIB Compiler), который поставляется в комплекте с NCM. MIB Compiler обрабатывает только те файлы MIB, которые соответствуют SMIv1 (SMI версии 1.0; SMI = Структура управляющей информации).

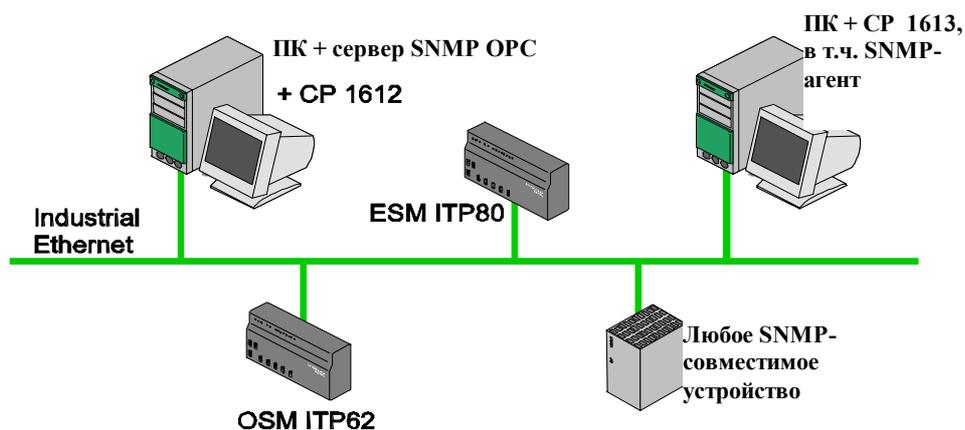
Более подробную информацию о MIB и SMI можно найти в следующей книге:

Межсетевой обмен с помощью TCP/IP
Douglas E. Comer
Prentice Hall

2.13.1 Типичная конфигурация системы для применения протокола SNMP

Пример

На следующем рисунке показан пример типичной конфигурации системы:



2.13.2 Принципы коммуникаций с использованием протокола SNMP

Протокол

SNMP базируется на протоколе UDP (Протокол датаграмм пользователя), не ориентированного на установление соединений.

Чтение и изменение значений переменных

С помощью сервисов для работы с переменными OPC-сервер передает запросы партнерским устройствам, а партнерские устройства на них отвечают.

Прерывания

Также предусмотрены сообщения, передаваемые устройством в случае возникновения определенных событий, без запроса со стороны SNMP OPC-сервера. Такие сообщения называют "SNMP-прерываниями" (SNMP trap). Они позволяют устройству сообщать об изменениях состояний переменных SNMP OPC-серверу.

2.13.3 Конфигурирование

Введение

Если вам необходимо использовать протокол SNMP в сети Industrial Ethernet, вам требуется однозначно описать конфигурацию системы. В этом разделе описываются параметры, которые должны быть настроены.

Для конфигурирования протокола SNMP в программе NCM предусмотрено следующее диалоговое окно:

Параметры протокола SNMP

Name (Имя)

Введите имя устройства или абонента, идентифицирующее его в соответствии с технологическими функциями, которые оно выполняет. Длина имени не должна превышать 255 символов.

IP address (IP-адрес)

Введите в этом поле IP-адрес абонента. IP-адрес состоит из 4-х десятичных чисел в диапазоне 0 ... 255, отделенных друг от друга точкой (например, 141.80.0.16).

Device profile (Профиль устройства)

Профиль устройства описывает структуру информации устройства, которая передается посредством SNMP. Можно выбрать либо существующий профиль, либо, в случае необходимости, создать новый профиль SNMP.

Для устройств, не поддерживающих SNMP, следует выбрать "No SNMP" (Не использовать SNMP).

Профили устройств содержат объекты MIB, выбранные компилятором MIB (MIB Compiler), с информацией об OPC-элементах, соответствующих этим MIB-объектам. Профили устройств – это текстовые файлы. В проекте STEP 7 файлы профилей размещаются в специальной папке (s7data/Snmp/Profile).

Для перечисленных ниже модулей профили устройств имеются на диске SIMATIC NET Software CD:

CP 1613 --> MIBII_V10.txt (поддерживает только объекты MIBII)

OSM --> Profil_OSM_V10.txt

ELS --> Profil_ELS_TP40_V10.txt

Эти файлы находятся в следующей папке:

<installationdrive>\SIEMENS\SIMATIC.NCM\S7data\SNMP\Profile

Кнопка "Create Profile" (Создать профиль)

Щелчок по этой кнопке приводит к запуску компилятора MIB (MIB Compiler). Подробное описание работы с этой программой можно найти в интерактивной справке программы MIB Compiler.

Community (Сообщество)

С помощью этого параметра устанавливаются права доступа. По умолчанию выбраны параметры **Public** (только чтение) или **Private** (чтение и запись). На этапе конфигурирования задаются символьные строки, выполняющие ту же роль, что и пароли. В этом поле следует ввести строку, которая совпадает со строкой описания сообщества, хранящейся у SNMP-агента.

Предусмотрено две строки описания сообществ. Одна назначена для сообщества с правом на чтение, а вторая – для сообщества с правом на запись. OPC-сервер считывает строку, хранящуюся в SNMP-устройстве, и разрешает доступ только в том случае, если строки совпадают.

Если в этом поле будет введена строка, соответствующая сообществу с правом на чтение, можно будет передавать запросы на чтение указанному устройству. Если вводится строка, соответствующая сообществу с правом на запись, указанному устройству можно будет передавать запросы и на чтение, и на запись.

Timeout (Превышение времени)

С помощью этого параметра указывается максимальное время отклика, в течение которого SNMP OPC-сервер должен ответить на запрос со стороны OPC-клиента.

- Значение устанавливается в миллисекундах (мс)
- Диапазон значений: 50 ... 60000 мс
- Значение по умолчанию: 9000 мс (9 с)

Опция "Optimization of SNMP access"
(Оптимизация доступа к SNMP-агенту)

В этом поле можно увидеть, будет ли оптимизироваться доступ (чтение) к SNMP-агенту. Чтобы свести к минимуму нагрузку на сеть и оптимизировать время передачи, доступ можно оптимизировать таким образом, чтобы за один сеанс передавалось как можно больше значений.

Такой тип доступа возможен только в том случае, когда SNMP-агент поддерживает такой тип передачи. Это выполняется для большинства SNMP-совместимых устройств, поэтому по умолчанию выбрана оптимизация доступа.

sysContact (Контактное лицо)

В этом параметре можно ввести имя контактного лица, отвечающего за обслуживание устройства. В этом поле также следует ввести информацию о том, как можно связаться с этим человеком. Для SNMP-совместимых устройств этот параметр нельзя ввести, поскольку он вводится с помощью инструментов, предназначенных именно для этого устройства, и хранится в самом устройстве.

Данные о контактном лице не должны содержать больше 255 символов.

sysLocation (Расположение)

Это поле предназначено для информации о расположении управляемого устройства, например, о номере комнаты. Для SNMP-совместимых устройств этот параметр нельзя ввести, поскольку он вводится с помощью инструментов, предназначенных именно для этого устройства, и хранится в самом устройстве.

Данные о местонахождении не должны содержать больше 255 символов.

sysName (Описание)

В этом параметре содержится описание устройства в виде простого текста. Информация должна содержать полное имя и версию оборудования, операционной системы или внутренней прошивки, а также сетевого программного обеспечения. Для SNMP-совместимых устройств этот параметр нельзя ввести, поскольку он вводится с помощью инструментов, предназначенных именно для этого устройства, и хранится в самом устройстве.

Описание устройства не должно содержать больше 255 символов

Comment (Комментарий)

В этом поле можно ввести дополнительные сведения об устройстве.

Комментарий не должен содержать больше 255 символов.

2.13.4 SNMP-прерывания

Введение

Прерывания (trap) – это сообщения, которые могут передаваться OPC-серверу без запроса со стороны последнего. В каждом SNMP-совместимом устройстве предусмотрено семь базовых прерываний. Также каждое устройство обладает специальными прерываниями, которые описаны в файле MIB.

Базовые (общие) прерывания

warmStart

Передается после "горячего" перезапуска устройства.

coldStart

Передается после "холодного" перезапуска устройства.

linkDown

Передается, когда соединение с устройством разрывается.

linkUp

Передается, когда соединение с устройством устанавливается.

authenticationFailure

Передается, когда происходит неавторизованное обращение к устройству.

egpNeighborLoss

EGP-сосед (EGP = Протокол внешнего шлюза) устройства не работает.

Протокол внешнего шлюза используется для обмена информацией о маршрутизации между двумя соседними станциями-шлюзами.

enterpriseSpecific

Передается, когда требуется передача специального прерывания устройства.

2.14 Сервисы для работы с буферами и OPC-интерфейс

Введение

В этом разделе описано применение буфер-ориентированных сервисов совместно с OPC. В разделе поясняется, каким образом выполняется привязка буферов данных к переменным OPC. В разделе также содержатся указания, которые следует учитывать в случае использования собственного протокола TCP/IP совместно с буфер-ориентированными сервисами.

2.14.1 Характеристики буфер-ориентированных коммуникаций

Поддерживаемые сервисы

В случае использования буфер-ориентированных сервисов между передающим и принимающим узлами через коммуникационную систему передаются буферы данных.

Буфер-ориентированные сервисы поддерживаются следующими протоколами:

- S7-коммуникации (BSEND/BRECEIVE)
- S5-совместимые коммуникации через Ethernet (SEND/RECEIVE)
- S5-совместимые коммуникации через PROFIBUS (SDA, SDN/Индикация)

Характерной чертой буфер-ориентированных сервисов является то, что данные передаются только, если передача инициируется передающим узлом. Принимающий узел не может инициировать передачу данных.

Действия передающего и принимающего узлов при обмене буферами данных

Передающий узел

- Объединяет воедино передаваемый буфер и его содержимое.
- Передает буфер через соединение коммуникационному партнеру.
- Принимает подтверждение результата передачи данных.

Принимающий узел

- Подготавливает буфер приема для соединения.
- Получает уведомление о том, что партнер передает ему буфер данных.
- Анализирует полученные данные.

2.14.2 Назначение буферов данных переменным OPC

Различия между элементами для передачи и приема

Интерфейс OPC Data Access (Доступ к данным через OPC) распознает только переменные процесса. Чтобы преимущества буфер-ориентированных сервисов можно было использовать для OPC, буферы должны быть сопоставлены (назначены) элементам OPC:

Элемент для передачи (S7-коммуникации: "BSEND", S5-совместимые коммуникации: "Send")

- OPC-элемент представляет буфер передачи или часть буфера передачи.
- Когда происходит запись в OPC-элемент (синхронно/асинхронно), в сеть выставляется задание на запись.
- Если несколько элементов, представляющих часть буфера, записываются одновременно с помощью групповой операции, буфер передачи составляется из всех частей, которые сначала объединяются, и лишь после этого буфер передается в сеть.
- Операция чтения возвращает последние данные, которые были переданы из буфера передачи. Если ничего отправлено не было, в этом случае элемент, возможно, будет доступен для чтения, хотя будет сопровождаться кодом качества "BAD" (Ошибка).

Элемент для приема (S7-коммуникации: "BRCV", S5-совместимые коммуникации: "RECEIVE")

- OPC-элемент представляет буфер приема.
- Если происходит чтение OPC-элемента из устройства (синхронно/асинхронно), модуль готов к приему данных. Такое состояние сохраняется до тех пор, пока пакет данных не будет принят или не будет превышен интервал времени, установленный для данного соединения. Если в течение этого времени не будет принят пакет данных, OPC-элемент получает код качества "BAD" (Ошибка).
- В случае мониторинга OPC-элемента (активный принимаемый элемент в активной группе), в коммуникационном модуле формируется стационарный приемный буфер. После того, как пакет данных получен, вызовы onDataChange сообщают об этом приложению в том случае, если данные отличаются от данных, принятых ранее.
- В элемент, предназначенный для приема, запись осуществляться не может.

2.14.3 Применение буфер-ориентированных сервисов

Работа с элементами для передачи и приема

В элементы для передачи должна происходить только запись; в случае чтения или мониторинга передаваемых элементов в ответ возвращаются только ранее записанные данные.

Для элементов приема должен выполняться мониторинг; другими словами, приемный элемент должен присутствовать как активный элемент в активной группе. Независимо от способа реализации функции возврата уведомлений (callback), кэш OPC-сервера обновляется после поступления каждого нового пакета данных.

Чтение из кэша и непосредственно из устройства

Задания на чтение, как синхронные, так и асинхронные, должны осуществлять чтение кэша. Если выполняется мониторинг принимаемого элемента, в кэше содержится последний принятый блок данных. Применение кэша также позволяет добиться того, что несколько клиентов, читающих элемент в одно и то же время, получают одно и то же значение.

Чтение непосредственно из устройства не имеет практического значения, поскольку в течение контрольного времени, выбранного для соединения (как правило, несколько секунд), доступен только буфер для приема. Передающий узел должен выполнить передачу в течение этого времени, в противном случае буфер данных не будет принят принимающим узлом.

Скорость обновления

Следите, чтобы OPC-сервер считывал приемный буфер быстрее, чем его передает передающий узел. Скорость, с которой считываются приемные буферы, указывается параметром "Update Rate" (Скорость обновления)(специальный параметр OPC-группы).

В зависимости от используемого протокола, принятые блоки данных будут либо перезаписаны без уведомления об этом OPC-клиента (S7, SEND/RECEIVE с ISO и RFC 1006), либо произойдет отставание и принимающий узел получит лишь устаревшие данные (SEND/RECEIVE с собственным протоколом TCP/IP).

2.14.4 Что требуется помнить при использовании буфер-ориентированных сервисов с собственным протоколом TCP/IP

Понятие "Сокет"

Протокол SEND/RECEIVE с собственным протоколом TCP/IP базируется на сокет-сервисах (Socket service) системы Windows. Буферы при этом в явном виде не передаются, а в сокет поступает непрерывный поток данных. Протокол использует внутренний буфер как на стороне передатчика, так и на стороне приемника, а управление поступающими и исходящими данными происходит по принципу буфера FIFO ("первым вошел – первым вышел").

Собственный протокол TCP/IP и OPC-переменные

Это свойство протокола также влияет на приложение, в котором используются OPC-переменные:

- Даже если партнер не получает никаких данных, в ответ на задание на передачу возвращаются положительные подтверждения до тех пор, пока не будет переполнен буфер на передающей стороне.
- Внутренний буфер для приема данных имеется всегда, независимо от того, контролируется ли переменная или передается ли задание на чтение.
- Если данные не считываются достаточно быстро на приемной стороне, в буфере приема образуется "отставание" данных. В результате, принимающая сторона получит не последние прочитанные данные, а наоборот, устаревшие данные.

3 Основы OPC-интерфейса

Эта глава познакомит вас с основной терминологией, которая потребуется для изучения OPC.

Вы узнаете об основах создания OPC-интерфейса и познакомитесь с OPC-интерфейсами *Data Access* (Доступ к данным) и *Alarms & Events* (Аварийные сообщения и события).

Последний раздел главы посвящен реализации OPC-сервера как компонента SIMATIC NET.

3.1 Основная терминология

Прежде чем изучать основы OPC и приступать к работе с этим интерфейсом, вы, вероятно, сочтете полезным познакомиться с некоторыми основными терминами. Этот раздел можно пропустить и вернуться к нему, когда потребуется пояснение того или иного термина.

3.1.1 Что такое объекты COM?

Что такое COM?

В основе построения OPC лежит компонентно-объектная модель (COM=Component Object Model) фирмы Microsoft.

COM – это стандарт, который позволяет создавать объекты в Windows в виде отдельных модулей и обращаться к этим объектам со стороны.

Каждый объект можно рассматривать как расширение операционной системы. Объекты не зависят от языков программирования и, в принципе, доступны для любых приложений.

Объекты COM – это компоненты, которые выполняются в среде Windows и предоставляют определенный набор функций другим компонентам через свои интерфейсы. COM-объект может использоваться одновременно несколькими приложениями.

Данные и объектный код объекта не являются прозрачными для пользователя COM-объекта.

Примечание

Понятие "объект" в рамках компонентно-объектной модели не соответствует определению "объекта", которое применяется в языках объектно-ориентированного программирования. СОМ-объект следует рассматривать как завершенный автономный модуль. СОМ-объект не обладает такими типичными свойствами языков объектно-ориентированного программирования, как, например, наследование.

Структура СОМ-объектов

Ниже показано схематичное представление СОМ-объекта с четырьмя интерфейсами, демонстрирующее его структуру. Обращение к объекту происходит только через интерфейсы. Для управления доступом используются различные методы. Ни к самому объекту в целом, ни к его данным и объектному коду напрямую обратиться нельзя.

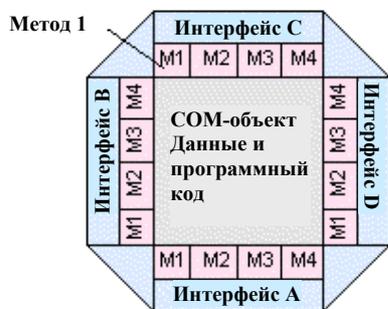


Рисунок 3–1 Структура СОМ-объекта

3.1.2 Как выглядят COM-объекты?

В документации, как правило, приводится графическое представление COM-объекта. Интерфейсы самого объекта располагаются с одной стороны, а неизвестный интерфейс (**IUnknown**), который имеется у всех объектов, находится сверху объекта.

Методы, предусмотренные для интерфейсов, скрыты от пользователя.

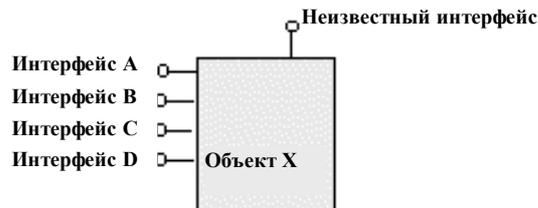


Рисунок 3–2 Представление COM-объекта

3.1.3 Что предоставляют COM-интерфейсы?

COM-интерфейс – это определенный набор методов, предназначенных для вызова функций COM-объекта. Он содержит таблицу указателей на методы. Функции COM-объекта заключены (инкапсулированы) внутри COM-интерфейса, благодаря чему обращение к объекту может произойти только установленным образом. COM-интерфейсам назначают уникальные идентификаторы, поэтому приложение, которому требуется обратиться к COM-объекту, может проверить, поддерживает ли объект интерфейс, прежде чем обратиться к нему.

Структура интерфейса

Ниже схематично представлена основная структура интерфейса.

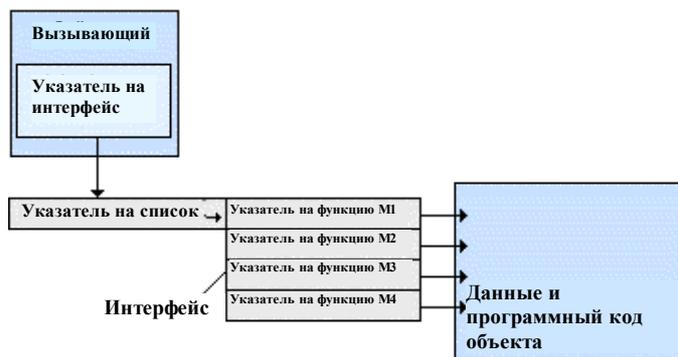


Рисунок 3–3 Структура интерфейса

3.1.4 Какие типы интерфейсов существуют?

В COM-модели различают следующие типы интерфейсов:

- Интерфейс автоматизации (Automation Interface)
- Пользовательский интерфейс (Custom Interface)

Интерфейсы отличаются способом вызова внутренних методов. Для каждого интерфейса существуют отдельные спецификации. Тем не менее, интерфейсы подходят для широкого круга задач, например, для доступа к переменным и для приема сообщений.

Для приложений, являющихся OPC-клиентами, созданных на базе языков описания сценариев, таких как Visual Basic или VBA, должен использоваться интерфейс автоматизации.

Для приложений, являющихся OPC-клиентами, написанных на C или C++, также можно использовать интерфейс автоматизации. Однако для достижения максимальной производительности рекомендуется применять пользовательский интерфейс.



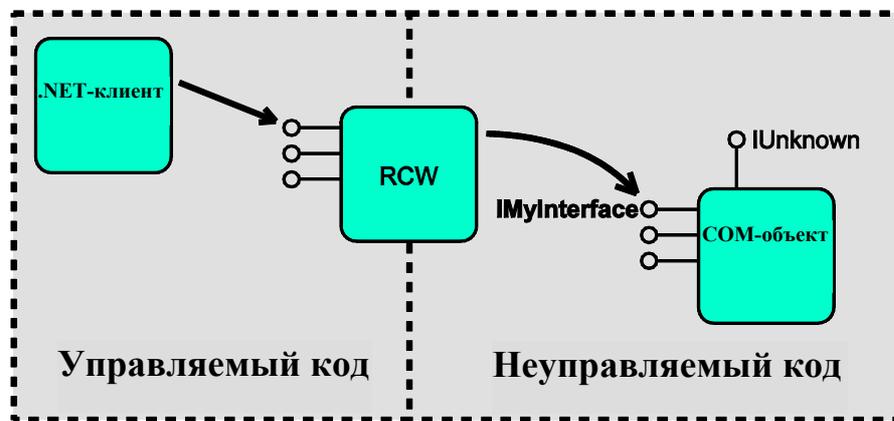
Рисунок 3–4 Пример использования интерфейсов приложениями

3.1.5 Как .NET-клиент может обратиться к COM-интерфейсу OPC?

Последовательность действий в случае применения пользовательского OPC-интерфейса

Находясь в области управляемого кода, .NET- клиент может обращаться к общему COM-объекту пользовательского OPC-интерфейса. Вследствие различия свойств моделей программирования COM и .NET (в модели .NET, например, не предусмотрен доступ с использованием указателей) прямой вызов невозможен.

Переход из области управляемого кода в область неуправляемого кода обеспечивается за счет применения RCW (RCW = Вызываемая оболочка режима выполнения). Оболочки RCW управляют потоком вызовов между управляемыми .NET-объектами и неуправляемыми COM-объектами.



Последовательность действий в случае применения OPC-интерфейса автоматизации

С помощью приложения импорта .NET Framework создается так называемая интероперабельная сборка (interop assembly) - .NET-компонент. .NET-клиенты могут использовать ее для создания экземпляров COM-объектов и вызова методов COM-объектов так, как если бы использовались экземпляры .NET. Таким образом, неуправляемый код (OPC-интерфейс автоматизации) преобразуется в компонент .NET.

3.1.6 Что означает управление объектами через интерфейс автоматизации?

Управление объектами служит для управления COM-объектами с помощью языков описания сценариев.

Пользовательские интерфейсы не подходят для целого ряда функций, которыми обладают средства проектирования, базирующиеся на языках описания сценариев. В результате добавления к COM-объектам распределительного интерфейса (интерфейса автоматизации) методы объекта становятся доступными даже для простых языков описания сценариев. Благодаря интерфейсу автоматизации команды, распознаваемые объектом, становятся "видимыми" внешним приложениям.

Программируемый доступ к COM-серверам автоматизации описывается набором свойств и методов. В результате достигается простое внедрение COM-объекта в такие среды разработки, как, например, VBA (Visual Basic для приложений).

3.2 Введение в XML и SOAP

XML и OPC–XML

XML (Расширяемый язык разметки) – это новый стандарт для Internet, который в настоящее время получил широкое применение в стандартном программном обеспечении во многих областях. Так же, как и HTML, XML позволяет дополнять данные метаданными. Отличием является возможность определения собственных структур данных и атрибутов.

На базе XML была создана новая спецификация для OPC, получившая название OPC–XML. Она описывает интерфейс обмена данными процесса посредством так называемых "записей" (record) данных XML.

Доступ к OPC через Internet

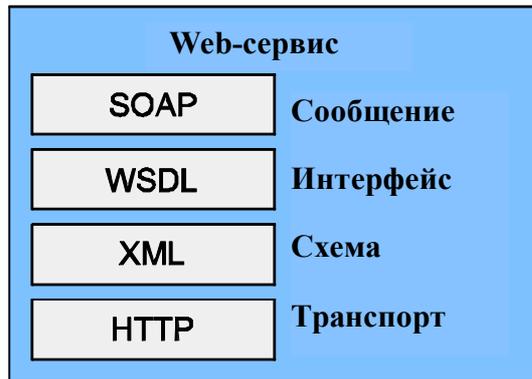
Связь через DCOM-интерфейсы OPC, как правило, не выходит за пределы локальных сетей. Более того, COM-интерфейсы в общем случае определяются для систем на базе Windows. С целью повышения безопасности применяются системы защиты (firewall), которые ограничивают доступ из/в Internet. С появлением OPC–XML также появился стандарт, обеспечивающий связь с помощью междуплатформенного протокола SOAP (Простой протокол доступа к объектам). Для обращения к данным с помощью OPC–XML предусмотрен целый ряд функций, базирующихся на OPC Data Access (Обращение к данным через OPC).

Описание интерфейса с помощью XML

Язык XML описывает интерфейсы доступа к данным и методы. Точное описание методов содержится в спецификации WSDL (WSDL = Язык описания Web-сервисов), предоставляемой организацией OPC Foundation вместе со спецификацией OPC–XML DA. Эти методы описываются с помощью SOAP (XML-протокол) и передаются с помощью протокола HTTP. Кратко это можно выразить следующим образом:

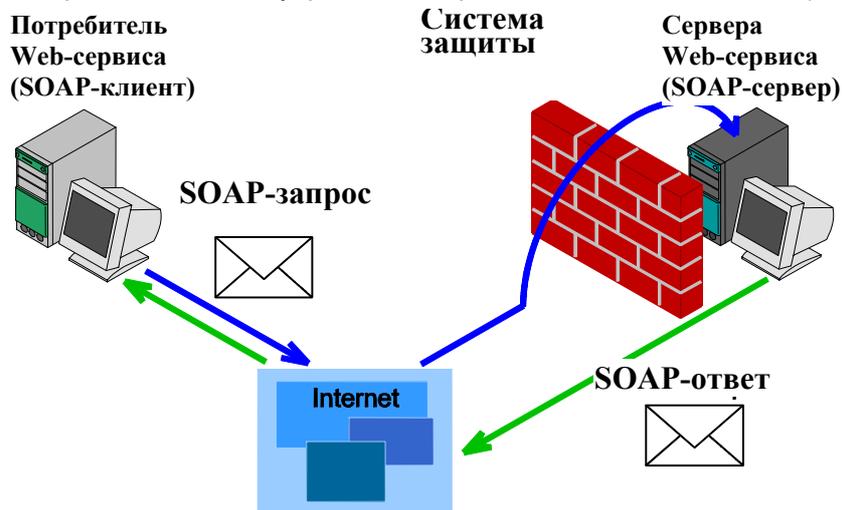
SOAP = HTTP + XML

Структура взаимодействия представлена на следующем рисунке:



Передача данных с помощью протокола HTTP

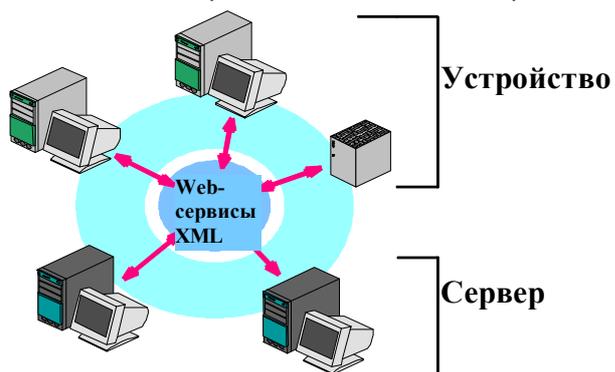
Возможность доступа к методам непосредственно из Internet связана со значительным риском. По этой причине SOAP использует HTTP-канал Internet (HTTP = Протокол передачи гипертекста) только для такой передачи данных, которой можно легко управлять посредством системы защиты (firewall):



3.2.1 Web-сервисы

Вызов функций через Internet

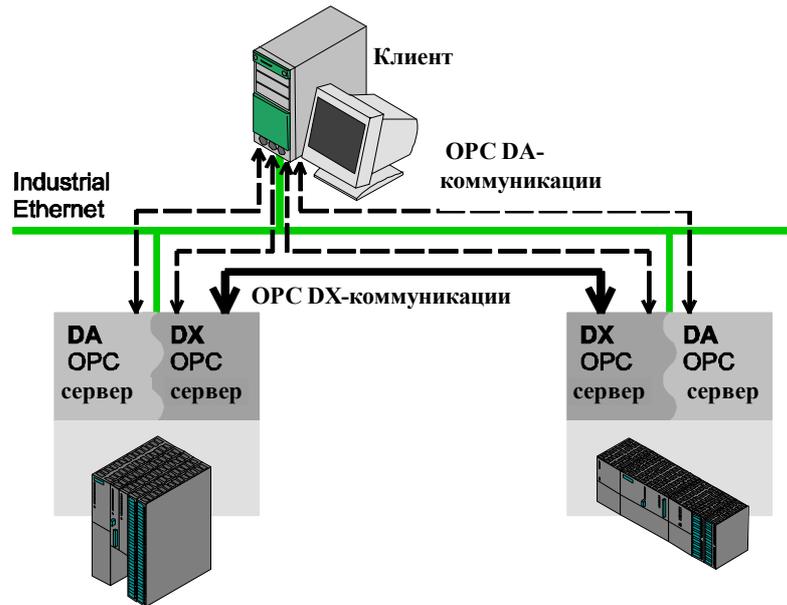
Вызовы функций могут передаваться серверу через Internet с помощью Web-сервиса. Описание методов и параметров, предоставляемых Web-сервисом, хранится в WSDL-файлах в формате XML. Клиент может запрашивать их у Web-сервера. Для эффективного использования Web-сервисов требуется только знание Internet-адреса или URL Web-сервиса. Данные передаются в виде HTTP-кадров с использованием протокола SOAP.



3.3 Введение в OPC DX

Обмен данными между OPC-серверами

OPC DX позволяет организовать связь между серверами OPC-DA и OPC-DX в сетях Ethernet. В отличие от OPC-DA, OPC-DX обеспечивает только классические коммуникации "клиент-сервер". Это наглядно показано на следующем рисунке.



Благодаря универсальности OPC-интерфейса возможен обмен данными между DX-серверами или между DX- и DA-серверами различных производителей.

DX-сервер предоставляет доступ к данным других OPC-серверов, запрашивая соответствующие элементы у этих серверов. В этом смысле DX-сервер выступает в роли DA-клиента.

DX-соединения

Обмен данными между OPC-серверами описывается в DX-соединениях. Помимо прочих параметров в DX-соединении определяется элемент источника данных (на DA- или DX-сервере) и целевой элемент (на DX-сервере), между которыми происходит обмен данными. DX-соединение отображается в качестве OPC-DA-элемента, предназначенного только для чтения.

База данных DX

Все DX-соединения объединяются в базу данных DX. Содержимое базы данных DX "привязывается" к части пространства имен DA-сервера. OPC-DA-клиенты могут обращаться к базе данных DX через OPC DA-интерфейс.

В базе данных DX также содержится список серверов-источников данных. Серверами-источниками являются OPC-DA- и DX-серверы, в которых содержатся элементы источника данных.

Конфигурирование сервера-источника данных и DX-соединения

DX-сервер передает данные в соответствии с конфигурацией, созданной DX-клиентом. OPC DX предоставляет методы, с помощью которых DX-клиент может добавлять, изменять и удалять серверы-источники или DX-соединения. Эти методы могут быть реализованы в рамках Web-сервиса или в форме DCOM-интерфейса.

Управление и контроль за передачей данных в режиме выполнения (runtime)

Для каждого DX-соединения предусмотрен ряд атрибутов (элементы в пространстве адресов DA-сервера), с помощью которых описывается процедура передачи данных. DA-клиент может влиять на передачу данных в режиме выполнения (runtime), записывая в эти элементы определенные значения.

Клиент, например, может изменять следующие параметры:

- Установить/разорвать соединение с элементом-источником
- Установить/разорвать соединение с целевым элементом

Клиент, например, может читать или контролировать следующие элементы:

- Подставлять значение, элемент-источник недоступен
- Параметры элементов в режиме выполнения (runtime) (такие параметры, как значение, код качества, метка времени)

3.4 Введение в OPC

Что такое OPC?

OPC = OLE for Process Control (OLE для управления процессами).

На момент появления OPC для всей компонентной архитектуры Microsoft применялся термин OLE. Первоначально под OLE понимались механизмы внедрения объектов в комбинированные документы.

С тех пор произошло много изменений, и теперь OPC следовало бы расшифровывать как "COM для управления процессами", поскольку OPC базируется на компонентно-объектной модели (COM). Модель COM является ключевым элементом операционных систем Windows, управляя взаимодействием многочисленных программных компонентов.

Благодаря применению COM OPC-сервер становится "известен" части операционной системы Windows и поэтому избавляется от зависимости от имен файлов, места их расположения и версий.

Как продолжение развития технологии COM, DCOM поддерживает работу распределенных приложений и обеспечивает взаимодействие между программными компонентами, расположенными на различных компьютерах в сети.

До появления OPC требовалось приложить значительные усилия для управления аппаратными средствами различных производителей с помощью прикладного программного обеспечения. Существовало огромное количество различных систем и протоколов, причем для каждого оборудования и для каждого протокола пользователю приходилось заказывать специальное программное обеспечение, открывающее доступ к специальным интерфейсам и содержащее необходимые драйверы. Пользовательские программы, таким образом, зависели от конкретного производителя, протокола или системы. OPC на базе COM или DCOM обладает унифицированным программным интерфейсом, не зависящим от конкретного производителя, что стало переломным этапом в организации обмена данными в системах автоматизации.

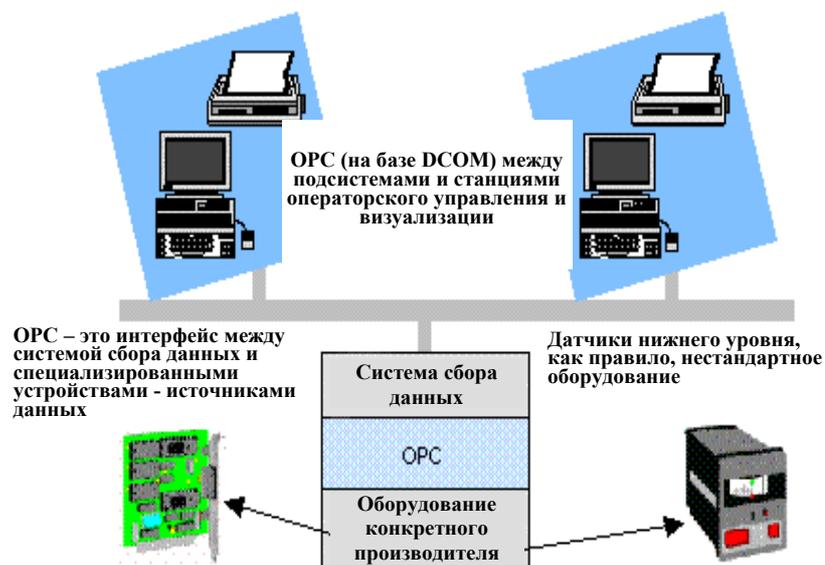


Рисунок 3–5 Области применения OPC

3.4.1 Что предоставляет OPC-интерфейс?

В качестве промышленного стандарта OPC описывает обмен данными для различных приложений в условиях промышленного производства. Работая на своем ПК, пользователь может наблюдать, вызывать и обрабатывать данные и события, происходящие в системах автоматизации.

OPC-интерфейс является частью программного обеспечения, работающего на ПК, и является платформой для систем операторского управления и визуализации или других приложений. Таким образом, он располагается ниже уровня прикладной программы.

Область применения OPC-интерфейса

Организация OPC Foundation занимается разработкой спецификаций для OPC-интерфейса, начиная с 1996 года. В настоящее время для проектирования систем автоматизации существуют следующие спецификации:

- для обмена данными (переменными процесса): Data Access (Доступ к данным)
- для обслуживания аварийных сообщений и событий: Alarms & Events (Аварийные сообщения и события)
- для доступа к архивным данным: Historical Data Access (Обращение к статистическим данным)
- для работы с рецептами: Batch (Серийное производство)

В качестве интерфейса взаимодействия с промышленными системами связи OPC-сервер предоставляет функции Data Access (Доступ к данным) и Alarms & Events (Аварийные сообщения и события). В настоящей документации описываются только эти спецификации.

Реализация OPC-интерфейса

OPC-интерфейс базируется на модели "клиент/сервер". Один компонент предоставляет свои сервисы другому компоненту через интерфейсы. Другой компонент использует эти сервисы. В OPC предусмотрено, что приложение может определять, какие OPC-серверы существуют в системе. Затем оно может обратиться к одному или нескольким из этих серверов и запросить предоставляемые ими сервисы. Поскольку к одному и тому же OPC-серверу могут обращаться одновременно несколько различных OPC-клиентов, один и тот же источник данных может использоваться любым OPC-совместимым приложением.

Производители оборудования, являющегося источником данных процесса (системы связи, измерительные приборы и т.п.) разрабатывают для своего оборудования OPC-сервер, который затем обеспечивает возможность обращения к соответствующим данным этого оборудования.

3.4.2 Что такое OPC-сервер?

Компоненты OPC, являющиеся источниками данных, называются OPC-серверами. Они обеспечивают взаимодействие (интерфейс) с существующими системами связи. Помимо сервисов они предоставляют OPC-клиенту данные любого источника данных. Источником данных может быть как оборудование, так и программа. Источниками данных могут быть, к примеру, порты связи, карты для подключения к полевым шинам, измерительные приборы или регуляторы (контроллеры).

Имена серверов

Каждому OPC-серверу производитель назначает уникальное имя, позволяющее однозначно идентифицировать этот сервер. Согласно стандарту COM-модели, эти имена обозначают как ProgID. Указывая соответствующий ProgID, пользователь может обращаться к конкретным OPC-серверам.

Типы серверов

Существует три типа OPC-серверов. Тип определяется тем, как сервер включается в систему связи:

- Внутрипроцессный (In-Process) сервер
- Локальный сервер
- Удаленный сервер

Производитель OPC-сервера указывает, является он внутрипроцессным или локальным. Работа в режиме удаленного сервера конфигурируется пользователем.

С точки зрения приложения не имеет никакого значения, какой сервер используется. Синтаксис для вызова методов всегда один и тот же.

На следующей схеме показано, к каким компонентам OPC-сервера обращаются клиенты различных типов:

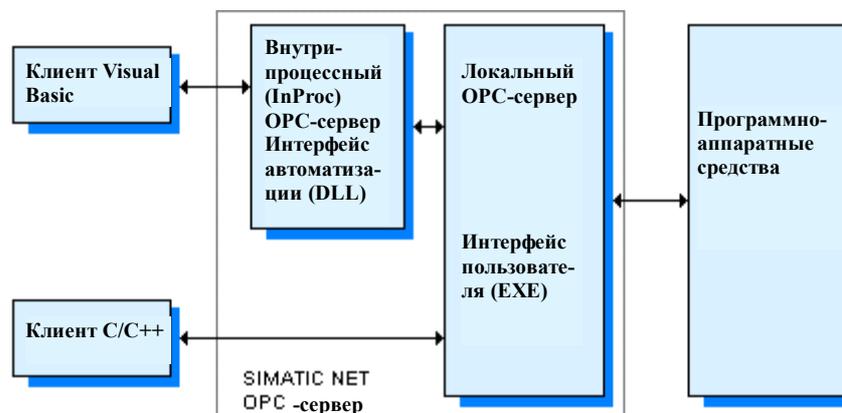


Рисунок 3–6 OPC-серверы

3.4.3 Что такое OPC-клиент?

Компоненты OPC, которые используют OPC-сервер в качестве источника данных, называют OPC-клиентами. На рынке OPC-клиенты доступны в качестве стандартного программного обеспечения. Также доступны программные модули, которые можно компоновать для создания собственных клиентских приложений. Чтобы выполнить индивидуальные требования своей системы и добиться наилучшей производительности, можно создавать собственные OPC-клиенты на различных языках программирования (например, Visual Basic, C и C++).

Некоторые свойства OPC-серверов (например, имена переменных) не описаны в стандарте на OPC и зависят, к примеру, от свойств системы автоматизации или производственного процесса, и описываются, например, производителем. Чтобы избежать возникновения проблем при работе OPC-клиентов с различными OPC-серверами, необходимо продумывать конфигурацию переменных при программировании. Это делает прикладную задачу более гибкой и позволяет ее использовать повторно.

3.4.4 Каким образом сервер и клиент взаимодействуют друг с другом?

Взаимодействие сервера и клиента происходит на основе технологии COM или DCOM. Клиент не обращается к серверу напрямую, а использует библиотеку COM. Указывая идентификатор ProgID, OPC-клиент может обращаться к любому OPC-серверу, который ему нужен.

Путь доступа для клиента (с применением COM или DCOM) не прозрачен. На следующем рисунке показаны возможные пути доступа: когда оба находятся на одной машине и для обмена между разными машинами.

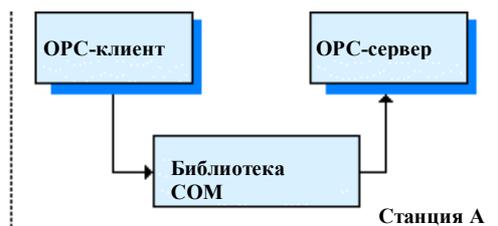


Рисунок 3–7 COM на локальном компьютере

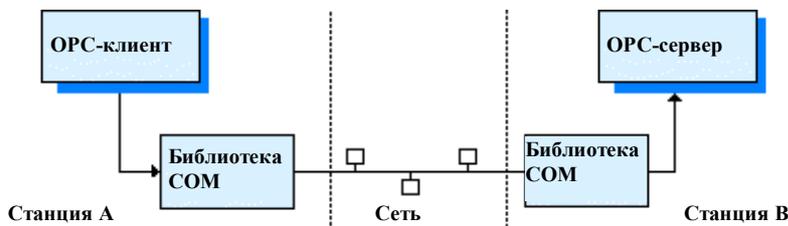


Рисунок 3–8 COM на другом компьютере

Свойства и методы

Функционирование OPC-серверов определяется их интерфейсами. Следовательно, OPC-клиенту известны предположительные функции сервера, и он может использовать имеющиеся сервисы нужным образом. С точки зрения объектно-ориентированного проектирования сервисы OPC-сервера представляются свойствами и методами. Все OPC-серверы обладают базовым набором одинаковых свойств и методов. В спецификациях OPC также предусматриваются некоторые необязательные интерфейсы. Если сервер не обеспечивает такую необязательную функцию, клиент узнает об этом и продолжает свою работу соответствующим образом. В результате удается избежать проблем при совместной работе компонентов от различных производителей.

Клиент может создавать, использовать и удалять объекты на сервере посредством OPC-интерфейсов. OPC-клиент ссылается на функции сервера и использует методы сервера для чтения и записи данных. Каждая функция сервера соответствует определенному вызову на стороне клиента.

3.5 OPC Data Access

Что такое OPC Data Access?

Data Access (Доступ к данным) – это спецификация OPC, предназначенная для доступа к данным процесса с использованием переменных. OPC-сервер для Data Access управляет переменными процесса и различными способами обращения к переменным. Сервер может:

- читать значение одной или нескольких переменных процесса
- изменять значения одной или нескольких переменных процесса, записывая новое значение
- контролировать значение одной или нескольких переменных процесса.

Сервер проверяет, произошло ли изменение значения переменной процесса. Если появляется новое значение, сервер сообщает об этом клиенту.

Скорость передачи сообщений клиенту ограничивается, однако, параметром "Update Rate" (Скорость обновления), установленным для клиента. Если значение изменяется быстрее, чем установленная "скорость обновления", клиент не узнает об изменении переменной, произошедшей в промежутке между соседними сообщениями.

Переменные процесса – это поля (тэги) для значений, которые должны опрашиваться в режиме выполнения (runtime).

Значения процесса

Источник значений процесса не описывается в спецификации OPC и зависит от конкретного производителя. Это означает, что через Data Access может быть передана любая информация.

3.5.1 Что предоставляет модель классов спецификации OPC Data Access?

Когда клиент обращается к данным, иерархическая модель классов спецификации Data Access позволяет привести метку времени и полученный результат в соответствие текущим требованиям приложения. В Data Access можно выделить три следующих класса:

- OPC-сервер
- OPC-группа
- OPC-элемент

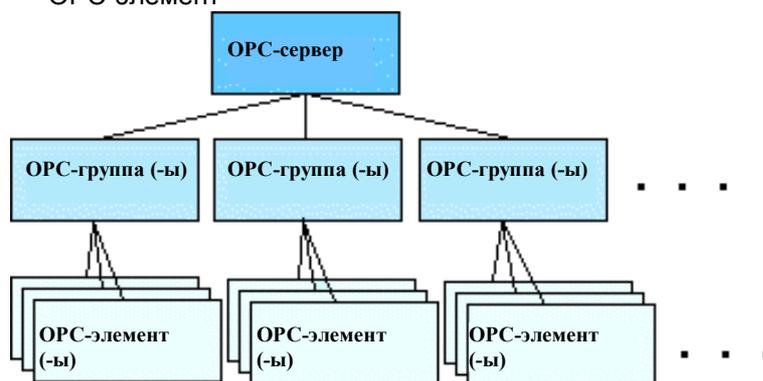


Рисунок 3–9 Модель классов интерфейса Data Access

Клиентское приложение использует вызовы СОМ операционной системы только для создания объекта класса "OPC-сервер". Прочие объекты создаются посредством соответствующих методов OPC класса "OPC-сервер" или нижестоящих классов.

Модель классов используется как для интерфейса автоматизации, так и для интерфейса пользователя (с учетом некоторых ограничений).

3.5.2 Класс "OPC-сервер"

Самым верхним является класс "OPC-сервер". Каждый OPC-сервер принадлежит этому классу. Этот класс предоставляет точку доступа для всех других сервисов сервера OPC Data Access.

С помощью специальных атрибутов и методов класса пользователь может получать информацию о статусе, версии, а также (необязательно), о пространстве имен имеющихся переменных процесса. Объект класса "OPC-сервер" управляет экземплярами нижестоящих классов, другими словами, класса "OPC-группа".

3.5.3 Класс "OPC-группа"

Класс "OPC-группа" располагается непосредственно под классом "OPC-сервер" и определяет структуру переменных процесса, используемых OPC-сервером. OPC-клиент может использовать несколько объектов этого класса одновременно. С помощью объектов "OPC-группа" клиент может формировать смысловые группы переменных процесса и выполнять над ними определенные операции. Например, может оказаться удобным объединить в одну группу все переменные процесса, расположенные на одной экранной форме системы операторского управления и визуализации. Класс "OPC-группа" определяет методы, с помощью которых можно читать и записывать переменные процесса.

Некоторые методы позволяют объединять несколько переменных в единое задание, чтобы передавать их одновременно. В частности, в случае применения OPC-сервера по сети эти операции над группами ускоряют работу.

3.5.4 Класс "OPC-элемент"

Объекты этого класса представляют фактические переменные процесса и обеспечивают чтение конкретных данных. Каждая переменная – это элемент (item) в пространстве имен OPC-сервера, который идентифицируется с помощью идентификатора элемента (Item ID). Идентификатор элемента определяется производителем сервера и должен быть уникальным в пространстве имен сервера. Каждый элемент обладает следующими свойствами:

- Значение
Последнее опрошенное значение переменной.
- Код качества
Достоверность значения. Если качество "хорошее", полученное значение достоверно.
- Метка времени
Время, когда было впервые опрошено текущее значение переменной. Всякий раз, когда клиенту сообщается об изменении переменной, происходит обновление метки времени. Если значение переменной не изменяется, метка времени остается той же самой.

Роль переменных

Переменные должны указываться в вызовах OPC-интерфейса с целью получения значений процесса. Указывая переменные, клиент может запрашивать требуемые значения от сервера. Клиент должен зарегистрировать каждую требуемую переменную на сервере, чтобы указать, какие значения будут читаться, и когда. Чтение и запись переменных можно осуществлять синхронно и асинхронно.

Клиент может возложить функции мониторинга переменных на сервер. Когда изменяется значение переменной, сервер передает сообщение клиенту.

Переменные, предоставляемые OPC-сервером, можно разбить на следующие группы:

- Переменные процесса
Соответствуют измеряемым и управляемым переменным процесса, источниками которых являются устройства ввода и вывода, или
- Переменные для управления
С помощью этих переменных запускаются определенные дополнительные сервисы, например, передача паролей.
- Информационные переменные
Эти переменные предусматриваются системой связи и несут в себе информацию о состоянии соединений, устройств и т.п.

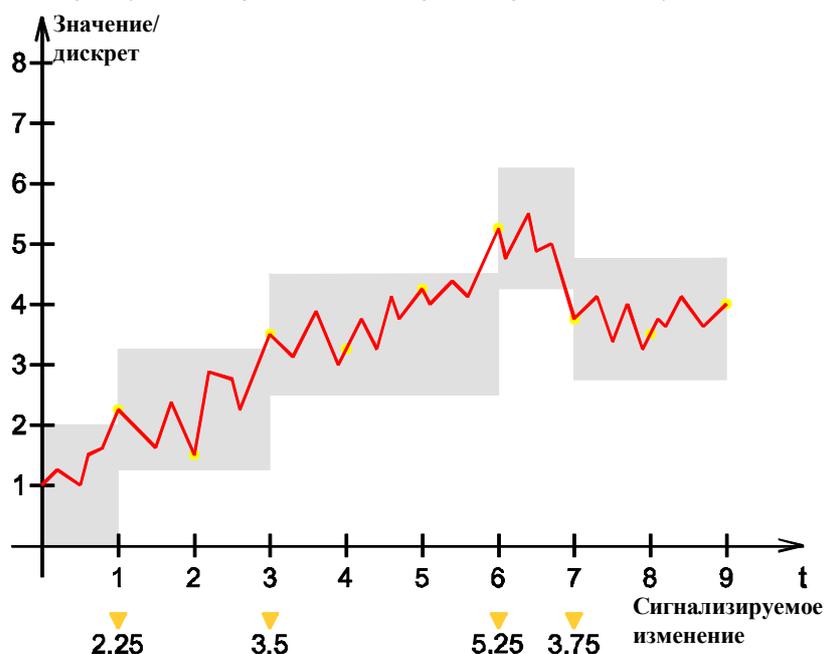
Ниже приведено несколько примеров переменных сервера OPC Data Access:

- Управляющие переменные программируемого контроллера
- Изменяемые данные системы сбора данных
- Переменные статуса системы связи

3.5.5 Использование параметра “Percent Deadband”

Данный параметр указывается в процентах и определяет диапазон, в пределах которого изменение всех элементов группы не сигнализируется. Абсолютное значение границы зоны нечувствительности определяется как процентное значение от всего диапазона значений переменной процесса.

В следующем примере: Percent Deadband (Зона нечувствительности) = 10% = 1 дискрет (нижняя граница = 0, верхняя граница = 10).



Вслед за каждым изменением значения вокруг последнего достигнутого значения формируется зона нечувствительности. Сообщение об изменении формируется и передается клиенту только в том случае, если значение выходит из зоны нечувствительности с момента последнего чтения, приведшего к установке новой зоны нечувствительности.

Примечание

В соответствии со спецификацией OPC выход за верхнюю или нижнюю границы не приводит к формированию каких-либо специальных аварийных сообщений!

3.5.6 Какие интерфейсы определены для OPC Data Access?

Для спецификации Data Access (Доступ к данным) определены интерфейс автоматизации и пользовательский интерфейс:

- Стандарт Data Access Automation Interface ("Интерфейс автоматизации для доступа к данным"), 4 февраля, 1999 г., версия 2.02
- Стандарт Data Access Custom Interface ("Интерфейс пользователя для доступа к данным"), 14 октября, 1998 г., версия 2.0

3.6 Интерфейс OPC DX

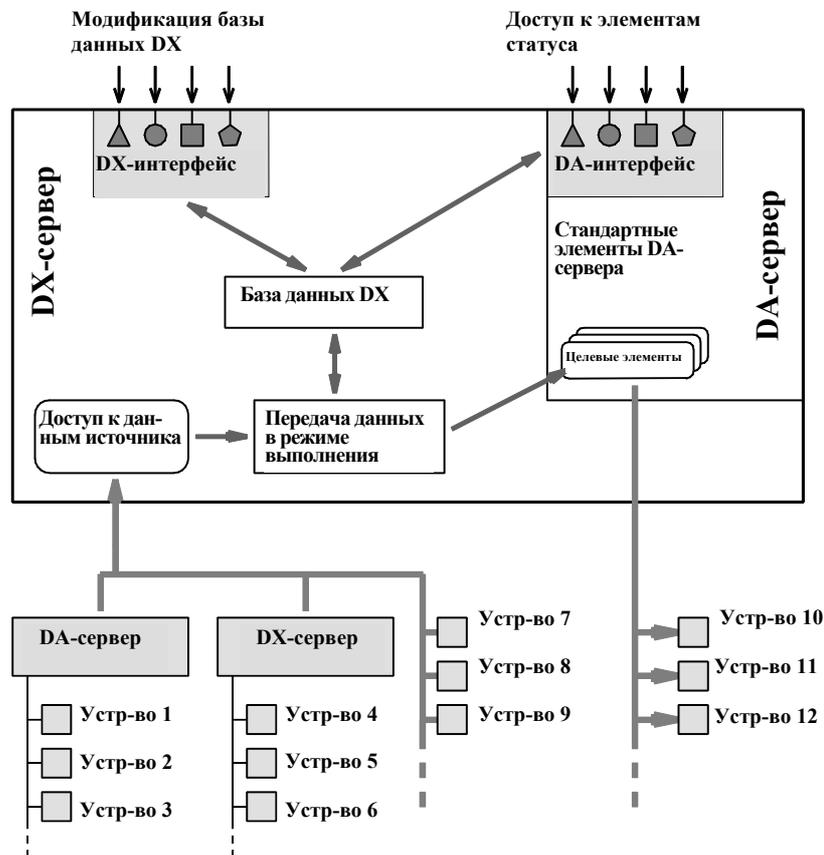
3.6.1 DX-сервер

DX-сервер как расширение DA-сервера

DX-сервер базируется на DA-сервере, который соответствует одной из спецификаций, предусмотренных для OPC-интерфейсов. DX-сервер также снабжается функциями, необходимыми для организации связи между серверами. DX-сервер – это DA-сервер, который дополнен следующими функциями:

- DX-интерфейс
- конфигурируемый OPC-DA-клиент

Структура DX-сервера показана на следующем рисунке:



Специальные функции DX-сервера

DX-сервер отличается от DA-сервера, обладая следующими дополнительными функциями:

- DX-сервер ведет базу данных DX. Эта база данных содержит все данные о самом сервере и о сервере-источнике, а также о DX-соединениях.
- Интерфейс конфигурирования предоставляет методы для добавления, изменения и удаления серверов-источников или DX-соединений.
- DX-сервер обладает механизмом, который соответствует DA-клиенту, с помощью которого он может регистрировать себя у источника данных с целью циклического обновления данных источника.
- В режиме выполнения (runtime) DX-сервер передает адресату копию данных источника. Клиенты могут разрывать соединение между DX-сервером и источником или целевыми элементами и указывать правила обновления данных. Клиент, например, может отменить обновление или задать фиксированные значения в целевом элементе.
- DA-элементы и методы DX-интерфейса, предназначенные для управления реакцией в режиме выполнения, описаны в предыдущем разделе.

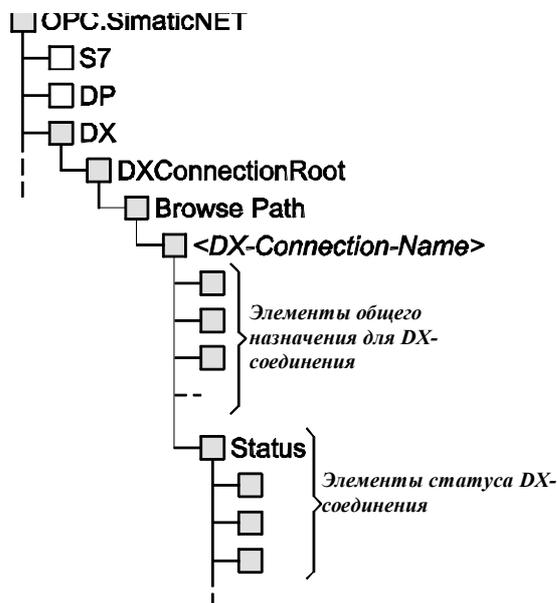
3.6.2 База данных DX

Создание структуры данных с помощью пространства имен

DX-сервер использует для базы данных DX часть пространства имен DA-сервера. К содержащимся в этой базе данных OPC DA-элементам можно обращаться с помощью методов DX-интерфейса (?).

DX-элемент – это корневой узел такой структуры данных. В отходящих от него ветвях располагаются элементы, являющиеся промежуточными точками обзора, а также элементы, соответствующие DX-соединениям. Элементы, располагающиеся в ветви DX-соединения, содержат всю информацию, необходимую DX-серверу для управления обменом данных через это DX-соединение. В эту информацию входят, например, сведения, идентифицирующие источник и адресуемый элемент, а также сервер-источник.

На следующем рисунке показано расположение описанных элементов в пространстве имен DA OPC:



Элементы общего назначения для DX-соединения

Каждое DX-соединение содержит несколько атрибутов, определяющих свойства этого соединения.

Элементы статуса

Элемент Status (Статус) содержит атрибуты двух типов. К одной группе относятся элементы для контроля за передачей данных. Эти элементы предназначены только для чтения и обновляются DX-сервером в режиме выполнения. Другая группа элементов служит для управления работой сервера в режиме выполнения. Если, например, в элементе *SourceItemConnected* лежит значение *FALSE*, в этом случае DX-сервер больше не запрашивает данные от соответствующего источника.

3.7 OPC Alarms & Events

Что подразумевается под Alarms & Events?

Alarms & Events (Аварийные сообщения и события) – это спецификация, предназначенная для передачи аварийных сообщений и событий процесса. Она имеет гибкую структуру, что позволяет использовать ее для широкого круга источников событий. Она охватывает широкий круг событий, от простых до составных, и даже позволяет передавать события, подлежащие квитированию.

Спецификация OPC с помощью диаграммы состояний определяет возможные переключения состояний для событий-условий.

Доступ к событиям базируется на модели классов.

3.7.1 Что такое события и сообщения о событиях?

Выбор сообщений с помощью фильтра

События – это определенные состояния процесса, о наступлении которых следует уведомлять сторону - получателя сообщений. OPC-клиент, используя соответствующие фильтры, решает, о каких событиях и каким программам следует сообщать.

Все события, отвечающие критерию, установленному с помощью фильтра, перенаправляются от источника события пользователю. В этом состоит отличие Alarms & Events от Data Access. Во время мониторинга переменных сигнализируются только изменения значений, происходящие за установленный промежуток времени.

Содержание сообщения

Сообщение содержит параметры, установленные спецификацией OPC, а также, возможно, сопутствующие значения, указанные изготовителем.

Сообщения могут быть простыми или более сложными, связанными с состояниями. В случае сложных, связанных с состояниями сообщений, источник события может потребовать квитиование (подтверждение получения) от пользователя.

Типы событий

Спецификацией OPC определено три типа событий:

- События, связанные с выполнением условий

Они сигнализируют об изменениях состояний, описанных моделью переключения состояний OPC, и формируются в случае соблюдения определенных условий.

- События слежения
Сигнализируют об изменениях в процессе, например, если пользователь изменяет уставку регулятора.
- Простые события
Сигнализируют обо всех остальных событиях, не связанных с переключением состояний, например, о неисправности компонента системы.

В спецификации OPC определен синтаксис интерфейса, предназначенного для приема сообщений. Типы событий, поддерживаемые сервером, определяются изготовителем OPC-сервера.

3.7.2 Что предоставляет модель классов OPC Alarms & Events?

Модель классов Alarms & Events позволяет адаптировать OPC-клиента к требованию системы автоматизации. В Alarms & Events различают три класса:

- OPC Event Server (OPC-Сервер событий)
- OPC Event Subscription (OPC-Подписка на события)
- OPC Event Area Browser (OPVC-Браузер области событий)

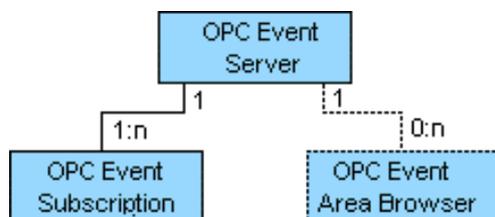


Рисунок 3–10 Модель классов интерфейса Alarms & Events

Примечание

Класс OPC Event Area Browser не является обязательным и сервером OPC Alarms & Event семейства SIMATIC NET не поддерживается.

3.7.3 Класс OPC Event Server

Функции OPCEventServer

Самым высшим является класс OPC Event Server (OPC-Сервер событий). Каждый сервер OPC Alarms & Events принадлежит этому классу. Данный класс представляет точку доступа для всех других сервисов сервера OPC Alarms & Events.

Объекты класса OPC Event Server главным образом управляют регистрацией клиентов для приема событий. Они также предоставляют информацию о поддерживаемых категориях событий, а также об атрибутах, которые могут передаваться вместе с сообщением.

Клиентское приложение обращается к объектам класса OPC Event Server с помощью механизмов COM и может принимать события, предоставляемые сервером событий.

Квитирование событий

Используя метод AckCondition класса OPCEventServer, клиент квитирует события, связанные с условиями, если необходимость квитирования указана в параметре события AckRequired. Как только поступает квитирование, параметр NewState события, связанного с условием, изменяется, а значит, наступает новое событие:

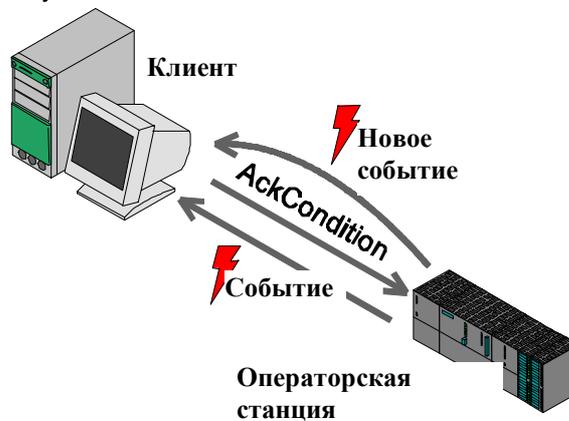


Рисунок 3–11 Последовательность наступления условных событий и их квитирования

3.7.4 Класс OPC Event Subscription

Функции класса OPCEventSubscription

Используя объекты класса OPC Event Server (OPC-Сервер событий), клиент создает один или несколько объектов класса OPCEventSubscription (OPC-

Подписка на события). Объект этого класса соответствует подписке на получение сообщений о ряде событий. Объекты этого класса управляют фильтрами и атрибутами, необходимыми для конкретного клиента. С помощью метода `SelectReturnedAttributes` пользователь указывает, какие именно атрибуты события передаются с определенным сообщением о событии. Используя объекты класса OPC Event Subscription, клиент может формировать нужные ему группы и выполнять над ними операции.

Фильтрация событий

Путем фильтрации клиент может указывать, о каких событиях ему следует сообщать. Фильтр – это всего лишь способ выбора событий на основании их свойств. Используются следующие условия выбора (критерии):

- EventType (Тип события)
- Category (Категория)
- Priority (Приоритет)
- Event source (Источник события)

События направляются клиенту только в том случае, если выполняются все критерии, установленные фильтром.

Буферирование событий

Если каждое сообщение передается клиенту отдельно, на это требуется больше ресурсов по сравнению с тем, если бы несколько событий передавались одновременно. С помощью параметра `BufferTime` клиент может указать, что событие может передаваться только по истечению определенного временного интервала. Событие, происходящее в пределах этого интервала, помещается в буфер до тех пор, пока не истечет интервал, после чего все содержащиеся в буфере события передаются вместе.

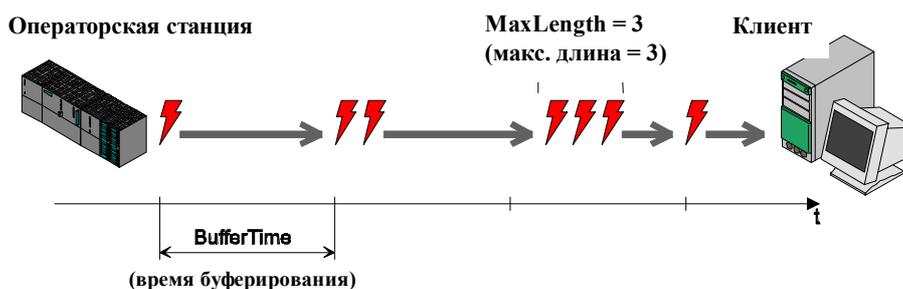


Рисунок 3–12 Значение параметров BufferTime и MaxLength

С помощью параметра `MaxSize` (Максимальный размер) можно задать максимальное количество сообщений, размещаемых в буфер. Как только указанное количество будет достигнуто, клиенту передаются все события, независимо от того, истек ли выбранный интервал `BufferTime`.

Параметры BufferTime и MaxSize используются для метода CreateEventSubscription (Создать подписку на события) класса OPCEventServer и для методов GetState (Получить состояние) и SetState (Установить состояние) класса OPCEventSubscription.

3.7.5 Класс OPC Event Area Browser

OPC Alarms & Events позволяет разбить обширное производство на отдельные области (зоны). Эти области можно использовать для фильтрации событий. Применение объектов класса OPC Event Area Browser (OPC-Браузер областей событий) позволяет обозревать отдельные области (зоны) производства.

Примечание

Объекты класса OPC Event Area Browser не являются обязательными и не поддерживаются сервером OPC Alarms & Event семейства SIMATIC NET.

3.7.6 Как выполняется прием сообщений?

Приложение регистрируется для приема сообщений, выполняя четыре действия:

1. Клиент регистрируется на сервере для получения сообщений.
2. Клиент создает один или несколько объектов класса OPCEventSubscription.
3. Клиент возвращает ответ (callback) через интерфейс IConnectionPointContainer.
4. Клиент предоставляет метод OnEvent (По событию), который вызывается сервером в случае возникновения события.

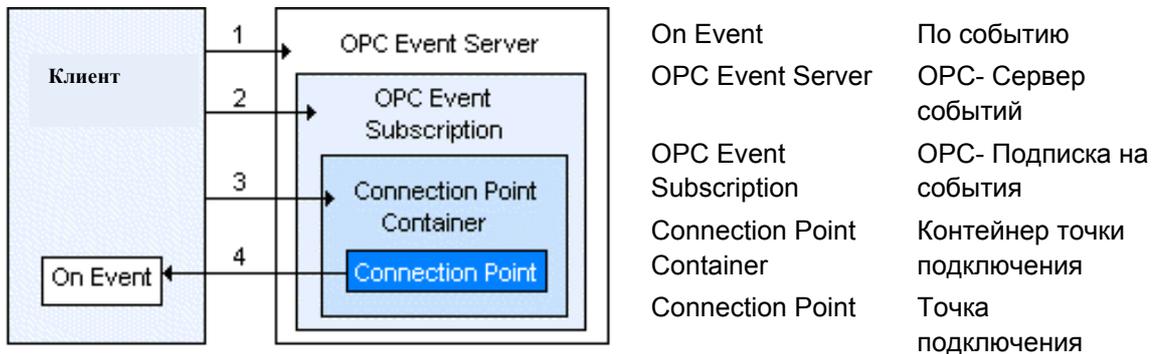


Рисунок 3–13 Подключение клиента к серверу для приема сообщений

3.7.7 Аварийные сообщения в SIMATIC S7

Аварийное сообщение (тревога) характеризуется следующими свойствами:

- Аварийное событие сигнализируется изменением состояния двоичного сигнала (перепадом фронта).
- Изменение состояния сигнала приводит к новому состоянию двоичного сигнала, которое длится в течение времени $t > 0$.
- Каждое изменение состояния сигнала может (но не обязательно) квитироваться получателем сигнала.
- Состояние квитирования может (но не обязательно) контролироваться формирователем аварийного сообщения.
- Сигнал может измениться вновь до поступления квитирования предшествующего изменения состояния сигнала.

В SIMATIC S7 аварийные сообщения могут генерироваться различными блоками:

Блок	Название	Кол-во сигналов	Квитирование	Значения	Степень важности
SFB36	NOTIFY	1	Нет	1-10	0-127
SFB33	ALARM	1	Да (с SFB33)	1-10	0-127
SFB34	ALARM_8	8	Да (с SFB34)	Нет	0-127
SFB35	ALARM_8P	8	Да (с SFB35)	1-10	0-127
SFC17	ALARM_SQ	1	Да (с SFC19)	1	Нет
SFC18	ALARM_S	1	Квитируется неявным образом	1	Нет

S7-программа пользователя указывает, требуется ли квитирование получателем аварийного сообщения. В S7-программе различают квитирование возникновения аварийного состояния (наступление аварии) и квитирование завершения аварийного состояния (уход аварии). OPC-интерфейс не предоставляет механизмы для такого различия и поддерживает только квитирование возникновения аварий. Квитирование завершения аварийного состояния выполняется неявным образом сервером Alarms & Events.

3.7.8 Примеры работы с аварийными событиями

Аварийное событие без квитирования

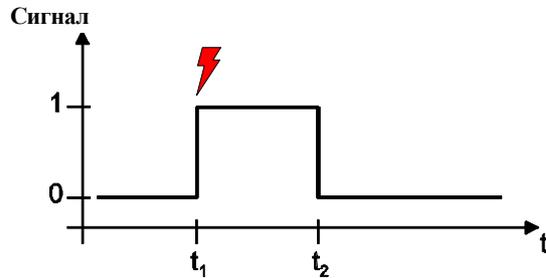


Рисунок 3–14 Состояния сигнала для аварийного события без квитирования

S7-блок следит за уровнем в резервуаре, заполняем материалом в ходе производственного процесса. Когда резервуар заполняется, S7-блок формирует аварийный сигнал (t_1) и процесс прекращается. Аварийное событие не нуждается в квитировании, производственный процесс прекращается без принятия контроллером каких-либо других мер. Как только контроллер обнаружит, что резервуар вновь опорожнен, он снимет аварийный сигнал (t_2) и производственный процесс возобновится.

Аварийное событие с квитированием

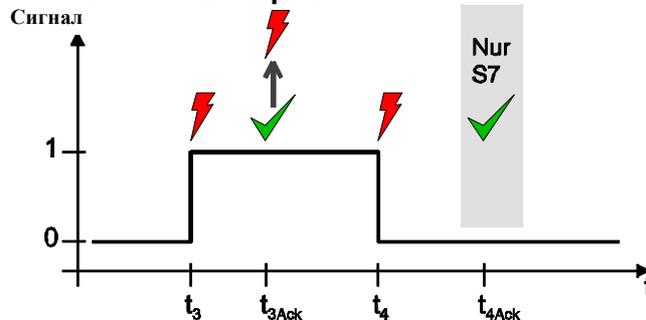


Рисунок 3–15 Состояния сигнала для аварийного события с квитированием

S7-блок следит за давлением в резервуаре. В случае превышения граничного значения S7-блок формирует аварийное событие (t_3) и одновременно включаются сигнальная лампа клапана избыточного давления и сирена.

Когда оператор квитирует аварию (t_{3Ack}), сирена выключается, хотя аварийное состояние остается, поскольку давление в резервуаре по-прежнему превышает граничное значение. Сигнальная лампа не выключается квитированием. Получив квитирование от оператора, контроллер S7 вновь генерирует аварийный сигнал.

После того, как давление будет снижено, и S7-блок определит, что значение не превышает верхнюю границу, он снимет аварийный сигнал (t_4).
Завершение аварийного состояния также приведет к формированию аварийного события.

Квитирование оператором завершения аварийного состояния приведет к выключению сигнальной лампы (t_{4Ack}). Это квитирование нельзя передать через OPC-интерфейс, поскольку OPC поддерживает только квитирование наступления аварийных состояний.

3.7.9 Какие интерфейсы используются для Alarms & Events?

Для Alarms & Events определены интерфейс автоматизации и пользовательский интерфейс:

- Стандарт Alarm & Events Automation Interface (Интерфейс автоматизации для доступа к аварийным сообщениям и событиям), 15 декабря, 1999г., версия 1.01.
Описание сервера OPC Alarms & Events, а также спецификация пользовательского интерфейса для этого сервера.
- OPC Alarms & Events Custom Interface (Пользовательский интерфейс для доступа к аварийным сообщениям и событиям, 2 октября, 2002г., версия 1.10).
Спецификация интерфейса автоматизации сервера OPC Alarms & Events.

3.8 Интерфейс OPC XML

Примечание

Интерфейс OPC XML предусматривается в составе программного обеспечения SIMATIC NET, начиная с версии 6.1.

OPC и доступ через Internet

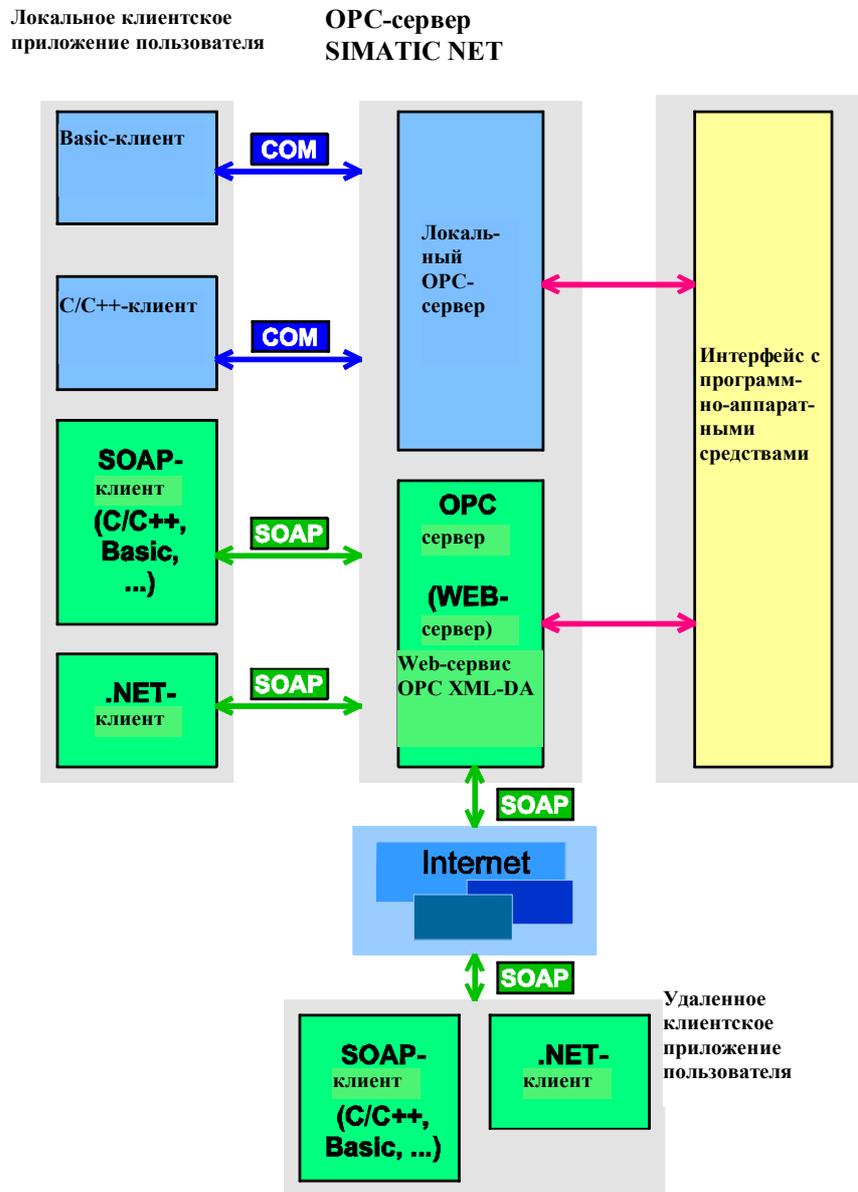
OPC–XML – это стандарт, который позволяет устанавливать связь по сети Internet с использованием протоколов межплатформенного обмена. Клиент отныне не ограничен средой Windows (COM). Контроль за и обмен данными OPC могут осуществлять и другие операционные системы, например, LINUX, используя протокол HTTP и интерфейс SOAP.

Для доступа к данным с помощью OPC XML имеется ряд функций, базирующихся на OPC Data Access, хотя предусмотрены только простые сервисы для записи и чтения. Из-за неустойчивости соединений в сети Internet передача сообщений, генерируемых по изменению состояния, предусмотренная, например, для интерфейсов DCOM OPC DA, для OPC XML не предусмотрена.

Одним из недостатком этого интерфейса является то, что для него необходимо наличие Internet-сервера. В сети Internet можно ожидать низкую производительность коммуникаций.

Интерфейсы OPC-сервера SIMATIC NET

На следующем рисунке показана внутренняя структура сервера SIMATIC NET OPC и предусмотренные для него интерфейсы:

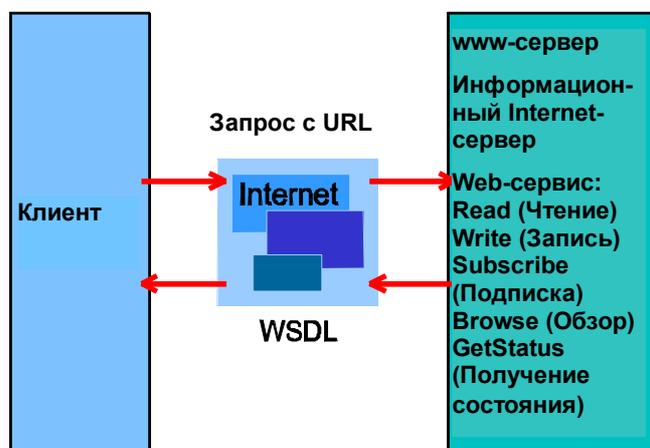


3.8.1 Web-сервис для OPC XML

Web-сервис информационного Internet-сервера

Спецификация OPC XML реализуется в SIMATIC NET посредством Web-сервиса, предоставляемого информационным Internet-сервером Microsoft (Microsoft Internet Information Server). Следует помнить о том, что Internet Information Server (IIS) является компонентом операционной системы, и должен устанавливаться и конфигурироваться отдельно.

Компонент OPC XML большей частью прозрачен для пользователя. Он запускается автоматически IIS-сервером, когда от Web-клиента поступают запросы на соответствующие сервисы OPC–XML. Взаимодействие в этом случае показано на рисунке ниже:



3.8.2 Простые сервисы для чтения/записи

Имеющиеся методы

Для сервера OPC XML SIMATIC NET предусмотрены следующие методы:

GetStatus (Прочитать состояние)

С помощью метода GetStatus можно прочитать общее состояние и специальную информацию изготовителя (версия, название продукта).

Read (Чтение)

С помощью сервиса Read можно прочитать значение одной или нескольких переменных.

Write (Запись)

Сервис Write осуществляет запись одной или нескольких переменных. Как вариант, сервер OPC–XML может после этого выполнить сервис Read (Чтение) и вставить полученные значения в задание на запись.

Subscription (Подписка), SubscriptionPolledRefresh (Подписка с обновлением по опросу), SubscriptionCancel (Отмена подписки)

В случае использования подписок выполняется регистрация переменных, после чего любые изменения читаются (циклически) с помощью метода SubscriptionPolledRefresh. Подписку можно отменить с помощью метода SubscriptionCancel.

Одним из свойств метода SubscriptionPolledRefresh является возможность задания временного окна с помощью параметров WaitTime и HoldTime. Вызов, отправляемый на сервер, удерживается в течение времени WaitTime (Время ожидания). Ответ на вызов возвращается, как только обнаруживается изменение значения, либо сразу по истечении времени HoldTime (Время удержания).

Browse (Обзор)

Сервис Browse (Обзор) обеспечивает навигацию по всей иерархии адресного пространства. В отличие от COM-интерфейса, здесь такой запрос можно использовать как для чтения "ветвей", так и для чтения "листьев".

Кроме того, имеется возможность указывать, какие свойства элемента должен возвращать сервер.

GetProperties (Прочитать свойства)

Вместо Browse (Обзор) для чтения свойств элемента можно использовать сервис GetProperties (Прочитать свойства).

Синхронное/асинхронное использование методов

Согласно спецификации OPC–XML DA эти методы являются асинхронными. Запрос и ответ являются отдельными частями протокола. Использование этих методов в языках программирования высокого уровня, например, в C# или Visual Basic и т.п., позволяет объединять запрос и ответ, то есть, создавать синхронный метод.

При создании класса "Ргоху" для клиентской программы для каждого метода генерируются и синхронный, и асинхронный варианты. Оба варианта используют одни и те же методы OPC–XML DA. Использование асинхронного варианта, однако, обладает преимуществами в режиме выполнения с точки зрения быстрогодействия клиентской программы.

Более подробную информацию можно найти в указанной ниже книге (Глава 9 Асинхронное программирование):

Professional ASP.NET Web Services
(Web-сервисы ASP.NET для профессионалов)
WROX Press Ltd
ISBN 1–861005–45–8

3.9 OPC в SIMATIC NET

Главным интерфейсом всех продуктов в составе PG/PC (программаторы и персональные компьютеры) семейства SIMATIC NET является OPC-интерфейс. OPC-сервер в составе SIMATIC NET поддерживает все коммуникационные протоколы и сервисы, предоставляемые коммуникационными модулями.

OPC-сервер SIMATIC NET поддерживает спецификацию интерфейса OPC Data Access (Доступ к данным через OPC) для всех протоколов. Для протоколов, предусматривающих механизмы сигнализации событий (S7-коммуникации) также поддерживается спецификация OPC Alarms & Events (Аварийные сообщения и события).

В следующих разделах будет показано, каким образом используется OPC-сервер семейства SIMATIC NET, будут представлены преимущества применения сервера и его характеристики. Вы также узнаете о способах оптимизации доступа к данным процесса с помощью OPC-сервера.

3.9.1 Что предоставляет OPC-сервер SIMATIC NET?

OPC-сервер семейства SIMATIC NET обеспечивает доступ к сетям промышленной связи PROFIBUS и Industrial Ethernet семейства SIMATIC NET. Он снабжает OPC-клиентов значениями переменных процесса или сигналами событий, источником которых является партнерское устройство. Он обращается к партнерским устройствам через коммуникационную сеть, используя ПО поддержки протоколов и коммуникационный процессор семейства SIMATIC NET (см. рисунок).



Рисунок 3-16 OPC-сервер SIMATIC NET и OPC-клиент

Специальные свойства

Поскольку OPC-сервер способен распределять задания между различными коммуникационными системами, OPC-клиент имеет возможность использовать несколько различных протоколов одновременно через один OPC-сервер. Если в конфигурации OPC-сервера предусмотрено применение только одного протокола, такое распределение не обязательно. При этом достигается оптимальная пропускная способность в канале связи.

Ограничения

OPC-сервер семейства SIMATIC NET поддерживает все интерфейсы, необходимые для спецификаций OPC Data Access и OPC Alarms & Events. Он также предоставляет большинство важных вспомогательных интерфейсов, таких как интерфейс обзора (Browsing) для OPC Data Access.

Следующие ограничения налагаются на вспомогательные интерфейсы:

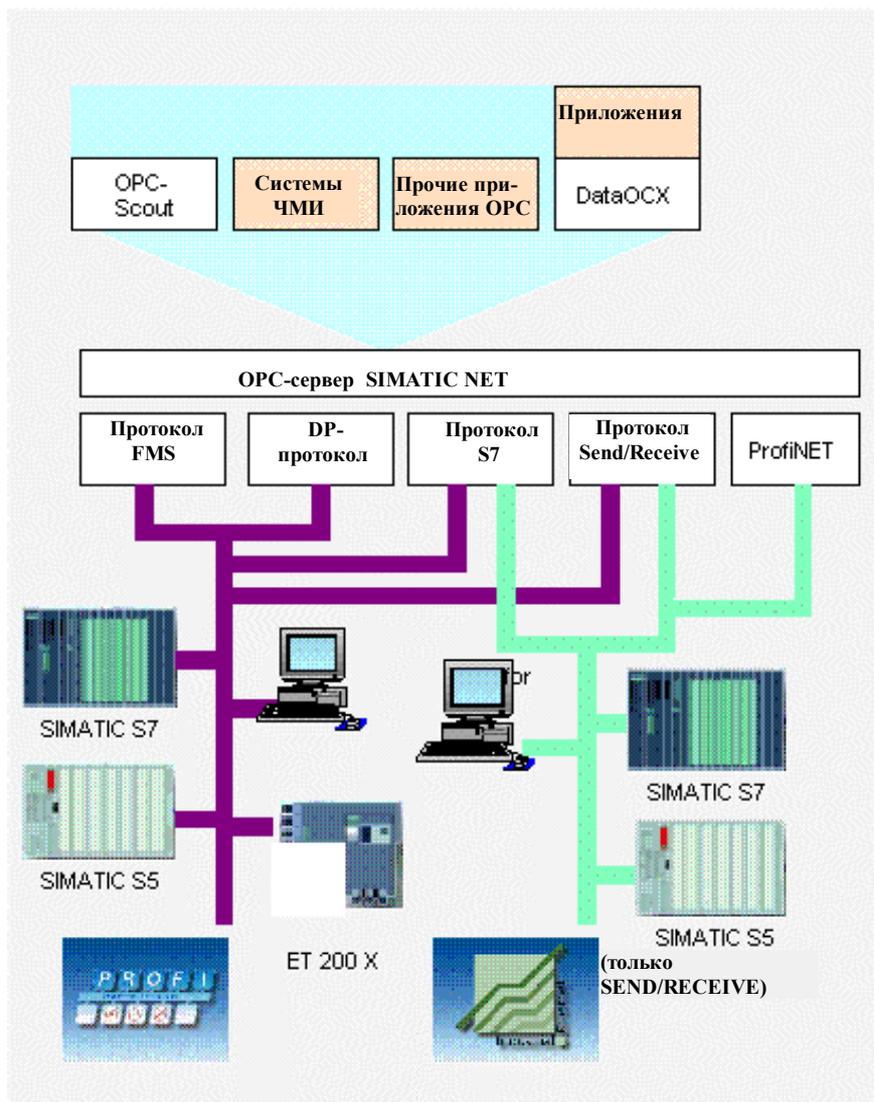
- OPC-сервер для Data Access не поддерживает “OPC Public Groups” (Общие группы OPC).
- OPC-сервер для Alarms & Events
 - Не поддерживается сигнализация событий, связанных с условиями
 - Квитирование событий не обязательно
 - Не поддерживается разделение производственного пространства на отдельные области (зоны)
 - Не поддерживается обзор областей (зон).

Тип сообщения и его содержимое спецификацией OPC не устанавливается. Сервер SIMATIC NET OPC Alarms & Events не содержит какой-либо конфигурационной информации относительно аварийных сообщений или событий, обрабатываемых системой операторского управления и визуализации. В этой версии OPC-сервер всего лишь доставляет простейшие события.

3.9.2 Как применяется OPC-сервер SIMATIC NET в мире автоматизации?

Применение OPC-сервера SIMATIC NET иллюстрируется на следующем рисунке. Сверху представлен ряд клиентов, которые могут обращаться к OPC-серверу.

В нижней части рисунка показаны коммуникационные сети и возможные коммуникационные партнеры.



Элементы этого цвета не являются частью SIMATIC NET

Рисунок 3–17 Роль OPC-сервера в мире автоматизации

3.9.3 Какие преимущества предоставляет семейству SIMATIC NET OPC-сервер?

Благодаря многочисленным достоинствам OPC-интерфейса семейство SIMATIC NET получает ряд преимуществ общего характера, а также преимущества в программировании и разработке отдельных клиентов. Не остался в стороне и персонал, выполняющий работы по вводу системы в эксплуатацию.

Преимущества при вводе систем в эксплуатацию

- Используется интерфейс, не привязанный к конкретному протоколу. Другими словами, для нескольких приложений устанавливается только один интерфейс.
- Предоставляется простой доступ к коммуникационным сетям SIMATIC NET.
- В коммуникационных сетях SIMATIC NET можно использовать программируемые контроллеры разных производителей, решающие многочисленные задачи автоматического управления.
- Появляется возможность интеграции с продуктами Microsoft Office.
- С помощью DCOM приложения, установленные на других компьютерах, могут обращаться к сервисам OPC-сервера через всемирные или локальные сети связи.
- Приложение "OPC Scout" в составе SIMATIC NET в роли OPC-клиента является мощным инструментом для простого доступа к переменным процесса.
- С помощью SIMATIC Computing можно, например, создавать простые служебные программы на Visual Basic.

Преимущества при программировании

- Пользователь работает с интерфейсом, не зависящим от производителя. Это означает сохранность вложенных средств. Рынок сбыта услуг расширяется, созданные программы можно использовать повторно.
- Разрабатываемые приложения не зависят от коммуникационных систем какого-либо одного производителя и могут обмениваться данными с OPC-серверами других производителей без каких-либо изменений.
- OPC-интерфейсы предоставляют приложениям эффективные средства доступа к OPC-серверам и нижестоящим коммуникационным системам.
- В OPC имеются высокопроизводительные интерфейсы для языков программирования C и C++.
- В таких средах разработки, как Visual Basic, возможен простой и удобный доступ к данным процесса посредством объектов Data Control (объект управления данными).
- Пользователю нет необходимости изучать специфические протоколы конкретных производителей.
- Возможность ведения журнала упрощает отладку и устранение ошибок.
- Поскольку имеется возможность имитации партнерского устройства, программу можно разрабатывать и отлаживать без установки дополнительных устройств.

3.9.4 Каков наилучший способ обращения к данным процесса?

С помощью OPC Data Access (Доступ к данным) можно обращаться к различным типам данных процесса. Выбирая соответствующие методы, пользователь может влиять на интенсивность потока данных приложения. Некоторые протоколы также позволяют влиять на производительность OPC-сервера, предоставляя возможность выбора сервиса и создания структуры переменных в пространстве имен.

Советы по достижению оптимальной пропускной способности

Следующие советы позволяют достичь максимально возможной пропускной способности (интенсивности потока данных):

- Используйте операции над группами
- Обращайтесь к OPC-кэшу
- Создавайте структуры элементов
- Используйте сервисы для передачи/приема буферов

Применение подходящих методов

Имеется ряд способов обращения к переменным. Поскольку они обладают различными характеристиками, следует иметь в виду преимущества и недостатки, перечисленные ниже, и выбирать наиболее подходящий тип доступа, в зависимости от ситуации.

- Используйте подходящие OPC-методы

3.9.5 Использование операций над группами

Многие методы позволяют передавать несколько переменных процесса в качестве параметров в едином массиве, используя один вызов функции. Применение групповых операций позволяет повысить пропускную способность, поскольку в этом случае OPC-клиент и OPC-сервер обмениваются меньшим количеством вызовов функций. Кроме того, OPC-сервер может самостоятельно оптимизировать связь по сети, например, объединяя вместе отдельные задания.

При работе с удаленным сервером с использованием DCOM применение групповых операций особенно эффективно, поскольку в этом случае вызов функции передается по сети.

Примечание

OPC-сервер также использует групповые операции при передаче клиенту сообщений о сменах состояний.

3.9.6 Примеры операций над группами

Ниже показаны примеры операций над группами, осуществляемых через пользовательский OPC-интерфейс:

IOPCItemMgt::AddItems (NumItems, pItem, ppResults, ppError)

В группу включены элементы, определенные в массиве pItemArray.

Примечание

Вызов AddItems (Добавить элементы) происходит быстрее, если группа не активна. Группу следует активизировать только после внесения в нее всех OPC-элементов.

IOPCSyncIO::Read (Source, NumItems, pServerhandle, ppResults, ppError)

Синхронное чтение значений, меток времени и кодов качества нескольких OPC-элементов, определенных в массиве, а также дескрипторов сервера.

IOPCSyncIO::Write (NumItems, pServerhandle, pItemValues, ppError)

Значения записываются синхронно в несколько OPC-элементов, определенных в массиве, вместе с дескрипторами серверов. Значения передаются в едином массиве.

3.9.7 Обращение к OPC-кэшу

OPC-кэш – это внутренний промежуточный буфер OPC-сервера, в который заносятся последние значения OPC-элементов, подлежащих чтению. OPC-сервер обновляет все активные элементы, вставленные в активные группы. Значения в кэше действительны при условии, что элемент был прочитан успешно.

Чтение переменной из кэша происходит существенно быстрее, по сравнению с обращением через сеть. Если значения в кэше обновляются с частотой, которая достаточна для определенного приложения, рекомендуется обращаться к кэшу.

Скорость обновления кэша указывается параметром *UpdateRate*.

Примечание

Следует помнить, что очень высокая скорость обновления (короткий интервал) оказывает большую нагрузку на процессор компьютера и систему связи.

3.9.8 Примеры сервисов, которые могут использовать кэш

Ниже показаны примеры сервисов, которые могут использовать кэш:

IOPCSyncIO::Read(...,OPC_DS_CACHE,...)

Синхронное чтение из кэша значений, меток времени и кодов качества нескольких OPC-элементов.

IOPCAsyncIO2::Refresh(...,OPC_DS_CACHE,...)

Независимо от значения для всех активных OPC-элементов формируется обратный вызов (callback). Клиенту передается значение, которое хранится в настоящий момент в кэше.

3.9.9 Структурирование элементов

В OPC-сервере семейства SIMATIC NET предусмотрен алгоритм оптимизации для сервисов обслуживания переменных с поддержкой следующих протоколов:

- Протокол S7
- S5–совместимые коммуникации через Industrial Ethernet

Несколько одновременно выставленных заданий на обращение к отдельным переменным преобразуются внутри OPC-сервера в единое задание на обращение к массиву переменных в партнерском устройстве. Это снижает количество передаваемых по сети пакетов данных, повышает эффективность использования пакетов и увеличивает "полезность" пакета.

Такая оптимизация применяется как для чтения, так и для записи и используется по умолчанию. Для OPC-клиента такой алгоритм оптимизации является "прозрачным".

Правила организации OPC-элементов в пространстве имен

Оптимизация возможна только в том случае, если для организации OPC-элементов в пространстве имен соблюдаются следующие правила:

- OPC-элементы, чтение или контроль которых происходит одновременно, должны располагаться в пространстве имен партнерского устройства последовательно, друг за другом.
Небольшие промежутки между соответствующими разделами допускаются, но это ухудшает пропускную способность.
- OPC-элементы, запись которых происходит одновременно, должны располагаться в пространстве имен последовательно, друг за другом.
Оптимальный доступ для записи достигается при отсутствии промежутков.
Массив, сформированный в процессе оптимизации, всегда передается партнерскому устройству целиком. Данные, расположенные по адресам, находящимся в промежутках, будут перезаписываться неопределенными значениями. Чтобы избежать этого, OPC-сервер передает после этого отдельные задания на доступ без какой-либо оптимизации.

Примечание

Если вашему OPC-клиенту не требуется использовать отдельные OPC-элементы для обработки переменных, в этом случае для обращения к соответствующим данным можно использовать непосредственно массивы, назначая элементы массива отдельным читаемым переменным процесса.

3.9.10 Пример структурирования элементов

Организация блока данных для чтения

DB10,W10

DB10,B12

DB10,B13

DB10,DW14

DB10,W20

Для чтения выполняется обращение к **DB10,B10,12** через систему связи. Несмотря на отсутствие в последовательности байта 18 и байта 19, последовательность преобразуется в задание на чтение. Ненужные прочитанные данные игнорируются. Вместо пяти отдельных запросов на чтение переменных по сети передается только один запрос.

Организация блока данных для записи

DB10,W10

DB10,B12

DB10,B13

DB10,DW14

DB10,W20

Обращение выполняется через систему связи за два сеанса: запись в массив DB10,B10,8 и в отдельную переменную DB10,W20.

3.9.11 Применение сервисов для передачи/приема буферов

Для пересылки больших пакетов данных в S7-коммуникациях и в S5-совместимых коммуникациях через Industrial Ethernet предусмотрены сервисы для передачи/приема буферов. В этом случае между коммуникационными партнерами передаются пакеты данных. Передача данных создает нагрузку в сети только в случае, когда партнер явным образом инициирует задание на передачу.

С помощью OPC-сервера семейства SIMATIC NET можно структурировать блоки данных. Это позволяет назначать OPC-элементам отдельные части пакета данных.

Примечание

S7-сервисы для передачи/приема буферов в настоящее время доступны только для устройств серии S7-400 и M7, а также для ПК-станций. S5-сервисы для передачи/приема буферов имеются практически для всех устройств серии S5 и S7 и ПК-станций. Сведения о дальнейшей поддержке буфер-ориентированных сервисов коммуникационным партнером можно найти в файле "readme", который находится в главной папке на диске SIMATIC NET Software Product CD.

3.9.12 Пример использования сервисов для передачи/приема буферов

На следующем рисунке показан принцип передачи устройством S7-400 пакета данных на ПК-станцию с S7 OPC-сервером.

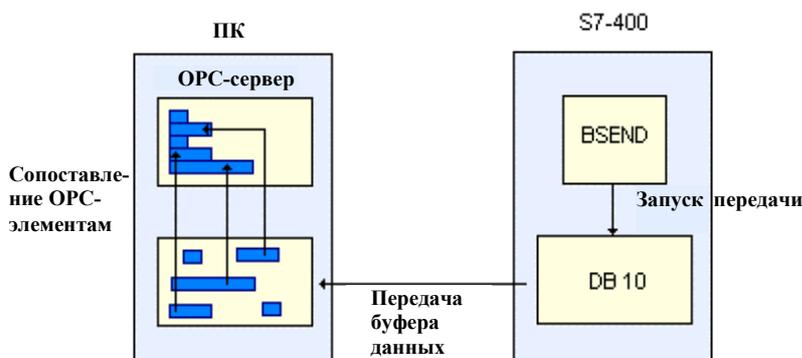


Рисунок 3-18 Передача пакета данных

Чтобы OPC-сервер мог принимать данные, в активную группу на ПК добавляются один или несколько OPC-элементов типа **RECEIVE** или **BRCV**. На станции S7-400 программа пользователя запускает функциональный блок **BSEND**. **BSEND** начинает передавать всю область данных целиком в виде буфера на ПК.

На ПК принятые данные поступают в OPC-сервер. OPC-сервер сопоставляет отдельные области блока данных соответствующим OPC-элементам. Если выполняется мониторинг этих OPC-элементов, OPC-сервер отправляет OPC-клиенту вызов в случае изменения значения.

3.9.13 Использование соответствующих OPC-методов

Для спецификации OPC Data Access предусмотрено три способа обращения к данным:

Синхронное чтение или запись

И чтение, и запись могут осуществляться синхронно.

В этом случае программа передает синхронный вызов функции для обращения к данным процесса. В процессе выполнения функции OPC-сервер управляет всеми коммуникациями в сети. Прочитанные значения передаются в программу пользователя в ответных параметрах функции, после чего приложение может продолжить выполнение следующих команд.

Применение синхронного доступа

Если допускаются продолжительные паузы в выполнении пользовательской программы (например, при последовательном выполнении отдельных программных модулей), всегда следует использовать синхронное обращение. Синхронное обращение является наиболее быстрым способом доступа к данным партнерского устройства.

Примечание

Пропускная способность в случае синхронных заданий не зависит от параметра UpdateRate (Скорость обновления) и продолжительности цикла, указанной в конфигурации.

Преимущества и недостатки синхронного доступа

Преимущества:

- Простота программирования
- Высокая пропускная способность, поскольку для одного задания требуется только один сеанс связи между OPC-клиентом и OPC-сервером.

Недостаток:

- Выполнение приложения прерывается до тех пор, пока не будет завершено синхронное задание. Приложение может продолжить работу только после того, как будут прочитаны все данные. Если функция не вызывается в отдельной ветви, в этом случае, например, во время вызова функции блокируется интерфейс пользователя прикладной задачи, предполагающей взаимодействие с оператором.

Асинхронное чтение или запись

Чтение или запись можно осуществлять асинхронно.

В этом случае программа передает асинхронный вызов функции для обращения к данным процесса. Функция сразу же возвращает программе уведомление об успешной передаче задания на OPC-сервер. После этого выполнение программы может быть продолжено.

OPC-сервер присваивает заданию идентификатор операции (TransactionID), с помощью которого клиент в дальнейшем может идентифицировать ответ на задание.

После этого в некоторый произвольный момент времени OPC-сервер вызывает функцию OPC-клиента **AsyncReadComplete** или **AsyncWriteComplete**. Клиенту в качестве параметра вызова передаются результаты предшествующего вызова функции (чтения или записи) и значение TransactionID.

Выполнение программы OPC-клиентом никак не влияет на время, в течение которого данные передаются по сети.

Применение асинхронного доступа

Асинхронное обращение к данным может быть полезно, когда предусматривается чтение больших объемов данных и приложение должно работать во время выполнения задания.

Асинхронный доступ к кэшу OPC-сервера не имеет практического смысла. Процессор оказывается сильно нагруженным сеансами связи между OPC-клиентом и OPC-сервером.

Примечание

Пропускная способность в случае асинхронных заданий не зависит от параметра UpdateRate (Скорость обновления) и длительности цикла, указанной в конфигурации.

Преимущества и недостатки асинхронного доступа

Преимущества

Локальное приложение прерывается лишь на короткое время, поскольку все коммуникации происходят одновременно с работой этого приложения.

Недостатки

- Программирование приложения несколько усложняется. В приложении должен быть реализован механизм обработки ответов, способный принимать результат выполняемого задания в любое время. В стандартных программах Windows предусмотрены механизмы асинхронного обращения к данным, позволяющие им реагировать на ввод информации пользователя.
- В случае передачи только нескольких переменных в одном задании процессор оказывается слишком сильно нагруженным сеансами связи "вызов – ответ". Их количество удваивается по сравнению с синхронным обращением.

Мониторинг переменных

В режиме мониторинга переменных OPC-сервер постоянно проверяет, не изменилось ли значение или код качества переменной.

Это означает, что OPC-клиент добавляет активные OPC-элементы в группу и активизирует группу. После этого контролируются все активные OPC-элементы во всех активных группах.

OPC-клиент предоставляет функцию DataChange (Изменение данных). OPC-сервер вызывает эту функцию, когда значение изменилось. В качестве параметров функции DataChange OPC-сервер передает измененные значения, коды качества и метку времени OPC-элементов.

Мониторинг переменных не оказывает дополнительную нагрузку на OPC-клиента. Программа клиента выполняется только в случае обнаружения изменения.

Чтобы OPC-клиент не был полностью перегружен сообщениями в случае, когда переменные процесса изменяются слишком быстро, можно использовать специальный параметр группы *UpdateRate* (*Скорость обновления*), чтобы задать минимальный интервал вызова клиента.

Воздействие помех на аналоговые входы приведет к возникновению потока сообщений об изменении состояния, поскольку значения будут изменяться все время. Помехи можно подавить, используя специальный параметр группы *PercentDeadBand* (*Уровень зоны нечувствительности*). Благодаря этому параметру сигнализируются только такие изменения уровня сигнала, которые превышают порог, определяемый в процентах от всего диапазона.

Такая фильтрация возможна только тогда, когда в конфигураторе символьных файлов (Symbol File Configurator) определен диапазон значений переменной.

Применение мониторинга переменных

Мониторинг переменных является идеальным решением, когда программе непрерывно требуется получать актуальные данные о процессе или части процесса.

Примечание

Интенсивность потока данных определяется параметром группы UpdateRate (Скорость обновления) и длительностью цикла, указанной в конфигурации. Длительность цикла определяет наименьший возможный интервал обновления. Скорость обновления должна указываться как кратное значение длительности цикла.

Преимущества и недостатки мониторинга переменных

Преимущества

- Уведомление передается приложению только в случае изменения данных процесса.
- Благодаря небольшому количеству сеансов связи обеспечивается высокая пропускная способность. В зависимости от способа структурирования элементов возможно достижение хорошей оптимизации.
- Клиент может включать и выключать мониторинг переменных для отдельного элемента и для группы.

Недостатки

- Время реакции с момента изменения значения процесса, до передачи нового значения клиенту, превышает интервал обновления для группы.
- Написание программы несколько усложняется. В программе необходимо предусмотреть асинхронный прием измененных значений.

В стандартных программах Windows предусмотрены механизмы асинхронной обработки информации, вводимой пользователем.

Взаимосвязь между длительностью цикла и скоростью обновления

Параметр "UpdateRate" (Скорость обновления), который можно задать в программе пользователя, определяет наиболее короткий возможный интервал проверки значений OPC-элементов активной OPC-группы. По истечении этого интервала сервер проверяет, произошли ли изменения активных OPC-элементов.

Скорости обновления, используемые OPC-сервером SIMATIC NET, являются кратными значениями длительности цикла, указанной на этапе конфигурирования. Максимальная скорость обновления (наиболее короткий интервал) совпадает с длительностью цикла.

Взаимосвязь между заданными длительностями цикла для конкретного протокола

Поскольку OPC-сервер семейства SIMATIC NET может одновременно использовать переменные различных протоколов, максимальная скорость обновления OPC-сервера – это наибольшее из значений, установленных в качестве длительности цикла опроса для активных протоколов (для тех протоколов, для которых сконфигурированы соединения).

4 OPC-переменные процесса для SIMATIC NET

В этой Главе поясняется синтаксис имен переменных процесса. Синтаксис имен переменных процесса OPC-интерфейса предусматривает информацию об адресах переменных процесса. Имена переменных выбираются при создании программы или конфигурировании OPC-клиента.

4.1 Поддерживаемые функции связи

OPC-сервер обеспечивает унифицированный доступ к сетям промышленной связи SIMATIC NET посредством интерфейса доступа к данным (Data Access).

OPC-сервер SIMATIC NET обеспечивает взаимодействие приложений с любыми компонентами автоматизации, включенными в сеть PROFIBUS или Industrial Ethernet. Он поддерживает следующие функции связи:

- S7- коммуникации
- S5- совместимые коммуникации
- PROFIBUS–DP
- PROFIBUS–FMS

Дополнительные функции связи находятся на стадии разработки.

Функции связи специально оптимизированы под различные требования.

OPC-сервер может одновременно поддерживать несколько функций связи.

4.2 Понятие переменной процесса

Переменная процесса – это доступный для записи и/или чтения элемент данных, представляющий входную/выходную физическую величину процесса, например, температуру в резервуаре, подаваемую на вход программируемого контроллера.

Переменные процесса представлены в модели классов OPC классом "OPC item" (OPC-элемент). Физические значения процесса в OPC представляются только элементами этого класса.

Идентификаторы элементов

Идентификатор элемента (ItemID) – это строка данных, однозначно идентифицирующая уникальную переменную процесса. Она информирует сервер о том, какая переменная процесса назначена OPC-элементу. Обращение к переменной процесса возможно лишь через OPC-элемент. OPC-сервер семейства SIMATIC NET назначает различные коммуникационные сервисы протоколов, оперирующих OPC-элементами, используя соответствующие сегменты идентификатора ItemID в качестве параметров для вызова коммуникационной функции.

4.3 Синтаксис идентификаторов (ItemID) переменных процесса

Переменные процесса описываются в OPC-интерфейсе уникальным именем-идентификатором элемента (ItemID). ItemID состоит из следующих сегментов:

Синтаксис

`<protocolID>:[<connectionname>]<variablename>`

Пояснения

<protocolID>

Выбор протокола для обращения к переменной процесса.

Существуют следующие идентификаторы протоколов:

DP	Протокол DP, включая поддержку DP-ведущего, DP-ведомого и DPC1
S7	S7 функции через PROFIBUS и Ethernet
FMS	Протокол FMS через PROFIBUS
SEND/RECEIVE	S5-совместимые коммуникации через Industrial Ethernet
FDL	S5-совместимые коммуникации через PROFIBUS

<connectionname>

Имя соединения идентифицирует соединение или коммуникационный модуль, через который будет происходить обращение к коммуникационному партнеру (например, к ПЛК, другим ПК-станциям или ведущим устройствам DP). Имя соединения указывается в STEP 7 / SIMATIC NCM PC в процессе конфигурирования аппаратных средств.

Имя соединения зависит от используемого протокола. В пределах протокола имя соединения должно быть уникальным.

<variablename>

Адресуемая переменная.

Имя переменной должно быть уникальным в пределах соединения, указанного с помощью имени соединения. Структура имени переменной зависит от выбранного протокола.

Примечание

Начиная с версии 6.0 OPC-сервера, синтаксис идентификаторов элементов был упрощен. Тем не менее, предшествующий синтаксис по-прежнему поддерживается. Программы и конфигурации, в которых используются предшествующие идентификаторы элементов, можно использовать без каких-либо изменений.

С другой стороны, старый синтаксис больше не отображается в окне OPC-браузера OPC-сервера. В дальнейших проектах следует использовать новый упрощенный синтаксис.

4.4 PROFIBUS–DP

Переменные процесса для ведущего устройства DP

OPC-сервер SIMATIC NET в режиме ведущего устройства DP предоставляет переменные процесса для следующих сервисов:

- Сервисы для ведущего устройства класса 1
Доступ и мониторинг DP-входов и выходов
- Сервисы DPC1
Ациклическая передача блоков данных
- Быстрая Логика (Fast Logic) для CP 5613 и CP 5614 (только для ведущего устройства DP)
Автоматический мониторинг данных ведомого устройства
- Диагностические переменные
Анализ статической диагностической информации

Переменные процесса для ведомого устройства DP

OPC-сервер SIMATIC NET для ведомого устройства DP предоставляет переменные процесса для следующих сервисов:

- Сервисы обслуживания переменных для доступа к локальным данным ведомого устройства
Обращение к входам и выходам ведомого устройства
- Диагностические переменные
Анализ статической диагностической информации ведомого устройства

Примечание

Модуль CP 5614 может работать одновременно в режиме ведущего и ведомого устройства DP только в том случае, если в SIMATIC NCM PC или в STEP 7 во время конфигурирования был выбран режим DP Base.

4.4.1 Идентификатор протокола

Для *протокола DP* используется идентификатор протокола (protocol ID) **DP**.

4.4.2 Имена соединений

Имя соединения определяет путь доступа, указывая коммуникационный процессор, включенный в сеть PROFIBUS.

В протоколе DP именем соединения является сконфигурированное имя коммуникационного модуля.

Примеры имен соединения

Типичные имена соединений:

CP 5613

CP 5614

CP 5611

4.4.3 Высокопроизводительный внутрипроцессный сервер (SIMATIC NET Inproc-Server) для протокола PROFIBUS-DP

В случае применения протокола PROFIBUS DP в коммуникационном процессоре компьютера содержится образ входных и выходных данных. Поэтому обращение к данным процесса происходит непосредственно в пределах локального компьютера. В частности, в случае использования SIMATIC NET CP 5613 через интерфейс "DP-Base", обращение к данным происходит чрезвычайно быстро.

В некоторых случаях, например, когда применяются контроллеры на базе ПК, необходимо обеспечить очень быстрый доступ к данным процесса.

В случае применения OPC, основанного на архитектуре "клиент-сервер" на базе технологии COM, выполнение внутренних процессов занимает определенное время, которое зависит от способа реализации OPC-сервера. Для локального сервера (который также называют "внепроцессным сервером"; EXE-файл с собственным пространством процесса) затраты времени, главным образом, связаны с переключением между процессами и передачей параметров функций от клиента к серверу (маршаллинг).

Если OPC-сервер реализован как внутрипроцессный сервер (in-proc server), чрезмерных затрат времени на переключение процессов и маршаллинг удается избежать, поскольку OPC-сервер реализуется как библиотека динамической компоновки (DLL) и работает в пространстве процесса клиента.

Применение внутрипроцессного сервера, однако, имеет ряд недостатков, которые нельзя игнорировать при выборе сервера:

- Одновременно только один клиент может работать с сервером.
Одновременное использование внутрипроцессного OPC-сервера несколькими клиентами будет означать создание нескольких серверов в различных пространствах процесса и будет приводить к одновременному нескоординированному обращению к одним и тем же аппаратным средствам. В результате доступ к данным процесса получит только тот клиент, который начнет работать раньше остальных, тогда как другим клиентам в доступе будет отказано.
- Стабильность работы OPC-сервера зависит от клиента.
В случае утраты контроля над OPC-клиентом, например, в случае некорректного доступа к данным со стороны клиента, также нарушается работа OPC-сервера. В результате коммуникационный модуль не сможет произвести сброс OPC-сервера, что может потребоваться в ряде случаев. Непосредственное прекращение работы OPC-сервера с помощью программ конфигурирования также будет невозможно.

Для поддержки чрезвычайно быстрого DP-протокола в SIMATIC NET предусмотрен внутрипроцессный сервер, который обеспечивает практически полную функциональность DP-протокола для OPC-клиентов.

В случае применения других протоколов, ориентированных на установление соединений, время доступа зависит, главным образом, от времени распространения сигналов в сети, необходимого для работы коммуникационных сервисов, поэтому внутрипроцессный сервер предусмотрен только для протокола DP.

Вызов внутрипроцессного сервера для протокола DP

В целях обращения к внутрипроцессному серверу для протокола DP предусмотрен отдельный идентификатор (ProgID):

OPC.SimaticNET.DP

Сервер выбирается путем указания ProgID OPC-клиентами или в вызовах функций.

Примеры вызовов функций

Visual Basic:

```
ServerObj.Connect ("OPC.SimaticNET.DP")
```

Visual C++:

```
r1 = CLSIDFromProgID(L"OPC.SimaticNET.DP", &clsid);
```

```
r1 = CoCreateInstance (clsid, NULL, CLSCTX_INPROC_SERVER,  
IID_IOPCServer, (void**)&m_pIOPCServer);
```

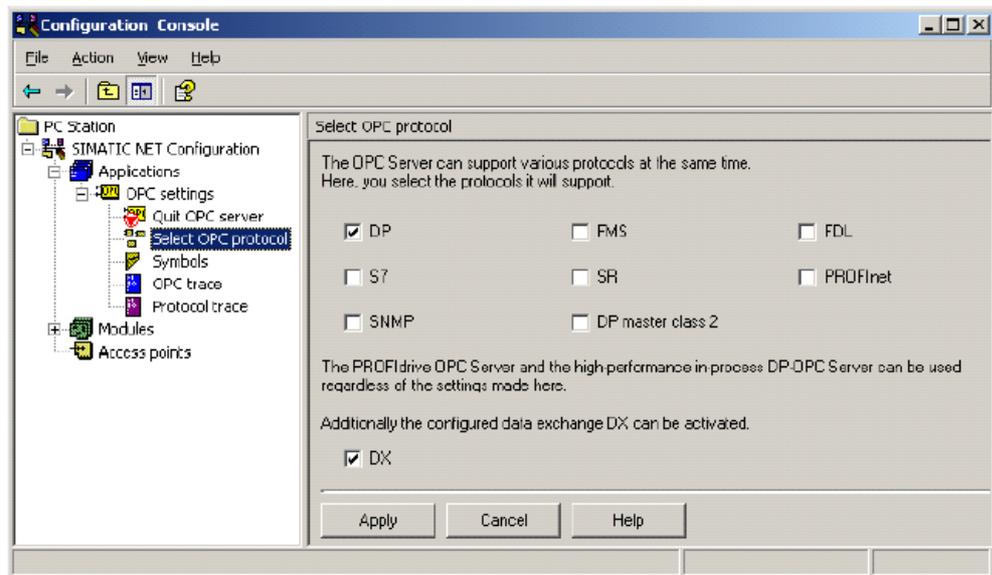
4.4.4 Мощный OPC-сервер SIMATIC NET для протокола PROFIBUS-DP

Высокая производительность даже при работе с несколькими клиентами

Как описывалось в предыдущем разделе, высокопроизводительный внутрипроцессный сервер для PROFIBUS-DP можно использовать лишь с одним клиентом. Если требуется использовать внутрипроцессный сервер для двух или большего количества клиентов при сохранении прежней производительности, можно использовать второй вариант конфигурации. С целью создания такой конфигурации все базовые библиотеки DP-протокола, а также COM-сервер в качестве внутрипроцессного сервера загружаются на внепроцессный OPC-сервер. Управление протоколом происходит внутри процесса OPC-сервера, что позволяет избежать дополнительных затрат времени, связанных с переключением между процессами и реализацией мультипротокольного режима. Переключение между процессами OPC-клиента и OPC-сервера, однако, по-прежнему необходимо.

Конфигурирование

Этот вариант повышенной производительности вступает в силу автоматически в результате выбора DP-протокола в программе Configuration Console ("Консоль конфигурирования").



Дополнительно можно активизировать сконфигурированный обмен данными (DX) через OPC и применение символов.

Преимущества / Недостатки

Применение высокопроизводительного OPC-сервера для протокола DP обладает недостатком, который состоит в том, что в этом случае может быть реализован только одно-протокольный режим (DP). С другой стороны, это позволяет получить следующие преимущества:

- Более высокая производительность по сравнению с мультипротокольным режимом.
- Простое конфигурирование.
- Для обращения используется идентификатор (ProgID) *OPC.SimaticNET*.
- Несколько клиентов могут использовать сервер одновременно.
- Стабильность работы OPC-сервера не зависит от клиентов.

4.4.5 Сервисы DPC1 и DPC2

Ведущее устройство DP класса 1

Ведущее устройство DP класса 1 обменивается данными с DP-ведомыми циклически.

В случае использования DPC1 ведущее устройство, осуществляющее циклический обмен, может обрабатывать дополнительный ациклический трафик данных и реагировать на прерывания со стороны ведомых устройств.

Коммуникации охватывают такие базовые функции, как:

- Конфигурирование и назначение параметров ведомым устройствам
- Циклический обмен данными с ведомыми устройствами DP
- Мониторинг ведомых устройств DP
- Предоставление диагностической информации

Ведущее устройство DP класса 2

В случае применения DPC2 ведущее устройство DP класса 2 может установить дополнительные коммуникационные контакты (соединения) с ведомыми устройствами DP. Ведомые устройства DP должны поддерживать дополнительные функции DP-V1. Наиболее важными дополнительными функциями DPC2 являются:

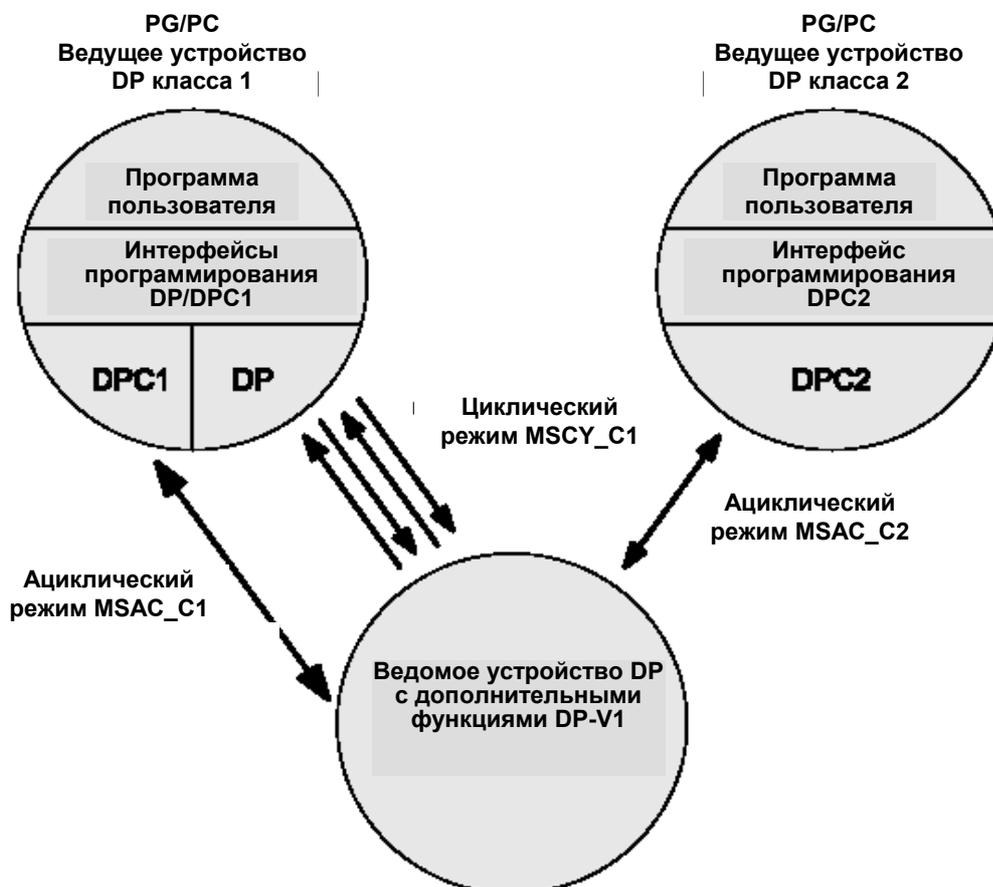
- Установление соединения
- Разрыв соединения
- Чтение "записей" данных
- Запись "записей" данных
- Передача данных

Ведомые устройства с дополнительными функциями DP-V1

Ведомые устройства с дополнительными функциями DP-V1 могут обмениваться данными с ведущими устройствами DP класса 1 и класса 2.

Обзор протокола DP

На следующем рисунке наглядно показаны отдельные составляющие протокола DP или DP-V1.



4.4.6 Переменные процесса для сервисов ведущего устройства класса 1

С помощью сервисов для доступа к циклическим данным можно обращаться к входам и выходам ведомых устройств и контролировать их.

При обращении указывается:

- Номер ведомого устройства.
Этот номер соответствует адресу PROFIBUS.
- Номер submodule.
Ведомое устройство DP может состоять из нескольких submodule с различными пространствами ввода/вывода.
- Пространство (область) ввода/вывода.

4.4.7 Синтаксис переменных процесса для ведущего устройства класса 1

Синтаксис

Входы:

DP:[<connectionname>]**slave**<address>{**M**<number>}_**I**{<format>
<offset>{.<bit>}{,<quantity>}}

Выходы:

DP:[<connectionname>]**slave**<address>{**M**<Nummer>}_**Q**{<format>
<offset>{.<bit>}{,<quantity>}}

Пояснения

DP

Протокол DP для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

slave

Признак обращения к ведомому устройству DP.

<address>

PROFIBUS-адрес ведомого устройства.

Диапазон 0 ...126

M

Обозначение номера submodule, которому принадлежит пространство входов или выходов.

<number>

Номер submodule, которому принадлежит пространство входов или выходов.

_I

Обозначение входа. Входы могут быть только прочитаны.

_Q

Обозначение выхода. Выходы могут быть прочитаны и записаны.

<format>

Формат доставляемых данных.

Тип данных определяется выбранным форматом.

В принципе, интерфейс автоматизации и пользовательский интерфейс OPC позволяют выполнять чтение всех перечисленных типов данных OLE. Тем не менее, некоторые средства разработки, например, Visual Basic, поддерживают ограниченное количество типов данных.

По этой причине в следующей таблице перечислены соответствующие типы данных Visual Basic, в которых может быть представлено значение переменной.

Идентификатор формата	Описание	Тип данных OLE	Тип в Visual Basic
X	Бит	VT_BOOL	Boolean
BYTE или B	Байт (8 битов без знака)	VT_UI1	Byte
CHAR	Символ (8 битов со знаком)	VT_I1	Integer
WORD или W	Слово (16 битов без знака)	VT_UI2	Long
INT	Целое (16 битов со знаком)	VT_I2	Integer
DWORD или D	Двойное слово (32 бита без знака)	VT_UI4	Double
DINT	Целое, двойное слово (32 бита без знака)	VT_I4	Long
REAL	Число с плавающей запятой (4 байта, IEEE)	VT_R4	Single

<Offset>

Позиция байта (смещение) в адресном пространстве ведомого устройства, в которой расположен адресуемый элемент. Если был указан субмодуль, смещение относится к адресному пространству субмодуля. Если субмодуль не указан, смещение относится ко всему пространству входов/выходов ведомого устройства.

<Bit>

Номер бита в адресуемом байте.

Указать бит можно только для идентификатора формата **X**, диапазон: 0...7.

<quantity>

Количество элементов.

Тип данных "VT_ARRAY" переменной соответствует массиву элементов указанного формата. Если параметр *quantity* не указан, предполагается, что количество элементов =1 и переменная не является массивом.

Не следует использовать параметр *quantity*, если указан идентификатор формата **X**.

Примечание

Если не указан submodule, переменная возвращает все пространство входов или выходов для всех submodule. Также можно структурировать доступ, указывая формат и количество.

4.4.8 Примеры переменных процесса для ведущего устройства класса 1

Ниже приведены примеры, иллюстрирующие синтаксис названий переменных DP-протокола.

Входы

DP:[CP 5613]Slave005M003_IB0

Slave005M003_IB0

Входной байт 0 submodule 3 ведомого устройства 5.

DP:[CP 5613]Slave005M003_IB1,3

Slave005M003_IB1,3

Массив из 3 байтов, начиная с входного байта 1 submodule 3 ведомого устройства 5.

DP:[CP 5613]Slave005M003_ID2

Slave005M003_ID2

Двойное слово, начиная с входного байта 2 submodule 3 ведомого устройства 5.

DP:[CP 5613]Slave004M003_IReal0

Slave004M003_IReal0

Число с плавающей запятой в пространстве входов ведомого устройства 4, submodule 3.

DP:[CP 5613]Slave004_IB0,8

Slave004_IB0,8

Первые 8 байтов всего пространства входов ведомого устройства 4, по всем submodule.

Выходы

DP:[CP 5613]Slave005M007_QB1

Slave005M007_QB1

Выходной байт 1 submodule 7.

DP:[CP 5613]Slave005M007_QX2.5

Slave005M007_QX2.5

Бит 5 в выходном байте 2 ведомого устройства 5, submodule 7.

DP:[CP 5613]Slave004_QW0,8

Slave004_QW0,8

Массив из 8-ми слов, начиная с пространства выходов ведомого устройства 4, по всем submodule.

4.4.9 Сервисы DPC1

Используя сервисы DPC1, вы можете обращаться к "записям" данных ведомых устройств DPC1. "Записи" данных передаются ациклически. Смысловое значение "записей" данных определяется изготовителем ведомого устройства. В них, например, могут содержаться параметры для электропривода.

При регистрации переменной DPC1 OPC-сервер может проверить лишь правильность синтаксиса, но на основании конфигурации ведомого устройства DPC1 он не может проверить действительность переменной на стороне партнерского устройства и соответствие длины "записи" данных на основании конфигурации ведомого устройства DPC1.

4.4.10 Синтаксис переменных процесса для сервисов DPC1

Синтаксис

DP:*[<connectionname>]slave<address>S<slot>data<index>, <length>{,<subarea>}*

Пояснения

DP

Протокол DP для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

slave

Признак обращения к ведомому устройству с помощью протокола DP.

address

PROFIBUS-адрес ведомого устройства.

Диапазон: 0...126

S

Обозначение слота ведомого устройства, как правило, для submodule.

<slot>

Слот (участок) в области расширенной памяти ведомого устройства для ациклических сервисов. Слот и индекс ("указатель") идентифицируют "запись" данных.

data

Признак обращения к "записи" данных.

<Index>

Индекс в пределах слота в области расширенной памяти ведомого устройства для ациклических сервисов. Слот и индекс идентифицируют "запись" данных.

<length>

Длина "записи". Диапазон: 1...244.

<subarea>

Подобласть состоит из следующих атрибутов:

formatoffset{.bit}{,quantity}

format

Формат доставки данных.

Если формат не указан, используется формат **byte**.

Идентификатор формата	Описание	Тип данных OLE	Тип в Visual Basic
X	Бит	VT_BOOL	Boolean
BYTE или B	Байт (8 битов без знака)	VT_UI1	Byte
CHAR	Символ (8 битов со знаком)	VT_I1	Integer
WORD или W	Слово (16 битов без знака)	VT_UI2	Long
INT	Целое (16 битов со знаком)	VT_I2	Integer
DWORD или D	Двойное слово (32 бита без знака)	VT_UI4	Double

Идентификатор формата	Описание	Тип данных OLE	Тип в Visual Basic
DINT	Целое, двойное слово (32 бита без знака)	VT_I4	Long
REAL	Число с плавающей запятой	VT_R4	Single

offset

Адрес байта в пределах "записи" данных для адресуемого элемента.

bit

Номер бита в адресуемом байте.

Бит можно указать только для идентификатора формата X.

quantity

Количество элементов (не используется в случае формата X).

Тип данных VT_ARRAY соответствует массиву из элементов указанного формата. Если это часть имени не указана, предполагается, что количество элементов =1 и переменная не является массивом.

Примечание

Размер данных, определяемый параметрами *quantity* (количество элементов) и *format* (формат), не должен превышать размер "записи" данных в ведомом устройстве. Размер "записи" данных зависит от конкретного ведомого устройства и не может быть проверен OPC-сервером.

Примечание

Если были определены подобласти, необходимо помнить о следующем: если производится чтение "записи" данных, партнерское устройство сначала всегда читает всю "запись" данных целиком, и лишь после этого анализируются отдельные подобласти.

Когда выполняется запись, на партнерское устройство передается вся "запись" данных целиком. Если производится запись в несколько подобластей с использованием группового вызова, в "запись" данных сначала вносятся все подобласти, и лишь после этого "запись" данных передается.

Вследствие сказанного необходимо группировать все OPC-элементы, для которых выполняются отдельные обращения, в одну "запись" данных и записывать всю группу целиком. Следует избегать перекрытия данных или промежутков, поскольку в противном случае нельзя предугадать, какое значение будет записано.

4.4.11 Примеры переменных процесса для сервисов DPC1

Ниже приведены примеры, иллюстрирующие синтаксис имен переменных для сервисов DPC1.

Имена переменных для DPC1

DP:[CP 5613]Slave005S003Data2,120,DWORD7

Slave005S003Data2,120,DWORD7

Обращение к двойному слову, начиная с 7-го по счету байта в "записи" данных, состоящей из 120-ти байтов, соответствующей слоту 3, индексу 2 ведомого устройства 5.

DP:[CP 5613]Slave005S003Data2,120,B8,4

Slave005S003Data2,120,B8,4

Обращение к массиву из 4-х байтов, начиная с 8-го по счету в "записи" данных, состоящей из 120-ти байтов, соответствующей слоту 3, индексу 2 ведомого устройства 5.

4.4.12 Быстрая логика (Fast Logic) для CP 5613 и CP 5614 (только для режима ведущего устройства)

CP 5613 и CP 5614 (только в режиме ведущего устройства) поддерживают функцию Fast Logic ("быстрая логика").

Это означает, что параметры CP можно настроить таким образом, чтобы CP производил запись значений в другие ведомые устройства в качестве реакции на изменение значений данных ведомого устройства.

Приложение пользователя также оповещается об изменении значений данных. В CP 5613/5614 предусмотрено четыре условия запуска Fast Logic, которые можно конфигурировать и анализировать с использованием управляющих OPC-переменных.

Преимущества "быстрой логики"

Использование Fast Logic ("быстрой логики") обладает следующими преимуществами:

- OPC-сервер и OPC-клиент испытывают меньшую нагрузку.
- Передача данных происходит быстрее, поскольку она не зависит от работающего на CP программного обеспечения и использует ресурсы аппаратных средств самого CP.
- Поскольку передача данных не зависит от программного обеспечения ПК, в этом случае гарантируется быстрая реакция на изменения входного сигнала.

Примечание

Функцию Fast Logic ("быстрая логика") можно применять только, если при конфигурировании CP 5613 или CP 5614 был выбран режим DP Base.

Примечание

Активизированный режим Fast Logic в дальнейшем будет автоматически отменен. Поэтому режим Fast Logic вновь следует активизировать с помощью переменной FLActivate.

Функция Fast Logic работает корректно только в том случае, если ведущее устройство DP находится в режиме OPERATE, а соответствующие ведомые устройства находятся в состоянии READY. Следовательно, режим Fast Logic должен активизироваться DP-приложением только тогда, когда программа пользователя перевела ведущее устройство DP в режим OPERATE.

Пока режим Fast Logic является активным, DP-программа пользователя не может осуществлять запись в выходные байты, связанные с входными байтами функцией Fast Logic.

4.4.13 Синтаксис управляющих переменных для Fast Logic

Синтаксис

DP:[<connectionname>]**FL**<parameter><N>

Пояснения

DP

Протокол DP для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

FL

Обозначение режима Fast Logic.

<parameter>

Можно указать одно из следующих значений:

State

Возвращает статус режима Fast Logic.

Возвращаемые значения:

- CLEARED (Сброшен)
Флаг (условие) *N* не установлен.
- ACTIVATED (Активизирован)
Флаг (условие) *N* установлен.
- TRIGGERED (Выполнен)
Флаг (условие) *N* был установлен и мониторинг был выполнен.

Тип данных OLE	Тип в Visual Basic
VT_BSTR	String (строка)

Activate

Параметр Activate можно только записать.

Запись значения в поле параметра позволяет указать функцию Fast Logic для флага (условия), указанного в ItemID.

Записываемое поле параметр является массивом из 8-ми байтов и имеет следующую структуру:

slave_addr_in_byte

Адрес ведомого устройства, входы которого выбраны в качестве условия запуска.

index_in_byte

Смещение входного байта условия запуска.

cmp_value_in_byte

Сравнение значения с входным байтом.

mask_in_byte

Отдельные биты входного байта можно маскировать, чтобы не учитывать их при сравнении. Бит маскируется, если для него выбрано значение **1**. Это означает, что если выбрано *mask_in_byte*==0x00, при сравнении будут использоваться все биты *cmp_value_in_byte*. Условие запуска будет выполнено, когда все немаскированные биты в выбранном входном байте совпадут с битами в *cmp_value_in_byte*.

slave_addr_out_byte

Выбор ведомого устройства, выходной байт которого будет изменяться при выполнении условия запуска.

index_out_byte

Смещение выходного байта.

value_out_byte

Значение выходного байта, которое будет записано.

mask_out_byte

Отдельные биты выходного байта можно маскировать, чтобы их состояние не изменялось при выполнении условия запуска. Бит маскируется, если для него установлено значение 1. Это означает, что если выбрано *mask_out_byte*==0x00, все биты в *value_out_byte* будут записаны в выбранный выходной байт.

Тип данных OLE	Тип в Visual Basic
Массив VT_ARRAY из элементов типа VT_UI1	Byte() (Байт)

Clear

Параметр Clear можно только записать.

Если записывается логическое значение TRUE (верно), флаг запуска Fast Logic, указанный идентификатором ItemID, сбрасывается.

Тип данных OLE	Тип данных в Visual Basic
VT_BOOL	Boolean (логический тип)

<N>

Номер используемого флага (условия) запуска Fast Logic. Диапазон значений: 1...4.

4.4.14 Специальные информационные переменные протокола DP

Специальные информационные переменные протокола DP для ведущего устройства класса 1 предназначены для получения сведений о ведущем устройстве DP и подключенных к нему ведомых устройств DP.

Можно запросить следующую информацию:

- Режим ведущего устройства DP
- Сообщение о событиях ведущего устройства DP
- Мониторинг рабочего состояния CP
- Режим ведомого устройства DP
- Тип ведомого устройства DP
- прочая информация для более подробной диагностики

4.4.15 Синтаксис специальных информационных переменных DP

Синтаксис

DP:[<connectionname>]<informationparameter>

Пояснения

DP

Протокол DP для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

<informationparameter>

Информационный параметр. Возможны следующие значения:

- Masterstate
- EvAutoclear
- EvWatchdog
- EvClass2Master
- WatchdogTimeout
- SlavenSlvState
- SlavenSlvDiag
- SlavenMiscSlvType
- Прочие сведения о ведомом устройстве

Masterstate

Текущий режим ведущего устройства DP.

Текущий режим может быть как записан, так и прочитан. Установка режима путем записи одного из указанных ниже значений ограничена пределами DP-приложения.

Вводимые и возвращаемые значения:

OFFLINE

Обмен данными между ведущим и ведомым устройствами не осуществляется.

STOP

Обмен данными между ведущим и ведомым устройствами не осуществляется, за исключением диагностических данных.

CLEAR

Этап назначения параметров и конфигурирования.

AUTOCLEAR

Этап автоматического обнуления, ведущее устройство DP больше не может обращаться ни к одному из ведомых устройств.

OPERATE

Рабочий (производительный) этап

Тип данных OLE	Тип в Visual Basic
VT_BSTR	String (строка)

EvAutoclear

Сигнализирует ошибки во время связи с ведомыми устройствами DP. Система DP автоматически прекращает работу и переходит в режим CLEAR.

Возвращаемые значения:

True (верно)

При обмене данными с ведомыми устройствами DP произошла ошибка, и система завершила работу, перейдя в режим CLEAR.

False (ложь)

Ошибок не произошло.

Тип данных OLE	Тип в Visual Basic
VT_BOOL	Boolean (логический тип)

Примечание

EvAutoclear можно использовать лишь в том случае, если в конфигурации было выбрано AUTOCLEAR.

Примечание

Если возвращается этот информационный параметр, реагировать на него необязательно.

EvWatchdog

Сигнализирует превышение контрольного времени для задания в модуле. OPC-сервер не произвел вызов DP-функции в течение контрольного времени. Возможно, OPC-клиент или OPC-сервер больше не доступны. Контрольное время задается в конфигурации.

Возвращаемые значения:

- True (верно)
Превышение контрольного времени в модуле
- False (ложь)
Превышения контрольного времени в модуле не произошло

Тип данных OLE	Тип в Visual Basic
VT_BOOL	Boolean (логический тип)

EvClass2Master

Сигнализирует обращение со стороны ведущего устройства DP класса 2.

Возвращаемые значения:

True (верно)

Ведущее устройство DP класса 2 участвует в обмене данными и обращается к внутренним диагностическим спискам ведущего устройства DP класса 1

False (ложь)

Ведущее устройство DP класса 2 не участвует в обмене данными

Тип данных OLE	Тип в Visual Basic
VT_BOOL	Boolean (логический тип)

Примечание

Реагировать на это событие в программе пользователя не требуется.

WatchdogTimeout

Сторожевой таймер коммуникационного процессора (CP). Записав значение, можно настроить сторожевой таймер. По умолчанию используется значение, заданное в конфигурации.

Возвращаемые значения:

- 0
Контроль отключен
- 400 - 102000
Любое значение в миллисекундах. Значение округляется до значения, кратного 400.
- 6000
Значение по умолчанию

Тип данных в OLE	Тип в Visual Basic
VT_UI4	Double (двойное слово)

SlavenSlvState

Текущий статус ведомых устройств DP с адресом *n*.

Возвращаемые значения:

- OFFLINE
Обмен данными между ведущим и ведомым устройствами не происходит.
- NOT_ACTIVE
Ведомое устройство DP не активно.
- READY
Ведомое устройство DP находится в режиме передачи данных.
- READY_DIAG
Ведомое устройство DP находится в режиме передачи данных; также имеются диагностические данные.
- NOT_READY
Ведомое устройство DP не находится в режиме передачи данных.
- NOT_READY_DIAG
Ведомое устройство DP не находится в режиме передачи данных; также имеются диагностические данные.

Тип данных OLE	Тип в Visual Basic
VT_BSTR	String (строка)

SlavenSlvDiag

Последние диагностические данные ведомого устройства DP с адресом *n*.

Структура диагностических данных зависит от ведомого устройства и должна быть описана в документации на ведомое устройство.

Тип данных OLE	Тип в Visual Basic
VT_ARRAY VT_UI1	Byte() (байт)

SlavenMiscSlvType

Тип ведомого устройства DP с адресом *n*.

Возвращаемые значения:

- NO_SLV
Ведомое устройство DP отсутствует
- NORM
Стандартное ведомое устройство DP
- ET200_U
Нестандартное ведомое устройство: ET 200 U

- ET200K_B
Нестандартное ведомое устройство: ET 200 K/B
- ET200_SPM
Нестандартное ведомое устройство: станция SPM общего назначения
- UNDEFINED
Неизвестное ведомое устройство DP

Тип данных OLE	Тип в Visual Basic
VT_BSTR	String (строка)

Прочие сведения о ведомом устройстве

Перечисленные ниже информационные параметры, зависящие от ведомого устройства, не предназначены для рабочего (производительного) режима связи. В случае возникновения проблем они могут оказаться полезными для анализа перечисленных выше переменных.

Все переменные в общем случае присваиваются непосредственно DP-сервисам нижнего уровня. Эти переменные читаются так же, как и прочие информационные переменные. Результатом вызова является дамп байтов конкретного DP-сервиса нижнего уровня.

SlavenMiscReadSlvParCfgData

Конфигурационные данные ведомого устройства с адресом *n*, описывающие пространство данных ввода/вывода и согласованность данных.

SlavenMiscReadSlvParUserData

Прикладные данные ведомого устройства *n* (часть файла параметров).

SlavenMiscReadSlvParPrmData

Данные о назначении параметров для ведомого устройства *n*.

SlavenMiscReadSlvParType

Флаг SI и тип ведомого устройства *n* (флаги, связанные с ведомым устройством, тип ведомого устройства и другие резервные данные).

SlavenMiscSlvDiag

Диагностические данные ведомого устройства *n*

4.4.16 PROFIDrive

Введение

В этом разделе описаны переменные сервера шины SIMATIC NET PROFIDrive. Этот сервер служит для организации связи между устройствами с использованием протокола DP.

4.4.17 Идентификатор протокола

В соответствии с требованиями спецификации сервера профиля, все элементы начинаются с префикса **DP:2//**

Примечание

В отличие от остальных протоколов, в синтаксисе PROFIDrive не используются квадратные скобки.

Идентификатор элемента (item ID) для PROFIDrive имеет следующую структуру:

```
DP2://<Devicename>/<Variablename>
```

4.4.18 Имя устройства (Device Name)

Номер слота платы в ПК-станции (виртуальный слот) в редакторе Station Configuration Editor (Редактор конфигурирования станции) определяется номером платы. Номер платы конфигурируется с помощью SIMATIC NCM.

Синтаксис

```
DP2://brd<n>.seg<s>/&lifelist()
```

Пояснения

brd

Признак адресации к плате.

n

Номер сконфигурированной платы. Номер платы конфигурируется в SIMATIC NCM. Номер сконфигурированной платы предоставляется Сервером конфигурации (Configuration Server) с использованием имени CP (CP name). Диапазон значений: 1...32.

seg

Признак адресации к сегменту.

s

Номер сегмента PROFIBUS платы. В настоящее время всегда 0.

&lifelist()

Возвращает список всех подключенных узлов шины.

Тип данных: VT_ARRAY

Описание: 127 элементов массива (0-126)

Права доступа: только чтение

Тип данных OLE: VT_UI1

Каждый элемент массива соответствует адресу станции PROFIBUS, а индекс элемента массива соответствует адресу PROFIBUS.

Значения элементов массива имеют следующий смысл:

Значение	Пояснение
0x00	STATION_PASSIVE (Пассивная станция)
0x10	STATION_NON_EXISTENT (Станция не существует)
0x20	STATION_ACTIVE_READY (Станция готова) (готовность к включению в логическое кольцо)
0x30	STATION_ACTIVE (Станция активна) (уже в логическом кольце)

Этот элемент является информационным, он предназначен для диагностики и не используется сервером профиля.

4.4.19 Обращение к узлу шины (к ведомому устройству)

Введение

В настоящем разделе описаны различные варианты синтаксиса, которые можно использовать для доступа к данным устройства PROFIDrive:

Синтаксис

```
DP2: //brd<n>.seg<s>.dev<address>/
```

Пояснения

dev

Обозначение узла шины (устройства).

address

Указывает адрес станции PROFIBUS как узла шины.

Чтобы можно было создать и использовать элемент, узел шины не обязательно должен быть включен в шину PROFIBUS (возможно, произойдет ошибка доступа).

Диапазон: 1 ... 32

Синтаксис

```
DP2: //brd<n>.seg<s>.dev<address>/DS_Vendor-Name
```

Пояснения

DS_Vendor-Name

Элемент, представляющий узел шины.

Сервис, предназначенный для распознавания узла шины сервером профиля.

Тип данных: VT_BSTR

Права доступа: только чтение

Синтаксис

```
DP2: //brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/Timeout
```

Пояснения

DS_MSAC2_Parameter/Timeout

Элементы узла шины (управление доступом).

Параметр *Timeout* служит для мониторинга времени (в мс) обращения к элементам MSAC2 для чтения или записи.

Тип данных: VT_I4

Права доступа: чтение - запись

Синтаксис

```
DP2://brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/
ConnectionStatus
```

Пояснения

DS_MSAC2_Parameter/Connection Status

Элементы узла шины (управление доступом).

Параметр *Connection status* – содержит статус соединения MSAC2.

Тип данных: VT_I2

Права доступа: только для чтения

Значение	Пояснение
1	CONNECTED Соединение установлено.
0	NO_CONNECTION (Нет соединения) Соединение еще не установлено. SIMATIC NET OPC-сервер еще не предпринимал попытки установления соединения (не было неуспешной попытки). Статус NO_CONNECTION в следующий раз наступит только тогда, когда клиент освободит все элементы, участвовавшие в производительных коммуникациях, например, элементы для MSAC2_Read/Write, MSAC2_DataTransport или приводов.
-1	SLAVE_NOT_PRESENT (Ведомое устройство отсутствует) Соединение не могло быть установлено или было прервано. Ведомые устройства не отвечают. В отсутствии заданий на чтение/запись, предназначенных для ведомого устройства, проблема (например, отсоединение кабеля) обнаружена быть не может и первоначально не оказывает влияние на этот элемент, другими словами, в нем может сохраниться статус соединения "CONNECTED".
-2	NO_RESOURCES (Отсутствуют ресурсы) Ведомое устройство не поддерживает установление новых соединений (например, отсутствуют новые точки SAP).
-3	TIMEOUT (Превышение времени) Превышение времени при обращении к ведомому устройству.

Синтаксис

```
DP2://brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/  
SendTimeout
```

Пояснения

DS_MSAC2_Parameter/SendTimeout

Элементы узла шины (Иницирующий параметр ведущего устройства). Управление этими элементами производится отдельно для каждого объекта типа "сервер". Другими словами, значения этих параметров могут быть различными для разных клиентов, и каждый клиент может установить собственное MSAC2-соединение с адресуемыми ведомыми устройствами. Если клиент создает несколько объектов типа "сервер", управление этими элементами также производится раздельно.

SendTimeout - контроль времени (в мс) установления соединения с ведомым устройством. Устанавливается с шагом в 10 мс.

Тип данных: VT_I4

Права доступа: Чтение - запись

Синтаксис

```
DP2://brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/  
MA_Features_Supported_1
```

Пояснения

DS_MSAC2_Parameter/MA_Features_Supported_1

Элементы узла шины (Иницирующий параметр ведущего устройства). Управление этими элементами производится отдельно для каждого объекта типа "сервер". Другими словами, значения этих параметров могут быть различными для разных клиентов, и каждый клиент может установить собственное MSAC2-соединение с адресуемыми ведомыми устройствами. Если клиент создает несколько объектов типа "сервер", управление этими элементами также производится раздельно.

MA_Features_Supported_1 - функциональный параметр ведущего устройства в составе *MSAC2_Initiate*.

Тип данных: VT_UI1

Права доступа: Чтение-запись

Значение по умолчанию: 1

Синтаксис

```
DP2://brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/
MA_Features_Supported_2
DP2://brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/
MA_Profile_Features_Supported_1
DP2://brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/
MA_Profile_Features_Supported_2
```

Пояснения

DS_MSAC2_Parameter/MA_Features_Supported_2

DS_MSAC2_Parameter/MA_Profile_Features_Supported_1

DS_MSAC2_Parameter/MA_Profile_Features_Supported_2

Элементы узла шины (иницирующий параметр ведущего устройства).

Управление этими элементами производится отдельно для каждого объекта типа "сервер". Другими словами, значения этих параметров могут быть различными для разных клиентов, и каждый клиент может установить собственное MSAC2-соединение с адресуемыми ведомыми устройствами. Если клиент создает несколько объектов типа "сервер", управление этими элементами также производится раздельно.

MA_Features_Supported_2, MA_Profile_Features_Supported_1,

MA_Profile_Features_Supported_2 - функциональные параметры ведущего устройства в составе *MSAC2_Initiate*.

Тип данных: VT_UI1

Права доступа: Чтение - запись

Значение по умолчанию: 0

Синтаксис

```
DP2://brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/
MA_ProfileIdentNumber
```

Пояснения

DS_MSAC2_Parameter/MA_ProfileIdentNumber

Элементы узла шины (иницирующий параметр ведущего устройства).

Управление этими элементами производится отдельно для каждого объекта типа "сервер". Другими словами, значения этих параметров могут быть различными для разных клиентов, и каждый клиент может установить собственное MSAC2-соединение с адресуемыми ведомыми устройствами. Если клиент создает несколько объектов типа "сервер", управление этими элементами также производится раздельно.

MA_ProfileIdentNumber - функциональный параметр ведущего устройства в составе *MSAC2_Initiate*.

Тип данных: VT_UI2

Права доступа: Чтение - запись

Значение по умолчанию: 0

Синтаксис

```
DP2://brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/
SL_Features_Supported_1
DP2://brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/
SL_Features_Supported_2
DP2://brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/
SL_Profile_Features_Supported_1
DP2://brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/
SL_Profile_Features_Supported_2
```

Пояснения

DS_MSAC2_Parameter/SL_Features_Supported_1
DS_MSAC2_Parameter/SL_Features_Supported_2
DS_MSAC2_Parameter/SL_Profile_Features_Supported_1
DS_MSAC2_Parameter/SL_Profile_Features_Supported_2

Элементы узла шины (инициирующий параметр ведущего устройства). Управление этими элементами производится отдельно для каждого объекта типа "сервер". Другими словами, значения этих параметров могут быть различными для разных клиентов, и каждый клиент может установить собственное MSAC2-соединение с адресуемыми ведомыми устройствами. Если клиент создает несколько объектов типа "сервер", управление этими элементами также производится раздельно. Эти элементы снабжаются кодом качества OPC_QUALITY_GOOD только, пока существует соединение MSAC2.

SL_Features_Supported_1, SL_Features_Supported_2, SL_Profile_Features_Supported_1, SL_Profile_Features_Supported_2 - функциональные параметры ведомого устройства в составе *MSAC2_Initiate*.

Тип данных: VT_UI1

Права доступа: только чтение

Синтаксис

```
DP2://brd<n>.seg<s>.dev<address>/DS_MSAC2_Parameter/
SL_ProfileIdentNumber
```

Пояснения

DS_MSAC2_Parameter/SL_ProfileIdentNumber

Элементы узла шины (инициирующий параметр ведущего устройства). Управление этими элементами производится отдельно для каждого объекта типа "сервер". Другими словами, значения этих параметров могут быть различными для разных клиентов, и каждый клиент может установить собственное MSAC2-соединение с адресуемыми ведомыми устройствами. Если клиент создает несколько объектов типа "сервер", управление этими

элементами также производится отдельно. Эти элементы снабжаются кодом качества OPC_QUALITY_GOOD только, пока существует соединение MSAC2.

SL_ProfileIdentNumber - функциональный параметр ведомого устройства в составе *MSAC2_Initiate*.

Тип данных: VT_UI2

Права доступа: только чтение

Синтаксис

```
DP2://brd<n>.seg<s>.dev<address>/SlotI<slot>S<index>{D8209}
```

Пояснения

Элементы для обращения к *MSAC2_Read/Write*.

Тип данных: VT_ARRAY

Тип данных OLE: VT_UI1

Выбор формата с помощью D8209 не обязателен.

SlotI

Идентификатор слота ведомого устройства (как правило, идентификатор субмодуля).

slot

Слот – это область расширенной памяти ведомого устройства. Номер слота и индекс (указатель) идентифицируют "запись" данных. Для обращения к параметрам PROFIDrive, как правило, используется слот 0.
Диапазон значений: 0...254

S

Обозначение слота ведомого устройства, как правило, субмодуля.

index

Индекс (указатель) данных в пределах слота ведомого устройства. Для обращения к параметрам PROFIDrive, как правило, используется индекс 47.
Диапазон значений: 0...254

Синтаксис

```
DP2://brd<n>.seg<s>.dev<address>/SlotI<slot>S<index>
{D8209}xDT
```

Пояснения

Элементы для обращения к *MSAC2_DataTransport*.

К элементам DPC2 разрешен доступ для чтения/записи. Передача новых данных и прием результатов производится в составе предусмотренного протоколом подтверждения. Обращение к этим элементам с целью чтения осуществляется локально, возвращается "запись" данных, полученная при последнем обращении с целью записи.

Элемент снабжается кодом качества "good" только в том случае, если вслед за операцией записи были получены действительные данные.

Тип данных: VT_ARRAY

Тип данных OLE: VT_UI1

Выбор формата с помощью D8209 не обязателен.

4.4.20 Синтаксис специальных информационных системных переменных

Синтаксис

```
DP2://&version()
```

Пояснения

Переменная возвращает идентификатор версии PROFIDrive OPC-сервера, например, строку

SIMATIC NET Core Server PD V 6.xxxx.yyyy.zzzz Copyright © 2002

Тип данных: VT_BSTR

Права доступа: только чтение

4.4.21 Сервисы ведущего устройства DP класса 2

Ведущее устройство DP класса 1

Ведущее устройство DP класса 1 осуществляет циклический обмен данными с ведомыми устройствами DP. В состав его основных коммуникационных функций входит:

- Конфигурирование и назначение параметров ведомым устройствам
- Циклический обмен данными с ведомыми устройствами DP
- Контроль (мониторинг) ведомых устройств DP
- Предоставление диагностической информации

Ведущее устройство DP класса 2

Ведущее устройство DP класса 2 может выполнять задачи конфигурирования и функции online-диагностики. Основными функциями ведущего устройства DP класса 2 является:

- Чтение входных и выходных данных ведомого устройства
- Чтение диагностических данных ведомого устройства
- Чтение конфигурации ведомого устройства
- Запись "записей" данных ведомого устройства
- Чтение диагностических данных ведущего устройства класса 1

С помощью сервисов, предназначенных для циклического доступа к данным, можно выполнять чтение и мониторинг входов и выходов ведомых устройств и их диагностических данных. Ведомые устройства DP должны поддерживать дополнительные функции DP-V1. Также можно обращаться к данным ведущего устройства класса 1.

Обращение к переменным производится с помощью:

- адреса PROFIBUS ведомого устройства. Ведомые устройства не могут инициировать связь по своей инициативе. Они не получают маркер и откликаются только на вызовы со стороны ведущего устройства. Таким образом, ведомые устройства являются пассивными узлами. Номер ведомого устройства является его адресом в сети PROFIBUS.
- адреса PROFIBUS ведущего устройства (обращение к данным ведущего устройства класса 1 со стороны ведущего устройства класса 2).

4.4.22 Синтаксис сервисов ведущего устройства DP класса 2

Идентификатор протокола

Названия всех элементов начинаются с префикса: **DP2:**

4.4.23 Имя коммуникационного процессора (Name of the CP)

Имя сконфигурированного коммуникационного процессора (CP) указывает, какой именно коммуникационный процессор используется.

Синтаксис

```
DP2:[ <configuredCPname> ]
```

Пояснения

configuredCPname

Имя сконфигурированного коммуникационного процессора конфигурируется с помощью SIMATIC NCM.

Пример

```
DP:[CP5613]
```

4.4.24 Обращение к узлу шины (к ведущему устройству DP класса 1)

Введение

В данном разделе описаны различные виды синтаксиса, с помощью которых можно обращаться к данным ведущего устройства DP класса 1:

DP2-элементы узла шины

master

Признак обращения к ведущему устройству с использованием протокола DP.

masteraddress

Адрес станции как узла шины PROFIBUS.

Допустимый диапазон: 0 ...126

Синтаксис

```
DP2:[ <configuredCPname> ]master<masteraddress>MstDiag
```

Пояснения

DP2-элементы узла шины для получения диагностических данных ведущего устройства

MstDiag

Последние системные диагностические данные ведущего устройства DP

Тип данных: VT_ARRAY

Права доступа: только чтение

Тип данных OLE: VT_BOOL

Описание: 126 элементов

Каждый элемент массива информирует о том, отправило ли назначенное ему ведомое устройство диагностические данные; индекс элемента массива соответствует адресу PROFIBUS.

Значение	Пояснение
FALSE	Ведомое устройство не отправило диагностические данные.
TRUE	Ведомое устройство отправило диагностические данные.

Синтаксис

```
DP2:[<configuredCPname>]master<masteraddress>MstState
```

Пояснения

DP2-элементы узла шины для получения диагностических данных ведущего устройства

MstState

Состояние ведущего устройства DP. Содержит информацию о текущем состоянии и некоторые сведения о версии.

Тип данных: VT_ARRAY

Права доступа: Только чтение

Тип данных OLE: VT_UI1

Описание: 16 элементов

Байт	Описание
1	Рабочее состояние 0x40 – STOP 0x80 – CLEAR 0xC0 – OPERATE
2 и 3	Идентификационный номер
4	Версия оборудования (DDLМ/интерфейс пользователя)
5	Версия программы (DDLМ/интерфейс пользователя)
6	Версия оборудования
7	Версия программы
8 ... 16	Резерв

```
DP2:[<configuredCPname>]master<masteraddress>  
DataTransferList
```

Синтаксис

```
DP2:[<configuredCPname>]master<masteraddress>
DataTransferList
```

Пояснения

DP2-элементы узла шины для получения диагностических данных ведущего устройства

DataTransferList

Список передаваемых данных ведущего устройства DP.

Тип данных: VT_ARRAY

Права доступа: только чтение

Тип данных OLE: VT_BOOL

Описание: 126 элементов

Каждый элемент массива сообщает о том, находится ли назначенное ему ведомое устройство в производительном (рабочем) режиме и передало ли оно данные; индекс элемента массива соответствует адресу PROFIBUS.

Значение	Описание
FALSE	Ведомое устройство не находится в режиме передачи.
TRUE	Ведомое устройство находится в режиме передачи данных.

Синтаксис

```
DP2:[<configuredCPname>]master<masteraddress>SlvDiag
```

Пояснения

DP2-элементы узла шины для получения диагностических данных ведущего устройства

address

Адрес станции как узла шины PROFIBUS. Элемент можно создать и использовать даже, если узел шины не подключен к PROFIBUS (возможно, произойдет ошибка доступа).

Допустимый диапазон: 0...126

OPC-сервер SIMATIC NET может определить, поддерживает ли адресуемый узел шины DP-функции, лишь после того, как будет успешно выполнен сервис ведомого устройства. Для ответственных узлов шины, реакция которых на такие сервисы может быть некорректной, пользователь должен создавать соответствующие элементы самостоятельно.

SlvDiag

Последние диагностические данные ведомого устройства DP, хранящиеся на ведущем устройстве DP.

Тип данных: VT_ARRAY

Права доступа: только чтение

Тип данных в OLE: VT_UI1

4.4.25 Обращение к узлу шины (к ведомому устройству)

Введение

В следующем разделе поясняются различные виды синтаксиса, с помощью которых можно обращаться к данным ведущего устройства:

DP2-элементы узла шины

slave

Признак обращения к ведомому устройству с помощью протокола DP.

address

Адрес станции как узла шины PROFIBUS. Чтобы можно было создавать и использовать элементы, узел шины не обязательно должен быть подключен к PROFIBUS (возможно, произойдет ошибка доступа).

Допустимый диапазон: 0...126

OPC-сервер SIMATIC NET может определить, поддерживает ли адресуемый узел шины DP-функции, лишь после того, как будет успешно выполнен сервис ведомого устройства. Для ответственных узлов шины, реакция которых на такие сервисы может быть некорректной, пользователь должен создавать соответствующие элементы самостоятельно.

Синтаксис

```
DP2: [<configuredCPname>]slave<address>S<slot>data
<index>, <length>{, subarea}
```

Где subarea = <format><offset{.bit}>{.quantity}

Пояснения

DPC2-элементы узла шины для чтения/записи в режиме DPC2

S

Обозначение слота ведомого устройства, как правило, submodule.

slot

Слот – это область расширенной памяти ведомого устройства. С помощью номера слота и индекса идентифицируется "запись" данных.

Допустимый диапазон: 0...254

data

Обозначение "записи" данных.

index

Индекс (указатель) данных в пределах слота ведомого устройства.

Допустимый диапазон: 0...254

length

Длина "записи".

Допустимый диапазон: 1...240

subarea

Обозначение подобласти.

format

Элемент "format" определяет формат, в котором будут доставляться данные. Выбранный формат определяет также используемый тип данных.

Идентификатор формата	Описание	Тип данных OLE	Тип данных в Visual Basic
X	Бит	VT_BOOL	Boolean
BYTE или B	Байт (8 битов без знака)	VT_UI1	Byte
CHAR	Символ (8 битов со знаком)	VT_I1	Integer
WORD или W	Слово (16 битов без знака)	VT_UI2	Long
INT	Целое (16 битов со знаком)	VT_I2	Integer
DWORD или D	Двойное слово (32 бита без знака)	VT_UI4	Double
DINT	Двойное целое (32 бита со знаком)	VT_I4	Long
REAL	Число с плавающей запятой	VT_R4	Single

offset{.bit}

Величина смещения в байтах, определяющая положение адресуемого элемента. К биту можно обращаться, используя только тип X .

Пример: X2.3 соответствует третьему биту второго байта.

quantity

Количество элементов. Тип данных, описывающий массив из элементов (тип данных VT_ARR) указанного формата. Если эта часть имени пропущена или выбрано количество 1, используется значение 1 и переменная не является массивом.

В случае чтения подобласти "записи" данных партнерское устройство сначала читает всю "запись" целиком, после чего анализирует соответствующую подобласть.

Когда производится запись в подобласть, партнерскому устройству передается вся "запись" данных целиком. Если несколько подобластей "записи" данных записываются в рамках одного задания на запись, "запись" данных записывается в партнерское устройство только после того, как будут обновлены все подобласти "записи" данных.

В случае использования OPC-сервера, удобно объединить все элементы, используемые для частичного доступа к конкретной "записи" данных, в одну группу и записывать всю группу целиком. Следует избегать перекрытия или промежутков между подобластями, поскольку в этом случае нельзя быть уверенным, какие именно значения будут записаны в требуемые области.

Синтаксис

```
DP2:[<configuredCPname>]slave<address>DTS<slot>data
<index>,<length>{,<subarea>}
```

Где subarea = <format><offset{.bit}>{,<quantity>}

Пояснения

DPC2-элементы узла шины для транспорта данных DPC2

К элементам транспорта данных DPC2 разрешен доступ для чтения/записи. Если подобласть не указана, используется следующий тип данных:

Тип данных: VT_ARRAY

Тип данных OLE: VT_UI1

Этот элемент реализует обмен данными в рамках протокола. Обращение к этим элементам с целью чтения производится локально, в ответ возвращается "запись" данных, которая была получена при последнем обращении с целью записи.

DTS

Обозначение транспорта данных.

slot

Слот – область расширенной памяти ведомого устройства. Номер слота и индекс идентифицируют "запись" данных.

Диапазон: 0... 254

offset{.bit}

Значение смещения в байтах, определяющее позицию адресуемого элемента. Для бита можно указать только тип X .

Пример: X2.3 соответствует третьему биту во втором байте.

quantity

Количество элементов. Тип данных, описывающий массив из элементов (тип данных VT_ARR) указанного формата. Если эта часть названия пропущена или выбрано количество 1, используется значение 1 и переменная не является массивом.

В случае чтения подобласти "записи" данных анализируется подобласть последней "записи" данных, принятой от партнерского устройства.

Когда производится запись в подобласть, партнерскому устройству передается вся "запись" данных целиком. Если несколько подобластей "записи" данных записываются в рамках одно задания на запись, "запись" данных записывается в партнерское устройство только после того, как будут обновлены все подобласти "записей" данных.

В случае использования OPC-сервера, удобно объединить все элементы, используемые для частичного доступа к конкретной "записи" данных, в одну группу и записывать всю группу целиком. Следует избегать перекрытия или промежутков между областями, поскольку в этом случае нельзя быть уверенным, какие именно значения будут записаны в требуемые области.

Синтаксис

```
DP2:[<configuredCPname>]slave<address>SlvDiag
```

Пояснения

DPMCL2-элементы узла шины для диагностики ведомых устройств

SlvDiag

Последние диагностические данные ведомого устройства DP

Тип данных: VT_ARRAY

Права доступа: Только чтение

Тип данных OLE: VT_UI1

Синтаксис

```
DP2: [ <configuredCPname> ]slave<address>SlvCFGData
```

Пояснения

DPMCL2-элементы узла шины для диагностики ведомых устройств

SlvCFGData

Конфигурационные данные ведомого устройства, хранящиеся на ведущем устройстве DP.

Тип данных: VT_ARRAY

Права доступа: только чтение

Тип данных OLE: VT_UI1

Синтаксис

```
DP2: [ <configuredCPname> ]slave<address>SetSlaveAddress
```

Пояснения

DPMCL2-элементы узла шины для диагностики ведомых устройств

SetSlaveAddress

Устанавливает новый адрес PROFIBUS для ведомого устройства.

Тип данных: VT_ARRAY

Права доступа: только запись

Тип данных OLE: VT_VARIANT

При осуществлении записи в элемент выполняется соответствующий сервис DPMCL2. Записываемое значение также содержит массив с параметрами, необходимыми для выполнения сервиса.

Элемент массива	Тип данных	Значение
1	VT_UI1	Новый адрес, назначаемый ведомому устройству.
2	VT_BOOL	Флаг, указывающий, может ли адрес ведомого устройства DP быть изменен позже.
3	VT_I4	Тип устройства (идентификационный номер PROFIBUS) для узла.
4	VT_ARRAY VT_UI1	Специальные данные пользователя. Может быть передан пустой массив.

После того, как запись нового адреса ведомого устройства будет успешно выполнена, может потребоваться создать новый элемент для ведомого устройства. Элементы со старыми адресами ведомого устройства (например, элементы DPMCL2_I/O) будут по-прежнему обращаться к ведомому устройству со старым адресом, поэтому операция чтения для них не сможет быть выполнена.

Синтаксис

```
DP2:[<configuredCPname>]slave<address>_E{<format>
<offset{.bit}>>{,quantity}}
```

или (английский язык)

```
DP2:[<configuredCPname>]slave<address>_I{<format>
<offset{.bit}>>{,quantity}}
```

Пояснения

DPMCL2-элементы узла шины для обращения к входам/выходам ведомого устройства

<address>_E

<address>_I

OPC-клиенту OPC-сервера SIMATIC NET доступно только чтением входов и выходов.

Субмодули и длины данных (байт или слово), назначенные в конфигурации, а также их согласованность ведущим устройством DP класса 2 игнорируются. В этом состоит его отличие от ведущего устройства DP класса 1. Хотя ведущее устройство DP класса 2 может прочесть конфигурацию ведомого устройства в режиме выполнения, ведомые устройства и соответствующее ведущее устройство DP класса 1 могут, тем не менее, изменить свою конфигурацию во время выполнения, не сообщив об этом ведущему устройству DP класса 2. В результате конфигурационные данные окажутся несогласованными, хотя и реализуется опрос конфигурационных данных, занимающий приличное время (не говоря уж о существенной нагрузке на сеть).

Обращение к области входов/выходов производится путем указания номера ведомого устройства <address> и области входов/выходов, где "_I" является признаком области ввода. Если дополнительные сведения о формате не указываются, данные возвращаются в виде массива байтов, а длина определяется количеством входов.

Тип данных: VT_ARRAY

Права доступа: только чтение

Тип данных OLE: VT_UI1

format

Элемент "format" определяет формат, в котором будут доставляться данные. Если формат не указан, используется байтовый формат. Выбранный формат определяет используемый тип данных.

Идентификатор формата	Описание	Тип данных OLE	Тип данных в Visual Basic
X	Бит	VT_BOOL	Boolean
BYTE или B	Байт (8 битов без знака)	VT_UI1	Byte
CHAR	Символ (8 битов со знаком)	VT_I1	Integer
WORD или W	Слово (16 битов без знака)	VT_UI2	Long
INT	Целое (16 битов со знаком)	VT_I2	Integer
DWORD или D	Двойное слово (32 бита без знака)	VT_UI4	Double
DINT	Двойное целое (32 бита со знаком)	VT_I4	Long
REAL	Число с плавающей запятой	VT_R4	Single

offset{.bit}

Значение смещения в байтах, определяющее позицию адресуемого элемента. Для бита можно указать только тип X.

Пример: X2.3 соответствует третьему биту во втором байте.

quantity

Количество элементов. Тип данных, описывающий массив из элементов (тип данных VT_ARR) указанного формата. Если выбран формат X, количество указать нельзя.

Если эта часть названия пропущена или выбрано количество 1, используется значение 1 и переменная не является массивом.

Синтаксис

```
DP2:[<configuredCPname>]slave<address>_A{<format>
<offset{.bit}>{,quantity}}
```

или (английский язык)

```
DP2:[<configuredCPname>]slave<address>_Q{<format>
<offset{.bit}>{,quantity}}
```

Пояснения

DPMCL2-элементы узла шины для обращения к входам/выходам ведомого устройства

Обращение к области ввода/вывода осуществляется путем указания номера ведомого устройства <address> и области ввода/вывода, где "_Q" обозначает область вывода. Если дополнительные сведения о формате не указываются, данные возвращаются в виде массива байтов, длина определяется количеством входов.

Тип данных: VT_ARRAY

Права доступа: только чтение

Тип данных OLE: VT_UI1

4.4.26 Элементы для коммуникационного процессора (CP)

Синтаксис

```
DP2:[<configuredCPname>]&lifelist()
```

Пояснение

&lifelist()

Элементы общего назначения.

Возвращается список всех подсоединенных узлов шины.

Тип данных: VT_ARRAY

Описание: массив из 127 элементов (0-126)

Права доступа: только чтение

Тип данных OLE: VT_UI1

Каждый элемент массива соответствует определенному адресу станции PROFIBUS. Значения элементов массива имеют следующий смысл:

Элемент	Пояснение
0x00	STATION_PASSIVE (Пассивная станция)
0x10	STATION_NON_EXISTENT (Станция не существует)
0x20	STATION_ACTIVE_READY (Станция активна и готова) (готовность к включению в логическое кольцо)
0x30	STATION_ACTIVE (Станция активна) (уже в логическом кольце)

4.4.27 Примеры специальных информационных DP-переменных

В этом разделе приведено несколько примеров возвращаемых значений специальных информационных DP-переменных.

Режим ведущего устройства DP

DP:[CP 5613]MasterState

MasterState= Статус ведущего устройства

Может быть возвращено, например, следующее значение:

OPERATE	Ведущее устройство в рабочем (производительном) режиме.
---------	---

Текущее состояние ведомого устройства

DP:[CP 5613]Slave3SlvState

Slave3SlvState

Может быть возвращено, например, следующее значение:

READY	Ведомое устройство DP с адресом 3 находится в режиме передачи данных.
-------	---

Тип ведомого устройства

DP:[CP 5613]Slave3MiscSlvType

Slave3MiscSlvType

Может быть возвращено, например, следующее значение:

ET200_U	Ведомым устройством 3 является ET 200 U.
---------	--

4.4.28 Сервисы обслуживания переменных для обращения к локальным данным ведомого устройства

OPC-сервер семейства SIMATIC NET может выполнять в сети PROFIBUS DP не только функции ведущего устройства, но также может быть ведомым устройством DP. OPC-сервер управляет областями памяти, назначенными для входов и выходов ведомого устройства DP, и назначает их OPC-переменным. OPC-клиент может прочитать выходы, установленные ведущим устройствам, и установить значения входов, которые будут прочитаны ведущим устройством в следующем цикле.

DP-ведомые имеют модульную структуру. Ведомое устройство DP может состоять из нескольких субмодулей с различными областями ввода/вывода. Назначение субмодулей производится на этапе конфигурирования.

Имя переменной идентифицирует область ввода/вывода в пределах субмодуля ведомого устройства. Обращение к входам и выходам ведомого устройства производится путем указания номера субмодуля и области ввода или вывода.

Информация о состояниях ведомых устройств и ведущего устройства DP также может быть запрошена с использованием имен переменных.

Примечание

Модуль CP 5614 может одновременно работать в качестве ведущего и ведомого устройства DP только в том случае, если на этапе конфигурирования был выбран режим DP Base.

4.4.29 Синтаксис переменных процесса для ведомого устройства DP

Синтаксис

Входы

```
DP: [<connectionname>]slave {M<number>}_I {<format><offset>
{.<bit>}{,<quantity>}}
```

Выходы

```
DP: [<connectionname>]slave {M<number>}_Q<format><offset>
{.<bit>}{,<quantity>}}
```

Пояснения

DP

Протокол для обращения к переменным процесса.

<connectionname>

Название коммуникационного модуля, указанное в конфигурации.

slave

Признак обращения к ведомому устройству по протоколу DP.

M

Обозначение номера submodule.

<number>

Номер submodule, которому принадлежит пространство входов или выходов.

_I

Обозначение входа.

_Q

Обозначение выхода.

<format>

Формат, в котором будут поставляться данные.

Выбранный формат также определяет используемый тип данных.

Все указанные типы данных, предусмотренные для OLE, могут быть прочитаны через OPC-интерфейс автоматизации. Тем не менее, некоторые средства разработки (например, Visual Basic) поддерживают только ограниченное количество типов данных. В следующей таблице перечислены соответствующие типы, предусмотренные в Visual Basic, в формате которых может представлено значение переменной.

Идентификатор формата	Описание	Тип данных OLE	Тип данных в Visual Basic
X	Бит	VT_BOOL	Boolean
BYTE или B	Байт (8 битов без знака)	VT_UI1	Byte
CHAR	Символ (8 битов со знаком)	VT_I1	Integer
WORD или W	Слово (16 битов без знака)	VT_UI2	Long
INT	Целое (16 битов со знаком)	VT_I2	Integer
DWORD или D	Двойное слово (32 бита без знака)	VT_UI4	Double
DINT	Двойное целое (32 бита со знаком)	VT_I4	Long
REAL	Число с плавающей запятой	VT_R4	Single

<offset>

Адрес байта в пространстве адресов ведомого устройства, по которому расположен адресуемый элемент.

<bit>

Номер бита в адресуемом байте. Допустимый диапазон значений: 0...7.

Бит можно указать только в случае формата **X**.

<quantity>

Количество элементов.

Этому типу данных (VT_ARRAY) соответствует массив элементов указанного формата.

Если параметр *quantity* пропущен, предполагается, что количество элементов = 1 и переменная не является массивом.

Нельзя использовать параметр *quantity*, если выбран формат **X**.

4.4.30 Примеры переменных процесса для ведомого устройства DP

Ниже представлены примеры, иллюстрирующие синтаксис имен для переменных DPS.

Входы

DP:[CP 5611]SlaveM003_IB0

SlaveM003_IB0

Входной байт 0 (смещение 0) submodule 3 ведомого устройства DP.

DP:[CP 5611]SlaveM003_IB1,3

SlaveM003_IB1,3

Массив из 3 байтов, начинающийся с входного байта 1 (смещение 1), submodule 3 ведомого устройства DP.

DP:[CP 5614]SlaveM003_IX0.0

SlaveM003_IX0.0

Входной бит 0 в байте 0, в submodule 3 ведомого устройства DP.

DP:[CP 5614]SlaveM003_IB3,8

SlaveM003_IB3,8

Массив из 8 входных байтов, начинающийся с байта 3 (смещение 3), в submodule 3 ведомого устройства DP.

Выходы

DP:[CP 5611]SlaveM003_QW3

SlaveM004_QW3

Выходное слово по адресу 3, в submodule 4 ведомого устройства DP.

DP:[CP 5611]SlaveM003_QDWORD2

SlaveM003_QDWORD2

Выходное двойное слово по адресу 2, в submodule 3 ведомого устройства DP.

DP:[CP 5614]SlaveM003_QX3.7

Slave_QX3.7

Выходной бит 7 в байте 3 ведомого устройства DP.

DP:[CP 5614]SlaveM001_QW0,4

SlaveM001_QW0,4

Массив из 4 выходных слов, в submodule 1 ведомого устройства DP.

4.4.31 Специальные информационные переменные ведомого устройства DP

Для диагностики ведомых устройств DP предусмотрено несколько предопределенных информационных переменных.

4.4.32 Синтаксис специальных информационных переменных ведомого устройства DP

Синтаксис

Имеется две формы синтаксиса:

```
DP:[<connectionname>]&<diagnosticitem>
```

```
DP:[<connectionname>]<parameteritem>
```

Пояснения

DP

Протокол для доступа к переменным процесса.

<connectionname>

Имя, указанное в конфигурации коммуникационного модуля.

<diagnosticitem>

Предопределенный элемент.

Имеются следующие варианты:

devicestate

Статус модуля, выполняющего функции ведомого устройства DP.

Модуль может находиться в одном из следующих состояний:

- ONLINE (связь установлена)
- OFFLINE (связь не установлена)

parameteritem

Предопределенные элементы, содержащие сведения о параметрах.

Имеются следующие варианты:

SlaveMiscReadSlvParCfgData

Конфигурационные данные ведомого устройства.

SlaveSlvState

Статус ведомого устройства.

Ведомое устройство может находиться в одном из следующих состояний:

- DATA_EXCHANGE (обмен данными)
- NO_DATA_EXCHANGE (обмена данными нет)

4.5 S7-соединения

Протокол S7 для PROFIBUS и Industrial Ethernet служит для организации связи между компонентами системы автоматизации на базе программируемых контроллеров SIMATIC S7, а также для связи между компонентами системы SIMATIC S7 и программаторами (PG) или персональными компьютерами (PC).

Особенности S7-коммуникаций с применением OPC

OPC-сервер для SIMATIC NET обладает следующими характеристиками:

- Сервисы для работы с переменными.
Обращение к S7-переменным и их контроль.
- Сервисы для работы с буферами.
Программно-управляемая передача больших блоков данных.
- Функции сервера.
В случае применения буфер-ориентированных сервисов ПК также можно использовать в качестве сервера.
- Сервисы для управления блоками.
Передача загружаемых областей данных из/в S7.
- Функции установки паролей для S7.
Установка паролей на доступ к защищенным блокам.
- События
Обработка S7-сообщений (S7-тревог) в соответствии со спецификацией OPC Alarms & Events.

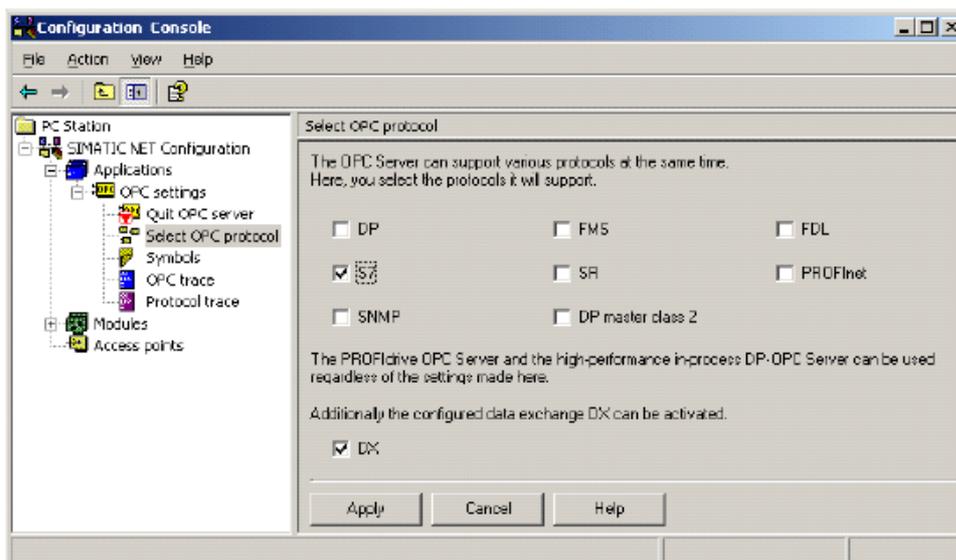
4.5.1 Мощный OPC-сервер SIMATIC NET для протокола S7

Введение

В этом разделе будет описан вариант конфигурации для протокола S7, который удовлетворяет требованиям к повышенной производительности. Чтобы можно было использовать этот вариант, в область внепроцессного OPC-сервера загружаются все базовые библиотеки протокола S7, а также COM-сервер в качестве внутрипроцессного сервера. Управление протоколом происходит в рамках процесса OPC-сервера, что позволяет избежать дополнительных временных затрат на переключение между процессами и применение мультипротокольного режима. Переключение между процессами OPC-клиента и OPC-сервера по-прежнему необходимо.

Конфигурация

Этот вариант повышенной производительности вступает в силу автоматически в результате выбора протокола S7 в программе Configuration Console ("Консоль конфигурирования").



Также можно выбрать сконфигурированный обмен данными через OPC (DX) и использование символов.

Преимущества/Недостатки

Применение высокопроизводительного S7 OPC-севера, однако, обладает недостатком, который состоит в том, что используется только протокол S7 (однопротокольный режим). С другой стороны, это дает пользователю следующие преимущества:

- Более высокая производительность по сравнению с мультипротокольным режимом.
- Простое конфигурирование.
- Возможность обращения с использованием идентификатора (ProgID) *OPC.SimaticNET*.
- Несколько клиентов могут использовать сервер одновременно.
- Стабильность работы OPC-сервера не зависит от клиентов.

4.5.2 Идентификатор протокола

Для протокола S7 используется идентификатор протокола **S7**.

4.5.3 Имена соединений

Имя соединения – это имя, сконфигурированное в STEP 7 для идентификации этого соединения. В STEP 7 это имя называют локальным идентификатором (local ID). Локальный идентификатор уникален в пределах OPC-сервера.

OPC-сервер поддерживает следующие типы соединений:

- S7-соединение
- Отказоустойчивое S7-соединение

Примечание

Чтобы обеспечить совместимость с версией 2.2 и более ранними версиями, необходимо иметь в виду следующее:

В более старых версиях информация о соединении включала сведения о S7-соединении, устройствах VFD и о коммуникационном процессоре. Эти элементы по-прежнему могут включаться в синтаксис идентификатора элемента (ItemID) для переменных, но больше не отображаются в окне обзора OPC (OPC Browsing). В STEP 7 по-прежнему можно сконфигурировать одновременное использование нескольких VFD OPC-сервером, но только из соображений совместимости, если это предусматривалось предшествующей конфигурацией.

Примеры имен соединений

Типичные имена соединений:

S7-connection-1

S7-OPC-connection

my_connection

4.5.4 Сервисы для работы с переменными

Сервисы для работы с переменными обеспечивают доступ к S7-переменным и их мониторинг в программируемом контроллере. Обращение к S7-переменным осуществляется с использованием символов. Тип обращения ориентирован на способ адресации, используемый в инженеринговых инструментах S7.

Объекты программируемого контроллера

OPC-сервер поддерживает следующие объекты:

- Блоки данных
- Экземплярные блоки данных
- Входы
- Выходы
- Периферийные входы
- Периферийные выходы
- Биты памяти
- Таймеры
- Счетчики

В некоторых программируемых контроллерах S7 поддерживаются не все типы объектов.

4.5.5 Синтаксис переменных процесса для сервисов обслуживания переменных S7

Синтаксис

Предусмотрено три формы синтаксиса:

S7:[<connectionname>]**DB**<no>,{<type>}<address>{,<quantity>}

S7:[<connectionname>]**DI**<no>,{<type>}<address>{,<quantity>}

S7:[<connectionname>]<object>{<type>}<address>{,<quantity>}

Пояснения

S7

Протокол S7 для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

DB

Обозначение блока данных для переменной S7, содержащейся в блоке данных.

DI

Обозначение экземплярного блока данных переменной S7, содержащейся в экземплярном блоке данных.

<Nr>

Номер блока данных или экземплярного блока данных.

<object>

Указывает тип блока/области в ПЛК S7.

Можно указать одно из следующих значений:

- I** Вход
- Q** Выход
- PI** Периферийный вход
- PQ** Периферийный выход
- M** Биты памяти
- T** Таймеры
- C** Счетчики

Масштаб времени и диапазон значений для S7-переменных типа "таймер" (тип T):

OPC-переменные процесса для S7 типа "таймер" (T) представляются в десятичном коде, в миллисекундах. Масштаб времени определяется в порядке убывания множителями 10000 мс, 1000 мс, 100 мс и 10 мс. Область возможных значений является произведением этого масштаба времени на коэффициент от 1 до 999:

Возможные значения	Масштаб времени	Комментарий
10 000 – 9 990 000 мс	10 000 мс	Значения должны быть кратны 10 000 мс.
1000 мс – 999 000 мс	1000 мс	Значения должны быть кратны 1000 мс.
100 мс – 99 900 мс	100 мс	Значения должны быть кратны 100 мс.
10 мс – 9 990 мс	10 мс	Значения должны быть кратны 10 мс. Меньший масштаб времени невозможен. 0 мс допускается, но не имеет силы.

Диапазон значений S7-переменных типа "счетчик" (тип C):

OPC-переменных процесса для S7 типа "счетчик" (C) могут принимать значения от 0 до 999 в десятичном коде.

<type>

Тип данных S7

Переменная с типом данных протокола S7 приводится к соответствующему типу данных OLE внутри OPC-сервера.

Все перечисленные типы данных OLE могут быть прочитаны через интерфейс автоматизации OPC. Тем не менее, некоторыми средствами проектирования (например, Visual Basic) поддерживается только ограниченный набор типов данных. В следующей таблице перечислены соответствующие типы, поддерживаемые в Visual Basic, в формате которых может быть представлена переменная.

Для объектов типа T и C тип указать нельзя.

Идентификатор формата	Описание	Тип данных OLE	Тип в Visual Basic
X	Бит (логический тип). Должен быть указан номер бита (0-7).	VT_BOOL	Boolean
B	Байт (без знака).	VT_UI1	Byte
W	Слово (со знаком).	VT_UI2	Long
D	Двойное слово (без знака).	VT_UI4	Double
CHAR	Байт (со знаком).	VT_I1	Integer
INT	Слово (со знаком).	VT_I2	Integer
DINT	Двойное слово (со знаком).	VT_I4	Long
REAL	Число с плавающей запятой.	VT_R4	Single
STRING	Строка. Должна быть указана длина строки.	VT_BSTR	String
DT	Дата и время, 8 байтов в формате BCD.	VT_DATE	Date
TIME	Значение времени (со знаком), формат IEC, в миллисекундах	VT_I4	Long
TOD	Время суток (без знака), 0 ... 86 399 999 (мс)	VT_UI4	Double

Типы данных CHAR, INT, DINT и REAL могут применяться только внутри блока данных (**DB**) или внутри экземплярного блока данных (**DI**).

<address>

Адрес первой адресуемой переменной. Возможны следующие значения:

- Смещение в байтах
- Смещение в байтах. Бит (только для типа данных X).
- Смещение в байтах. Длина строки (только для типа данных "String", длина строки от 1 байта до 254 байтов).

<quantity>

Количество адресуемых переменных указанного типа, начиная с переменной, определяемой смещением, указанным в параметре *address*.

Для объектов **T** и **C** количество ввести нельзя.

Для типа данных **X** можно ввести только количество, кратное числу 8, поэтому адрес бита должен быть нулевым.

Пример: S7:[S7-OPC-1]DB1,X10,0,64

4.5.6 **Примеры переменных процесса для сервисов обслуживания переменных S7**

Ниже приведены примеры синтаксиса имен переменных для сервисов обслуживания переменных.

Параметр блока данных

S7:[S7-connection-1]DB5,B12

DB5,B12

Указывает на байт данных 12 в блоке данных 5.

Параметр экземплярного блока данных

S7:[S7-connection-1]DI5,W10,9

DI5,W10,9

Указывает на 9 слов данных, начиная с байта по адресу 10, внутри экземплярного блока данных 5.

Параметр объекта

S7:[S7-connection-1]IB0

IB0

Указывает на входной байт 0.

4.5.7 Буфер-ориентированные сервисы

Буфер-ориентированные сервисы позволяют организовать передачу больших блоков данных, управляемую программно. Для передачи данных используются следующие переменные:

- Переменные для передачи блоков данных
- Переменные для приема блоков данных

Независимо от размера PDU, можно передать до 65 534 байтов "полезных" данных. Разбиение данных на фрагменты выполняется самими функциями.

Примечание

Буфер-ориентированные сервисы можно использовать только для двухсторонних соединений. Конфигурация соединений должна быть загружена в программируемый контроллер S7

4.5.8 Синтаксис переменных процесса для буфер-ориентированных сервисов

Синтаксис

Предусмотрено две формы синтаксиса:

S7:[<connectionname>]**BRCV**,<RID>{,<type>}<address>
{,<quantity>}}

S7:[<connectionname>]**BSEND**<length>,<RID>{,<type>}<address>
{,<quantity>}}

Пояснения

S7

Протокол S7 для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

BRCV

В BRCV содержится последний блок данных, полученный от коммуникационного партнера.

Содержание и длина принятых данных указываются передающим партнером.

Эта переменная может быть только прочитана.

Эти переменные следует использовать для реализации контроля. Затем уведомление о приеме блока данных OPC-сервером отправляется клиентскому OPC-приложению.

По истечении времени установления соединения (время, указанное в конфигурации для конкретного соединения) или после приема блока данных возвращается явное задание на чтение этих переменных.

Тип данных OLE	Тип в Visual Basic
VT_ARRAY VT_UI1	Byte() (байт)

BSEND

В BSEND содержится блок данных, предназначенных для передачи коммуникационному партнеру.

Блок данных передается партнерскому устройству только после записи значения в переменную. При чтении возвращается содержимое последнего успешно переданного блока данных.

Тип данных OLE	Тип в Visual Basic
VT_ARRAY VT_UI1	Byte() (байт)

<length>

Длина передаваемого блока данных в байтах.

<RID>

Идентификатор параметра адресации. Он устанавливается для пары блоков (**BSEND/ BRCV**) и является уникальным внутри соединения.

Через одно соединение можно передать несколько блоков **BSEND** или принять несколько блоков **BRCV**, но каждый блок должен иметь свой собственный идентификатор. В других соединениях можно использовать такие же идентификаторы.

<type>

Тип данных S7.

Тип данных S7 преобразуется в соответствующий тип данных OLE в OPC-сервере.

Все перечисленные типы данных OLE могут быть прочитаны через OPC-интерфейс автоматизации. Тем не менее, некоторыми средствами проектирования (например, Visual Basic) поддерживается только ограниченный набор типов данных. В следующей таблице перечислены соответствующие типы, поддерживаемые в Visual Basic, в формате которых может быть представлена переменная.

Идентификатор формата	Описание	Тип данных OLE	Тип в Visual Basic
X	Бит (логический тип). Должен быть указан номер бита.	VT_BOOL	Boolean
B	Байт (без знака)	VT_UI1	Byte
W	Слово (со знаком)	VT_UI2	Long
D	Двойное слово (без знака)	VT_UI4	Double
CHAR	Байт (со знаком)	VT_I1	Integer
INT	Слово (со знаком)	VT_I2	Integer
DINT	Двойное слово (со знаком)	VT_I4	Long
REAL	Число с плавающей запятой	VT_R4	Single

<address>

Адрес первой адресуемой переменной.

Возможны следующие значения:

номер байта

номер байта.бит (только для формата X)

<quantity>

Количество адресуемых переменных указанного типа, начиная с адреса, указанного в параметр *address*.

Для типа данных **X** можно ввести только количество, кратное 8-ми. Для бита должен быть указан адрес 0.

Пример: S7:[S7-OPC-1]BRCV,1,X10.0,64

Примечание

Указывая дополнительные сведения (тип, адрес и количество), можно структурировать обращение к подобластям блоков данных.

4.5.9 Примеры переменных процесса для буфер-ориентированных сервисов

Ниже приведено несколько примеров синтаксиса имен переменных для буфер-ориентированных сервисов.

Блок данных, принимаемый целиком в один буфер

S7:[S7-OPC-1]BRCV,1

BRCV,1

Блок данных принимается в буфер приема с RID 1. В буфер размещается полностью весь массив байтов.

Обращение к отдельным частям принятого блока данных

S7:[S7-OPC-1]BRCV,1,W2,4

BRCV,1,W2,4

Для обращения к отдельным данным принятого блока данных указывается смещение 2, считывается массив из 4 слов. Всего считывается 8 байтов блока данных.

Передача двойного слова

S7:[S7-OPC-1]BSEND16,1,D2

BSEND16,1,D2

Производится обращение к двойному слову в блоке данных длиной в 16 слов, RID = 1, смещение = 2. Если команда записи применяется для переменной, записываемое значение вводится в блок по указанному адресу, после чего передается блок данных.

Передача чисел в формате с плавающей запятой

S7:[S7-OPC-1]BSEND32,1,REAL20,2

BSEND32,1,REAL20,2

Обращение производится к массиву чисел в формате с плавающей запятой, принадлежащих блоку данных длиной 32, RID = 1, смещение = 20. Если команда записи применяется для переменной, записываемое значение вводится в блок по указанному адресу, после чего передается блок данных.

4.5.10 Специальные информационные переменные протокола S7

Предусмотрен ряд специальных S7-переменных, которые позволяют получить информацию о состоянии S7-коммуникаций и установленных соединениях.

Можно получить следующую информацию:

- Атрибут виртуального полевого устройства (VFD).
- Состояние S7-соединения
- Состояние виртуального устройства
- Статус регистрации для получения сообщений

4.5.11 Синтаксис специальных информационных переменных протокола S7

Синтаксис

S7:[<connectionname>]<informationparameter>

Пояснения

S7

Протокол S7 для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

<informationparameter>

Возможны следующие значения:

&identify()

Атрибуты производителя коммуникационного устройства-партнера.

Возвращаемые значения (элементы строки):

- Производитель
- Модель
- Версия

Тип данных OLE	Тип в Visual Basic
VT_ARRAY of VT_BSTR	String() (строка)

&vfdstate()

Состояние виртуального полевого устройства

Тип данных: массив **VT_ARRAY** из элементов типа **VT_VARIANT**.

Содержит следующие элементы:

Логический статус

Поддерживаемые сервисы

Возвращаемое значение:

S7_STATE_CHANGES_ALLOWED	Разрешены все сервисы
--------------------------	-----------------------

Тип данных OLE	Тип в Visual Basic
VT_BSTR	String (строка)

Физическое состояние

Работоспособность физического устройства

Возвращаемые значения:

S7_OPERATIONAL	Физическое устройство работоспособно.
S7_NEEDS_COMMISSIONING	Физическое устройство работоспособно после выполнения локальных настроек.

Тип данных OLE Data	Тип в Visual Basic
VT_BSTR	String (строка)

Подробные сведения о статусе локального VFD

Состояние возвращается в формате строки октетов. Более подробные сведения о смысле возвращаемых значений можно найти в соответствующей документации по коммуникационному партнеру.

Тип данных OLE	Тип в Visual Basic
массив VT_ARRAY из элементов типа VT_BSTR	String() (строка)

&statepath()

Состояние коммуникационного соединения с партнерским устройством.

Возвращаемые значения:

- DOWN
Соединения не установлено
- UP
Соединение установлено
- RECOVERY
Соединение устанавливается
- ESTABLISH
(Зарезервировано для дальнейшего использования)

Тип данных OLE	Тип в Visual Basic
VT_BSTR	String (строка)

Примечание

Переменные &statepath() и &statepathval() могут переключаться между значения UP и RECOVERY.

Если в файле конфигурации для протокола SR для параметра "S5Access" выбрано значение 1 и используется параметр TIMEOUT, в этом случае при отсутствии принятых данных значения переменных &statepath() и &statepathval() изменяются между UP и RECOVERY (2 или 3).

Способ устранения: выберите для параметра "S5Access" значение 0. Описанная ситуация не произойдет.

&statepathval()

Состояние коммуникационного соединения с партнерским устройством.

Возвращаемые значения:

- 1
Соединение не установлено
- 2
Соединение установлено
- 3
Соединение устанавливается
- 4
(Зарезервировано для дальнейшего использования)

Тип данных OLE	Тип в Visual Basic
VT_UI1	Byte (байт)

4.5.12 Примеры специальных информационных переменных протокола S7 и возвращаемых значений

Ниже представлены примеры, иллюстрирующие синтаксис имен переменных для буфер-ориентированных сервисов.

Сведения об атрибутах производителя виртуального полевого устройства

S7:[S7-OPC-1]&identify()

&identify()

В ответ могут быть возвращены, например, следующие значения:

- Производитель: "SIEMENS AG"
- Модель виртуального устройства: "GES7413-1AE0-0AB0"
- Версия: "V1.0"

Статус устройства

S7:[S7-OPC-1]&vfdstate()

&vfdstate()

В ответ могут быть возвращены, например, следующие значения:

- Логический статус: "S7_STATE_CHANGES_ALLOWED"
Разрешены все сервисы.
- Физическое состояние: "S7_OPERATIONAL"
Физическое устройство работоспособно.
- Подробная информация: "02.00.00"
Подробная информация о статусе локального VFD

Состояние коммуникационного соединения в формате строки

S7:[S7-OPC-1]&statepath()

&statepath()

Могут быть возвращены, например, следующие значения:

- Состояние соединения: "RECOVERY"
Соединение в настоящий момент устанавливается.

Состояние коммуникационного соединения в виде числового значения

S7:[S7-OPC-1]&statepathval()

&statepathval()

Могут быть возвращены, например, следующие значения:

- Состояние соединения: "2"
Соединение устанавливается

4.5.13 Сервисы для управления блоками

Сервисы управления блоками служат для управления передачей и приемом данных и элементов программы между ПК и программируемым контроллером. В ПК данные и элементы программы хранятся в виде файлов. Сервисы для управления блоками можно защитить с помощью пароля.

Блоки

Данные и элементы программ хранятся в памяти программируемых контроллеров S7 в виде блоков. Эти блоки компилируются с помощью STEP 7 и передаются в память S7-устройств. В S7-устройствах могут присутствовать следующие блоки:

- Организационные блоки
- Функциональные блоки
- Блоки данных (DB / DI)

Функции для управления блоками

С помощью OPC-сервера SIMATIC NET можно выполнять следующие функции:

- Обмен блоками между ПК и программируемыми контроллерами
- Удаление блоков
- Связывание блоков
- Сжатие памяти программируемого контроллера

Примечание

С помощью OPC-сервера нельзя создавать блоки, для этого используется STEP 7.

4.5.14 Синтаксис управляющих переменных для сервисов для управления блоками

Синтаксис

S7:[<connectionname>]<serviceparameter>

Пояснения

S7

Протокол S7 для обращения к переменной процесса.

<connectionname>

Специальное имя соединения для протокола. Имя соединения указывается в конфигурации.

<serviceparameter>

Эти переменные могут быть только записаны. Запуск сервиса осуществляется путем записи значения в переменную.

Возможны следующие значения:

&blockread()

Передача блока из программируемого контроллера в ПК и запись его в виде файла.

Записываемое значение содержит параметр сервиса. Последний представляется в виде массива переменных типа *variant* (произвольный тип), состоящего из следующих элементов:

Флаги

Можно записывать следующие шестнадцатеричные значения:

0x0001	Чтение несвязанного блока. Если адресуемый файл существует, он не перезаписывается, отображается сообщение об ошибке.
0x0040	Чтение связанного блока. Если адресуемый файл существует, он не перезаписывается, отображается сообщение об ошибке.
0x1001	Чтение несвязанного блока. Существующий адресуемый файл перезаписывается.
0x1040	Чтение связанного блока. Существующий адресуемый файл перезаписывается.

Тип данных OLE	Тип в Visual Basic
VT-BSTR	String (строка)

Блок

Тип и номер блока

OB	Организационный блок
FB	Функциональный блок
FC	Функция
DB	Блок данных

Тип данных OLE	Тип в Visual Basic
VT-BSTR	String (строка)

Файл

Полное имя (путь) файла, в котором будет храниться блок.

Тип данных OLE	Тип в Visual Basic
VT-BSTR	String (строка)

&blockwrite()

Передача блока из ПК в программируемый контроллер.

Записываемое значение содержит параметр для данного сервиса. Он представляется в виде массива переменных типа *variant* (произвольный тип), состоящего из следующих элементов:

Флаги

Возможна запись следующих шестнадцатеричных значений:

0x1000	Несвязанный блок с тем же именем в системе автоматизации должен быть перезаписан
0x0000	Несвязанный блок с тем же именем в системе автоматизации не перезаписывается

Тип данных OLE	Тип в Visual Basic
VT-BSTR	String (строка)

Файл

Полное имя (путь) файла, в котором хранится блок

Тип данных OLE	Тип в Visual Basic
VT-BSTR	String (строка)

Примечание

Доменные сервисы "blockread()" и "blockwrite()" не обеспечивают доступ к исходному и адресуемому файлам блоков, расположенных на сетевом диске.

Перезапись (0x1000) блока возможно лишь в том случае, если установлена связь с соответствующей частью этого блока.

&blocklinkin()

Связывание блока, имеющего пассивный статус, с программной последовательностью в программируемом контроллере. Исполняемая часть блока копируется в рабочую память программируемого контроллера.

После этого блок становится доступен для программы. В результате связывания блока существующий активный блок перезаписывается без формирования сообщения об ошибке.

Записанное значение содержит идентификатор связываемого блока в качестве параметра для сервиса.

Блок

Тип и номер блока

Возможны следующие типы:

OB	Организационный блок
FB	Функциональный блок
FC	Функция
DB	Блок данных

Тип данных OLE	Тип в Visual Basic
VT-BSTR	String (строка)

&blockdelete()

Удаление блока в программируемом контроллере. Можно удалить как пассивные блоки, так и блоки, включенные в программную последовательность (активные блоки).

Записываемое значение содержит параметр для данного сервиса. Последний представляется в виде массива переменных типа *variant* (произвольный тип), состоящего из следующих элементов:

Флаги

Возможны следующие шестнадцатеричные значения:

0x0001	Несвязанный блок удаляется
0x0040	Связанный блок удаляется
0x0041	Удаляются связанные и несвязанные блоки

Тип данных OLE	Тип в Visual Basic
VT-BSTR	String (строка)

Блок

Тип и номер блока

Возможны следующие типы:

OB	Организационный блок
FB	Функциональный блок
FC	Функция
DB	Блок данных

Тип данных OLE	Тип в Visual Basic
VT-BSTR	String (строка)

&blockcompress()

Сжатие памяти программируемого контроллера.

Отдельные сегменты областей памяти объединяются, освобождая место для новых передаваемых блоков.

В эту переменную значение можно только записать. Сервис запускается путем записи пустой строки.

Примечание

Настройка параметров для некоторых сервисов управления блоками производится путем записи значения. В этом случае в качестве типов данных используются массивы. С точки зрения пользователя может оказаться более предпочтительным передавать значения параметров в виде значений с типом данных VT_BSTR. OPC-сервер конвертирует значения в требуемый тип данных автоматически. Массив представляется в виде строки, например, следующим образом: "{первый элемент|второй элемент|третий элемент}".

Всегда следует пользоваться примерами.

4.5.15 Примеры применения сервисов для управления блоками

Ниже представлено несколько примеров, иллюстрирующих синтаксис имен переменных для сервисов управления блоками.

Чтение блока

S7:[S7-OPC-1]&blockread() &blockread()

Для чтения блока **OB1**, хранящегося в файле **c:\temp\ob1.blk**, в элемент должно быть записано следующее значение:

```
{0x0040|OB1|c:\temp\ob1.blk}
```

Примечание

Чтобы указать такое значение, как в примере, должен использоваться тип данных string (строка). После этого значения для массивов можно указывать в конвертированном виде.

Запись блока

***S7:[S7-OPC-1]&blockwrite()
&blockwrite()***

Для загрузки блока, хранящегося в файле **c:\temp\ob1.blk**, в программируемый контроллер, в элемент должно быть записано следующее значение:

{0x1000|c:\temp\ob1.blk}

Примечание

Чтобы указать такое значение, как в примере, должен использоваться тип данных **string** (строка). После этого значения для массивов можно указывать в конвертированном виде.

Пример связывания блока

***S7:[S7-OPC-1]&blocklinkin()
&blocklinkin()***

Чтобы включить блок **DB1** в программу программируемого контроллера, в элемент должно быть записано следующее значение:

DB1

Пример удаления блока

***S7:[S7-OPC-1]&blockdelete()
&blockdelete()***

Чтобы удалить включенный в программу (связанный) блок **DB1**, в элемент должно быть записано следующее значение:

{0x0040|DB1}

Пример сжатия памяти программируемого контроллера

***S7:[S7-OPC-1]&blockcompress()
&blockcompress()***

Чтобы выполнить сжатие памяти, в элемент должна быть записана пустая строка.

4.5.16 Пароли

На этапе конфигурирования для сервисов управления блоками можно установить пароли на доступ к модулю CPU с целью записи и чтения. Пароль имеет большую силу по отношению к ключу-переключателю модуля CPU.

Уровни защиты

Для программируемых контроллеров S7 предусмотрено три уровня защиты. Эти уровни конфигурируются в STEP 7.

- Защита, установленная ключом-переключателем
- Защита от записи
- Защита от записи и чтения

В случае передачи правильного пароля снимаются все уровни защиты для данного соединения. Если вместо пароля передается пустая строка, защита соединения вновь вступает в силу.

На этапе конфигурирования для двухсторонних соединений можно передать уровень защиты, применяемый по умолчанию. Если защита для двухсторонних соединений на этапе конфигурирования не установлена, все сервисы этих соединений будут доступны, даже если ключ CPU установлен в положение "RUN".

В следующей таблице поясняется взаимосвязь между передаваемым паролем и уровнем защиты для соединений, конфигурируемых на одной стороне.

Соединения, сконфигурированные на одной стороне	Устанавливаемый уровень защиты			
	Пароль не был передан		Был передан правильный пароль	
	Положение ключа		Положение ключа	
	RUN	RUN-P STOP	RUN	RUN-P STOP
Положение ключа "RUN" - можно отменить паролем	Чтение	Чтение и запись	Чтение и запись	Чтение и запись
Положение ключа "RUN" - нельзя отменить паролем	Чтение	Чтение и запись	Чтение	Чтение и запись

Соединения, сконфигурированные на одной стороне	Устанавливаемый уровень защиты			
	Защита от записи с помощью пароля	Чтение	Чтение	Чтение и запись
Защита от записи/чтения	Нет	Нет	Чтение и запись	Чтение и запись

Таблица 4-1 Взаимосвязь между передаваемым паролем и уровнем защиты

Пояснения:

"Чтение и запись" = доступны все сервисы; "Чтение" = только чтение блоков; "Нет" = сервисы обслуживания блоков не доступны

4.5.17 Синтаксис управляющих переменных для паролей

Синтаксис

S7:[<connectionname>]&password()

Пояснения

S7

Протокол S7 для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

&password()

Идентификатор сервиса управления паролями.

Пароль передается после записи значения.

Для передачи пароля можно использовать одно из следующих представлений:

- Строка октетов
Строка, состоящая из отдельных символов, представленных шестнадцатеричными кодами, разделенными точкой (до 8 символов).
- Строка
Строка из любых алфавитно-цифровых символов, начинающаяся с символа **&** (до 8 символов)

В ответ на операцию записи пароля возвращаются следующие значения:

- OPC_E_BADRIGHTS
Пароль не верен
- S_OK
Пароль верен

Тип данных OLE	Тип в Visual Basic
String (строка)	String (строка)

4.5.18 Пример использования паролей

Ниже приведен пример передачи паролей.

Пример передачи пароля

**S7:[S7-OPC-1]&password()
&password()**

Чтобы передать пароль **SetMeFre** в программируемый контроллер, в переменную необходимо записать следующее значение:

&SetMeFre

4.5.19 Несконфигурированное S7-соединение

Доступ к партнерскому устройству при отсутствии конфигурации

В общем случае, соединения, устанавливаемые с партнерскими устройствами, определяются в конфигурации. Это делается с помощью программ STEP 7 или NCM PC / NCM S7. Встречаются, однако, ситуации, когда, например, требуется выполнить чтение данных из партнерского устройства либо выполнить запись или мониторинг переменных. Эти задачи можно реализовать и в отсутствие конфигурации.

Требования

Доступ к устройству при отсутствии конфигурации возможен, если известны все коммуникационные параметры партнерского устройства. К этим данным относятся имя соединения, точка доступа и адрес станции. Необходимые параметры описаны в настоящем разделе. При выборе имени соединения необходимо обеспечить, чтобы оно было уникальным и не повторяло имя, уже имеющееся среди сконфигурированных имен.

Параметры для несконфигурированного соединения

Для несконфигурированного соединения используется следующий синтаксис:

```
S7:[<connectionname>|<VFD>|<access>|
  <address_specification>]<dataelement>
```

Элементы синтаксиса имеют следующее значение:

- *<connectionname>*
Соединения с таким именем существовать не должно. Если выбранное имя соединения уже было сконфигурировано ранее и используется для другого произвольно определенного S7-соединения, в этом случае дополнительные сведения о произвольно определяемых соединениях игнорируются и элемент присваивается существующему соединению.
- *<VFD>*
Любое имя VFD. Все соединения можно создать в рамках одного VFD. Длина имени VFD не должна превышать 32 символа. Нельзя использовать уже существующее имя VFD.
- *<accesspoint>*
Точка доступа коммуникационного модуля должна быть сконфигурирована заранее с помощью программы "Configuration Console" (Консоль конфигурирования).

- $\langle address_specification \rangle = \langle local\ TSAP \rangle, \langle stationaddress \rangle, \langle remote\ TSAP \rangle, \langle mode \rangle$

В спецификации адреса содержатся следующие сведения, разделяемые запятыми:

- **Local TSAP (Локальная Точка Доступа к Транспортному Сервису)**
Локальная точка TSAP для протокола S7 состоит ровно из двух чисел, разделяемых пробелами или точкой и имеющих следующее значение:

Первый байт: Может содержать идентификатор устройства.
02 OS (Операторская станция для визуализации и управления)
03 Прочие устройства

Второй байт: Всегда 0.
Рекомендуемое значение: 01.00

- **stationaddress (адрес станции)**
Адрес станции можно представить в одном из трех форматов:

Способ передачи	Представление адреса станции	Пример
PROFIBUS	Адрес PROFIBUS, десятичный формат	65
TCP/IP	Адрес TCP/IP	192.168.0.7
ISO	Адрес MAC	08-00-06-01-3a-00

- **Remote TSAP (Удаленная Точка Доступа к Транспортному Сервису)**
Представляется в том же формате, что и локальная точка TSAP, хотя второй байт имеет другое значение:

Первый байт: Содержит идентификатор устройства:
02 OS (Операторская станция для визуализации и управления)
03 Другие устройства
Рекомендуемое значение:: 02

Второй байт: Содержит адрес SIMATIC S7 CPU, который состоит из:
биты 7... 5 Стойка (подсистема) S7 CPU
биты 4... 0 Слот в модуле S7 CPU

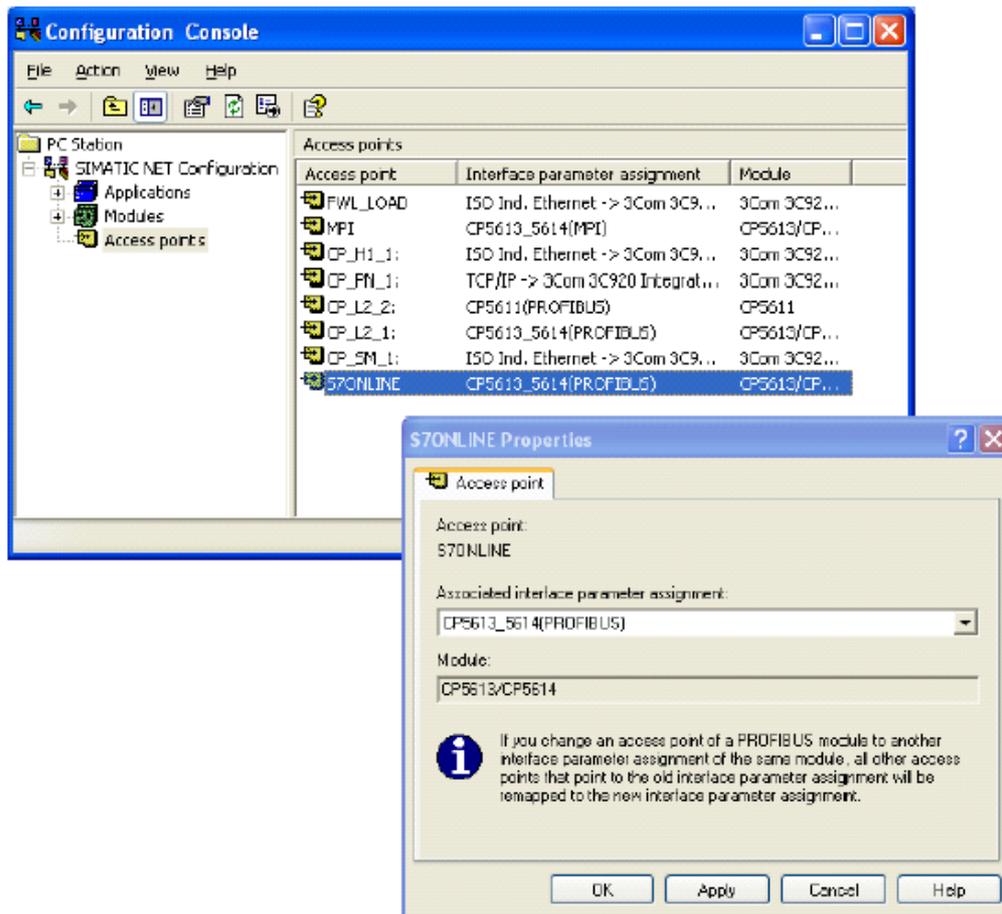
- **mode (режим)**
S7-соединения устанавливаются либо OPC-сервером, либо коммуникационным партнером. Через такие соединения также возможен оптимизированный доступ для записи или чтения. Для режима можно выбрать одно из следующих значений:
 - 1 Установление активного соединения с оптимизацией
 - 3 Установление активного соединения без оптимизации
- **<dataelement>**
Здесь, к примеру, можно указать номер и тип блока данных, (байт, слово и т.п.), а также адрес (например, смещение в байтах). В главе "OPC-переменные процесса для SIMATIC NET" (раздел "S7-коммуникации") поясняется, какие элементы данных предусмотрены для S7.

Примеры

```
S7:[S7 connection 1|VFD1|S7ONLINE|01.00,192.168.0.7,02.02,1]DB10,B0  
S7:[S7 connection 2|VFD2|S7ONLINE|01.00,65,02.02,1]DB10,B0  
S7:[S7 connection 3|VFD3|S7ONLINE|01.00,08.06.05.e4.3a.00,02.02,1]MB0
```

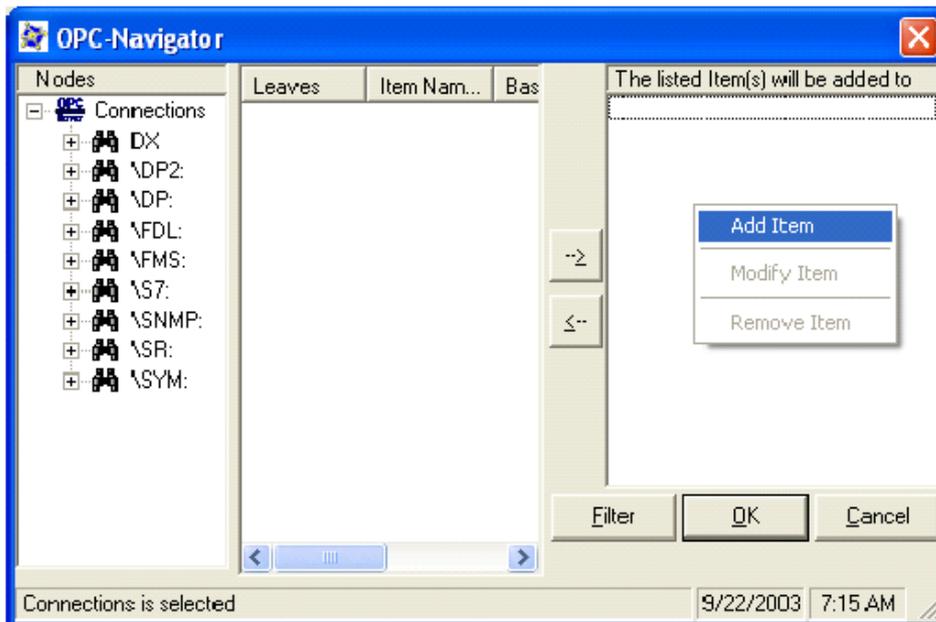
Определение точки доступа

Чтобы определить точку доступа и присвоить ей назначение параметров интерфейса (interface parameter assignment), запустите программу "Configuration Console" (Консоль конфигурирования).



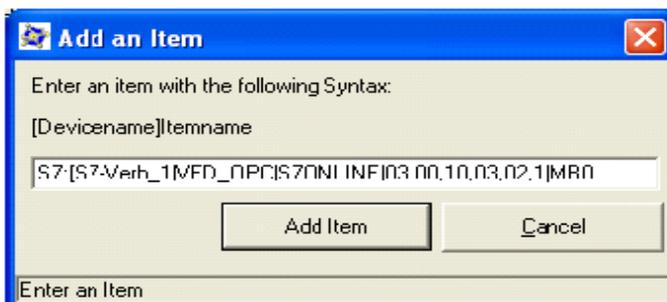
Конфигурирование программы-клиента

Откройте программу-клиент и создайте элемент, используя приведенный выше синтаксис. В программе OPC Scout откройте панели для вставки элементов, вызвав в правой части окна программы всплывающее меню и выбрав команду "Add Item" (Добавить элемент). В настоящее время обзор неконфигурированного соединения не поддерживается.



Добавьте элемент

В диалоговом окне "Add Item" (Добавить элемент) введите элемент с описанными ранее параметрами и щелкните по кнопке "Add Item" (Добавить элемент). При отсутствии ошибок синтаксиса элемент появится в пространстве имен, в ветви "S7".



После того, как элемент добавлен и является активным, соединением можно пользоваться, как сконфигурированным соединением. Это означает, что можно просматривать пространство имен, а также добавлять другие элементы, не используя синтаксис неконфигурированного соединения. Требуется лишь указывать имя соединения, например: S7:[S7-Verb_1]MB1.

4.6 PROFIBUS FMS

Введение

Протокол FMS для PROFIBUS – это открытый коммуникационный протокол, относящийся к уровню 7 эталонной модели ISO/OSI.

С помощью FMS можно устанавливать связь с любыми программируемыми контроллерами, поддерживающими стандарт FMS. Это, в частности, относится к программируемым контроллерам S5 и S7 фирмы Siemens.

Характеристики PROFIBUS–FMS

OPC-сервер SIMATIC NET обладает следующими характеристиками:

Чтение и запись FMS-переменных.

- Поддержка функций FMS-сервера
- Поддержка массивов и структур, включая доступ к отдельным элементам
- Поддержка соединений, через которые нельзя загрузить словарь объекта (OD), короткий OD или длинный OD
- Поддержка FMS-сервисов Identify и VfdState
- Поддержка PROFIBUS-сервиса Lifelist
- Отображение и мониторинг состояния соединения

4.6.1 Идентификатор протокола (FMS)

Для *протокола FMS* используется идентификатор протокола (protocol ID) **FMS**.

4.6.2 Имена соединений

Имя соединения – это имя, сконфигурированное в STEP 7 с целью уникальной идентификации соединения. На этапе конфигурирования в STEP 7 обеспечивается, чтобы имя FMS-соединения для OPC-сервера было уникальным.

Примеры имен соединений

Типичными именами соединений являются:

Simocode

S5_5431

connection1

4.6.3 Типы данных

В протоколе FMS используются собственные типы данных при объявлении FMS-переменных. Типы данных FMS должны быть приведены к типам данных, поддерживаемым OPC-интерфейсом.

* По умолчанию выбран тип VT_BSTR

Тип данных FMS	Канонический тип данных OLE	Тип в Visual Basic	Назначение
unsigned8	VT_UI1	Byte	Однобайтовое целое без знака (символ)
unsigned16	VT_UI2	Long	Короткое целое без знака (2 байта)
unsigned32	VT_UI4	Double	Целое двойное слово без знака (4 байта)
integer8	VT_I1	Integer	Однобайтовое целое со знаком (символ)
integer16	VT_I2	Integer	Короткое целое со знаком (2 байта)
integer32	VT_I4	Long	Целое двойное слово со знаком (4 байта)
float	VT_R4	Single	Вещественное (4 байта)
boolean	VT_BOOL	Boolean	Короткое целое со знаком (True = -1; False = 0)
vstringX	VT_BSTR	String	Отображаемая строка X: Количество символов
bstringX	VT_BSTR	String	Битовая строка X: Количество битов Структура: b.b.b.b.b где b=0 или 1
ostringX	VT_BSTR	String	Строка октетов X: количество октетов Структура: hh.hh.hh.hh.hh где h=0..f

Тип данных FMS	Канонический тип данных OLE	Тип в Visual Basic	Назначение
time_of_day	VT_DATE	Date	<p>Время суток TimeOfDay4 (4 байта) 0 ... 86399999 мс (альтернатива VT_UI4)</p> <p>Дата в OPC всегда имеет значение 01.01.1984 и при записи игнорируется.</p> <p>TimeOfDay6 (6 байтов) с 01.01.1984 00:00:00.000 по 31.12.2083 23:59:59.999</p>
Date	VT_DATE	Date	<p>Дата 7 байтов с 01.01.1984 00:00:00.000 по 1.12.2083 23:59:59.999</p>
time_difference	VT_DATE	Date	<p>Разница во времени TimeDifference4 (4 байта) 0 ... 86399999 мс (альтернатива VT_UI4)</p> <p>В OPC дата всегда имеет значение 30.12.1899 и игнорируется при записи.</p> <p>TimeDifference6 (6 байтов) от 00:00:00.000 0 дней до 23:59:59.999 65535 дней (альтернатива VT_UI8) Дата в OPC начинается с 30.12.1899.</p>
array (Data type[N])	Массив VT_ARRAY из элементов определенного типа	Массив из элементов с определенным типом VB	Массив FMS-переменных, состоящий из N элементов определенного типа.
struct ({Type1 Type2 Type3 Type4})	Массив VT_ARRAY из элементов типа VT_VARIANT	Массив из элементов соответствующих элементарных типов	Структура FMS-переменных с четырьмя различными типами данных Тип 1...Тип 4.

4.6.4 Сервисы для работы с переменными

Под FMS-переменными понимаются переменные коммуникационного партнера. Обращение к этим переменным выполняется с помощью словаря объекта, предоставляемого партнером.

На стороне устройства-партнера имеется два типа словарей объекта:

- Краткий словарь объекта
Для обращения к переменным можно использовать только индексы.
- Расширенный словарь объекта
Можно использовать как символьные имена, хранящиеся в словаре объекта, так и индексы.

Выбор словаря объекта, который будет использоваться для обращения к переменным, осуществляется на этапе конфигурирования.

4.6.5 Синтаксис переменных процесса для сервисов обслуживания переменных FMS

Синтаксис

Имеется две формы синтаксиса:

FMS:[<connectionname>]<index>{:<subindex>}

FMS:[<connectionname>]<objectname>{:<subindex>}

Пояснения

FMS

Протокол FMS для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

<index>

Индекс переменной в кратком словаре объекта партнерского устройства.

<objectname>

Символьное имя переменной в расширенном словаре объекта в партнерском устройстве.

<subindex>

Поэлементное обращение к элементам структуры.

4.6.6 Примеры переменных процесса для сервисов обслуживания FMS-переменных

Ниже приведены примеры, демонстрирующие синтаксис имен FMS-переменных.

Адресация с использованием символьных имен через демонстрационное соединение

FMS:[DEMO]Valve_In

Valve_In

Обращение к переменной **Valve_In** через демонстрационное соединение.

Адресация с использованием индекса через обычное соединение

FMS:[CR1]200

200

Обращение к переменной с индексом 200 через соединение CR1.

Адресация с использованием символьных имен через обычное соединение

FMS:[CR1]Count

Count

Обращение к переменной **Count** через соединение CR1.

Обращение к элементу структуры через обычное соединение

FMS:[CR1]200:2

200:2

Обращение ко второму элементу структуры переменной с индексом 200

4.6.7 Специальные информационные переменные протокола FMS

OPC-сервер предоставляет переменные, позволяющие получить информацию о системе связи FMS и о состояниях соединений.

Можно прочитать следующую информацию:

- Атрибуты виртуального полевого устройства (VFD).
- Состояние виртуального устройства
- Состояние FMS-соединения
- Адреса станций доступных узлов (не работающих)

4.6.8 Синтаксис специальных информационных переменных протокола FMS

Синтаксис

FMS:[<connectionname>]&<informationparameter>()

Пояснения

FMS

Протокол FMS для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

<informationparameter>

Возможны следующие значения:

identify

Атрибуты производителя виртуального устройства.

Виртуальное полевое устройство (VFD) – это модель реального устройства.

Возвращаемые значения:

- Производитель
- Модель
- Версия

Тип данных OLE	Тип в Visual Basic
Массив VT_ARRAY из элементов VT_BSTR	String() (строка из трех элементов)

vfdstate

Состояние виртуального полевого устройства

Тип данных: массив VT_ARRAY, состоящий из следующих компонентов типа VT_VARIANT:

Логическое состояние

Поддерживаемые сервисы

Возвращаемые значения:

FMS_STATE_CHANGES_ALLOWED	Разрешены все сервисы.
FMS_LIMITED_SERVICES_PERMITTED	Разрешены только сервисы Status virtual device и Identify virtual device .
FMS_OD_LOADING_NON_INTERACTING	Загрузка словаря объекта не разрешена
FMS_OD_LOADING_INTERACTING	В настоящее время в интерфейсный модуль загружается словарь объекта.

Тип данных OLE	Тип в Visual Basic
VT_BSTR	Массив из трех элементов строкового типа

Физическое состояние

Работоспособность реального устройства

Возвращаемые значения:

FMS_OPERATIONAL	Реальное устройство работоспособно.
FMS_PARTIALLY_OPERATIONAL	Реальное устройство частично работоспособно.
FMS_NEEDS_COMMISSIONING	Реальное устройство будет работоспособно после выполнения локальной настройки.
FMS_INOPERABLE	Реальное устройство неработоспособно.

Тип данных OLE	Тип в Visual Basic
VT_BSTR	Массив из трех элементов строкового типа

Подробные сведения о состоянии локального VFD

Подробное описание состояния локального VFD передается в виде строки октетов, имеющей структуру **hh.hh.hh**, где h = 0..f. Для интерпретации этого значения необходимо наличие описания локального VFD.

Тип данных OLE	Тип в Visual Basic
VT_BSTR	Массив из трех элементов строкового типа

statepath

Состояние коммуникационного соединения с партнерским устройством.

Возвращаемые значения:

- DOWN
Соединение не установлено
- UP
Соединение установлено
- RECOVERY
Соединение устанавливается
- ESTABLISH
Словарь объекта партнера прочитан.

Тип данных OLE	Тип в Visual Basic
VT_BSTR	String (строка)

statepathval

Состояние коммуникационного соединения с партнерским устройством.

Возвращаемые значения:

- 1
Соединение не установлено.
- 2
Соединение установлено.
- 3
Соединение устанавливается.
- 4
Словарь объекта партнерского устройства прочитан.
- Другие значения.
Состояние соединения неизвестно.

Тип данных OLE	Тип в Visual Basic
VT_UI1	Byte (Байт)

livelist

Соответствует сервису *Lifelist* протокола PROFIBUS FMS. Содержит сведения о доступных узлах шины.

В 127-ми элементах этого массива содержатся сведения о каждом возможном адресе станции. Каждому адресу станции соответствует определенный индекс массива

Значения, возвращаемые в элементах массива:

- FMS_STATION_NONEXISTENT
Узлы отсутствуют (значение 0)
- FMS_STATION_PASSIVE
Пассивный узел (значение 1)
- FMS_STATION_READY_FOR_RING
Узел готов к включению в маркерное кольцо PROFIBUS (значение 2)
- FMS_STATION_ACTIVE
Активный узел (значение 3)

Тип данных OLE	Тип в Visual Basic
Массив VT_ARRAY из элементов типа VT_UI1	Массив из 127 элементов типа byte() (байт)

4.6.9 Примеры специальных информационных переменных протокола FMS и значений, возвращаемых в этих переменных

Ниже приведены примеры значений, возвращаемых для специальных информационных переменных протокола FMS.

Сведения об атрибутах производителя виртуального полевого устройства

***FMS:[connection1]&identify()
&identify***

Могут быть возвращены, например, следующие значения:

SIEMENS AG	Производитель
ET 200U-DP/FMS V1.0	Модель виртуального устройства Версия

Состояние устройства

***FMS:[connection1]&vfdstate()
&vfdstate()***

Могут быть возвращены, например, следующие значения:

FMS_STATE_CHANGES_ALLOWED	Разрешены все сервисы.
FMS_OPERATIONAL	Реальное устройство работоспособно.
00.00.00	Статус локального VFD.

Состояние коммуникационного соединения

***FMS:[connection1]&statepath()
&statepath()***

Могут быть возвращены, например, следующие значения:

UP	Соединение установлено.
----	-------------------------

Сведения о доступных узлах шины

**FMS:[connection1]&livelist()
&livelist()**

Может быть возвращено, например, следующее значение (отображается в виде строки):

```
{0|1|0|0|1|0|0|1|0|2|1|0|1|0|1|0|3|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|2|0|1|2|0|0|0|0|0|0|0|1|1|1|0|1|1|0|1|0|1|0|1|0|1|0|1|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|3|0|0|0|0|0|0|0|0|0|1|0|1|0|1|0|1|0|1|0|0|1|0|0|1|0|0|1|0|1|3|0|3|0|1|0|3|1|0|0|3|0|3|0|3|0|0|0|0|0|0|0|0}
```

В этом смысле &livelist() соответствует сервису Livelist протокола FMS.

4.6.10 Переменные FMS-сервера

FMS–OPC-сервер сам может быть сервером протокола FMS. В качестве FMS-сервера он предоставляет переменные и словарь объекта, к которым могут обращаться другие участники FMS-коммуникаций. Это позволяет установить связь по протоколу FMS между персональными компьютерами.

Переменные FMS-сервера создаются на этапе конфигурирования, ими управляет OPC-сервер.

Примечание

Чтобы ПК выполнял функцию FMS-сервера и был доступен для других устройств, OPC-клиентом должен быть запущен OPC-сервер.

4.6.11 Переменные FMS-сервера: ItemID

Синтаксис

Имеются две формы синтаксиса:

FMS:[<connectionname>]<index>{:<subindex>}

FMS:[<connectionname>]<symbolicname>{:<subindex>}

Пояснения

FMS

Протокол FMS для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

<index>

Индекс для переменной локального краткого словаря объекта.

<symbolicname>

Символьное имя переменной локального расширенного словаря объекта.

<subindex>

Обращение к элементу структуры.

Примечание

С точки зрения OPC-клиента эти переменные являются локальными элементами, которыми управляет OPC-сервер. Данные передаются через PROFIBUS только в том случае, когда участник FMS-коммуникаций обращается к этим переменным.

4.7 S5–совместимые коммуникации

В зависимости от применяемой сети связи различают два вида S5–совместимых коммуникаций:

- S5–совместимые коммуникации по сети Industrial Ethernet
Идентификатор протокола: SR
- S5–совместимые коммуникации по сети PROFIBUS
Идентификатор протокола: FDL

4.7.1 S5–совместимые коммуникации по сети Industrial Ethernet

S5–совместимые коммуникации через Industrial Ethernet также называют протоколом SEND/RECEIVE. Возможна связь с устройствами S5 и S7.

Характеристики S5–совместимых коммуникаций по сети Industrial Ethernet

- Для связи используются управляющие блоки (handling block) SIMATIC S5 и функциональные блоки (function block) S7.
- Возможна связь между двумя ПК-станциями.
- Поддерживаются функции WRITE и FETCH для доступа к объектам коммуникационного партнера.
- Возможен быстрый доступ к большим пакетам данных в режимах SEND и RECEIVE.
- Возможен поэлементный доступ к данным внутри пакета данных.
- Возможно отображение и контроль состояния соединения.

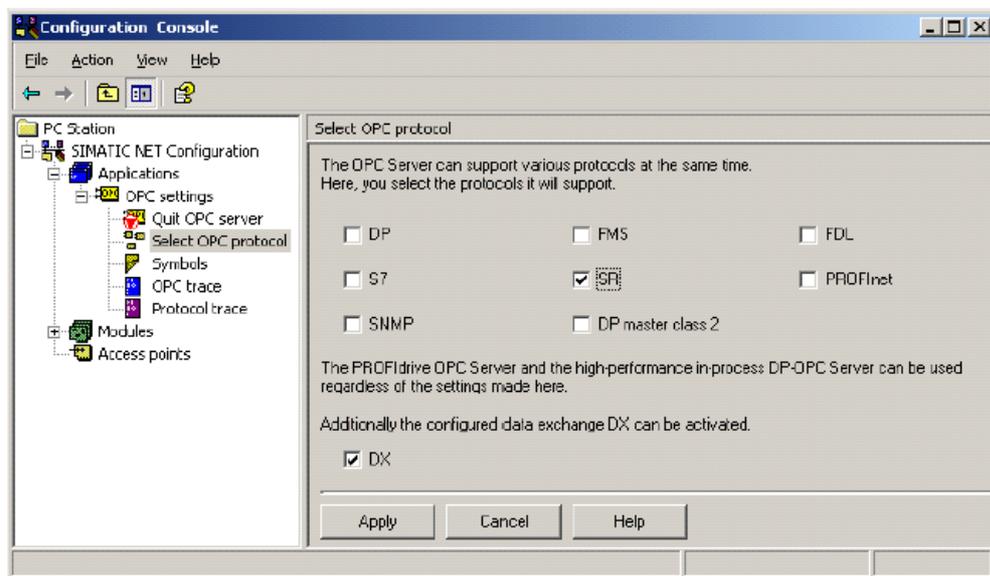
4.7.2 Мощный OPC-сервер семейства SIMATIC NET для протокола SR

Введение

В этом разделе описан вариант конфигурации для применения протокола SR, удовлетворяющий требованиям высокой производительности. Этот вариант можно использовать в том случае, если все базовые библиотеки протокола SR, а также COM-сервер в качестве внутривычислительного сервера, загружаются в пространство вневычислительного OPC-сервера. Управление протоколом производится в пространстве процесса OPC-сервера, что позволяет избежать дополнительных временных затрат на переключение между процессами и применения мультипротокольного режима. По-прежнему, однако, остается необходимость в переключении между процессами OPCклиента и OPC-сервера.

Конфигурация

Этот вариант повышенной производительности вступает в силу автоматически в результате выбора протокола SR в программе Configuration Console ("Консоль конфигурирования").



Также можно выбрать сконфигурированный обмен данными через OPC (DX) и применение символов.

Преимущества/ Недостатки

Применение высокопроизводительного SR OPC-севера, однако, обладает недостатком, который состоит в том, что используется только протокол SR (однопротокольный режим). С другой стороны, это дает пользователю следующие преимущества::

- Более высокая производительность по сравнению с мультипротокольным режимом.
- Простое конфигурирование.
- Доступ с использованием ProgID *OPC.SimaticNET*.
- Несколько клиентов могут пользоваться сервером одновременно.
- Стабильность работы OPC-сервера не зависит от клиента.

4.7.3 Идентификатор протокола

Для протокола SEND/RECEIVE используется идентификатор протокола **SR**.

4.7.4 Имена соединений

Имя соединения – это имя, сконфигурированное в STEP 7 в целях однозначной идентификации соединения.

OPC-сервер поддерживает следующие типы соединений:

- Соединение ISO transport (соединение через ISO-транспорт)
- Соединение ISO-on-TCP (соединение ISO на TCP)
- TCP-соединение

Примеры имен соединений

Типичные примеры имен соединений:

MyConnection

SR_Connection

4.7.5 Сервисы для работы с переменными

Сервисы обслуживания переменных обеспечивают прямой доступ к переменным программируемого контроллера и их мониторинг. Для обращения к переменным используются символы (символьная адресация). Синтаксис имен переменных ориентирован на применяемые средства программирования. В случае прямого доступа с целью чтения OPC-сервер передает требуемую информацию об адресе принимающей стороне, а последняя возвращает в ответ запрошенные данные. Во время сеанса записи OPC-сервер передает информацию об адресе вместе с записываемым значением.

Примечание

Запрошенная переменная должна быть согласована с конфигурацией коммуникационного партнера. В противном случае обращение к некоторым областям приведет к возникновению ошибок связи. OPC-сервер может проверить синтаксис только при регистрации переменной. На основании данных о конфигурации партнера нельзя установить, является ли переменная действительной для коммуникационного партнера.

Синтаксис переменных процесса для сервисов обслуживания переменных SEND/RECEIVE

Синтаксис

SR:[<connectionname>]<area>{,}<type><address>{,<quantity>}

Пояснения

SR

Протокол SEND/RECEIVE для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

<area>

Объект, к которому выполняется обращение

- DBnn,
Блок данных nn
- Q
Выход
- I
Вход
- M
Биты памяти
- P
Входы/выходы для CPU S5
- PII
Входы/выходы для CPU S7
- C
Счетчики
- T
Таймеры
- RS
Системная область
- IA
Абсолютный начальный адрес
- DXnn,
Расширенный блок данных
- DEnn,
Блок данных в расширенной памяти
- OY
Расширенные входы/выходы

<type>

Тип данных.

Тип данных преобразуется в соответствующий тип данных OLE в OPC-сервере.

Тип данных	Описание	Тип данных OLE	Тип в Visual Basic
X	Бит (логический тип) Только для объектов I, Q, F, P, PII и OY (доступ только для чтения)	VT_BOOL	Boolean
B или BYTE <i>Возможен только для объектов I, Q, F, P, PII и OY</i>	Байт (8 битов, без знака)	VT_UI1	Byte
CHAR <i>Возможен только для объектов I, Q, F, P, PII и OY</i>	Байт (8 битов, со знаком)	VT_I1	Integer
W или WORD	Слово (16 битов, без знака)	VT_UI2	Long
INT	Слово (16 битов, со знаком)	VT_I2	Integer
D или DWORD	Двойное слово (32 бита, без знака)	VT_UI4	Double
DINT	Двойное слово (32 бита, со знаком)	VT_I4	Long
REAL	Число с плавающей запятой в формате IEEE	VT_R4	Single
S5REAL	Число с плавающей запятой в формате S5	VT_R4	Single

<address>

Адрес переменных в области:

В зависимости от области указанный адрес должен иметь формат байта или слова. В случае обращения к перечисленным ниже областям, указанный адрес воспринимается как адрес слова: **DBnn**

- RS
- IA
- DXnn,
- DEnn,
- C
- T

<quantity>

Количество переменных указанного типа, к которым выполняется обращение, начиная с адреса, указанного в параметре *address* (адрес).

4.7.6 Сервисы для работы с буферами

Буфер-ориентированные сервисы позволяют организовать передачу больших блоков данных, управляемую программным образом. Эти сервисы также называют сервисами SEND/RECEIVE. Для передачи данных в OPC-сервере используются следующие переменные:

- Переменные, которые передают блоки данных
- Переменные, которые принимают блоки данных

Принимаемый по умолчанию размер блоков данных указывается в конфигурации. При передаче переменных длину можно ограничить. Возможен доступ к данным внутри блока данных.

Фиксированные имена переменных

Для каждого соединения указываются следующие фиксированные названия переменных:

- RECEIVE
- SEND

Синтаксис переменных процесса для буфер-ориентированных сервисов

Имеются следующие формы синтаксиса:

SR:[<connectionname>]receive{,<type><address>{,<quantity>}}

SR:[<connectionname>]send{<n>}{,<type><address>{,<quantity>}}

Пояснения

SR

Протокол SEND/RECEIVE для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

receive

Последний буфер данных, принятый от коммуникационного партнера.

Буфер данных может иметь произвольную структуру. Поэтому буфер всегда передается в виде массива, состоящего из байтов.

Тип данных OLE	Тип в Visual Basic
VT_ARRAY VT_UI1	Byte (байт)

Примечание

Переменная RECEIVE соответствует приемному буферу. Эта переменная, поэтому, может быть только прочитана. В случае обращения к устройству с целью чтения в коммуникационной системе явным образом подготавливается приемный буфер. Если в течение определенного времени в этот буфер не поступают данные, формируется уведомление о превышении времени. Следовательно, эти переменные требуется лишь контролировать или читать их из кеша.

send

Буфер передачи, который может быть передан коммуникационному партнеру.

Принимаемый по умолчанию размер буфера передачи указывается на этапе конфигурирования. Буфер всегда передается в виде массива, состоящего из байтов.

Обращение к этим переменным с целью записи всегда приводит к пересылке буфера передачи коммуникационному партнеру.

Тип данных OLE	Тип в Visual Basic
VT_ARRAY VT_UI1	Byte (Байт)

Примечание

Не следует выполнять чтение или активизацию этой переменной и переменных, получаемых из нее. Обращение к этим переменным с целью чтения может привести к тому, что коммуникационный партнер разорвет соединение.

<n>

Размер буфера передачи.

Параметр *n* можно использовать в том случае, если через соединение передаются буферы различных размеров. Если параметр *n* опущен, используется размер буфера, указанный в конфигурации.

Если применяется собственный протокол TCP/IP (TCP/IP native) и отключен минипротocolный режим (miniprotocol) (см. описание средств конфигурирования SIMATIC STEP 7 или SIMATIC NCM PC), в этом случае размер буфера передачи указать невозможно. В этой ситуации используется значение размера буфера, определенное на этапе конфигурирования.

Примечание

Для переменных, принадлежащих буферам передачи различных размеров, используются отдельные области памяти!

<type>

Тип данных.

Тип данных приводится к соответствующему типу данных OLE на стороне OPC-сервера.

Тип данных	Описание	Тип данных OLE	Тип в Visual Basic
X	Бит (логический тип)	VT_BOOL	Boolean
B или BYTE	Байт (целое без знака, 8 битов)	VT_UI1	Byte
CHAR	Байт (целое со знаком, 8 битов)	VT_I1	Integer
W или WORD	Слово (целое без знака, 16 битов)	VT_UI2	Long
INT	Слово (целое со знаком, 16 битов)	VT_I2	Integer
D или DWORD	Двойное слово (целое без знака, 32 бита)	VT_UI4	Double

Тип данных	Описание	Тип данных OLE	Тип в Visual Basic
DINT	Двойное слово (целое без знака, 32 бита)	VT_I4	Long
REAL	Число с плавающей запятой, в представлении IEEE	VT_R4	Single
S5REAL	Число с плавающей запятой, в представлении S5	VT_R4	Single

<address>

Адрес байта, определяемый параметром *byte number* (номер байта).

В случае обращения к биту данных указывается номер байта и номер бита:

Номер байта.бит.

<quantity>

Количество адресуемых переменных указанного типа, начиная с адреса, указанного параметром *address* (адрес).

Массивы битов не поддерживаются. Для типа данных **X** (логический тип) количество либо не указывается, либо устанавливается равным 1.

Примечание

В случае использования сервисов обслуживания переменных SEND/RECEIVE для систем автоматизации S7 следует помнить, что к счетчикам и таймерам нельзя обращаться как к массивам (нельзя указывать количество переменных > 1).

Примеры переменных процесса для буфер-ориентированных сервисов

Ниже приведены примеры синтаксиса имен переменных для буфер-ориентированных сервисов.

Переменные приема

SR:[MyConnection]receive,w4,6

receive,w4,6

6 слов данных в буфере приема, начиная с байта 4

SR:[MyConnection]receive,dword7

receive,dword7

Двойное слово в буфере приема, начиная с байта 7

SR:[MyConnection]receive,REAL0,2

receive,REAL0,2

Массив из двух значений с плавающей запятой в буфере приема, начиная с байта 0.

Переменные передачи

SR:[MyConnection]send30,dword7

send30,dword7

Двойное слово в буфере передачи размером в 30 байтов, начиная с байта 7. Если принимаемое по умолчанию значение размера буфера передачи отличается от 30, переменная обращается к отдельному буферу.

SR:[MyConnection]send,B20,6

send,B20,6

Массив из 20 байтов в буфере передачи с принимаемым по умолчанию размером, начиная с байта 6. Принимаемый по умолчанию размер буфера передачи указывается в конфигурации.

SR:[MyConnection]send8,DINT0

send8,DINT0

Двойное слово со знаком в буфере передачи размеров 8, начиная с адреса 0.

4.7.7 Специальные информационные переменные для сервисов SEND/RECEIVE

Специальные информационные переменные для SEND/RECEIVE позволяют получить информацию о состоянии соединения.

Можно получить следующую информацию:

Состояние соединения

Синтаксис информационных переменных для S5-совместимых коммуникаций (SEND/RECEIVE)

SR:[<connectionname>]&<informationparameter>()

Пояснения

SR

Протокол SEND/RECEIVE для обращения к переменной процесса.

<connectionsname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

<informationparameter>

Можно ввести следующий параметр:

statepath

Состояние коммуникационного соединения с устройством-партнером.

Результат представляется в виде строки.

Возвращаемые значения:

- DOWN
Соединение не установлено.
- UP
Соединение установлено.
- RECOVERY
Соединение устанавливается.
- ESTABLISH
Зарезервировано для использования в будущем.

Тип данных OLE	Тип в Visual Basic
VT_BSTR	String (строка)

statepathval

Состояние коммуникационного соединения с устройством-партнером.

Результат представляется в виде числа.

Возвращаемые значения:

- 1
Соединение не установлено
- 2
Соединение установлено
- 3
Соединение устанавливается
- 4
Зарезервировано для использования в будущем.

Тип данных OLE	Тип в Visual Basic
VT_UI1	Byte (байт)

Примеры специальных информационных переменных для SEND/RECEIVE

Ниже приведено несколько примеров возвращаемых значений информационных переменных для S5–совместимых коммуникаций (SEND/RECEIVE):

SR:[SR_CONNECTION]&statepath()
&statepath()

Может быть, например, возвращено следующее значение:

UP	Соединение установлено.
----	-------------------------

SR:[SR_CONNECTION]&statepathval()
&statepathval()

Может быть, например, возвращено следующее значение:

2	Соединение установлено.
---	-------------------------

4.7.8 S5–совместимые коммуникации по сети PROFIBUS

Эталонная модель ISO/OSI

S5–совместимые коммуникации в сети PROFIBUS реализуются на основе простого протокола для устройств SIMATIC, предусматривающего передачу кадров. Для S5–совместимых коммуникаций используются сервисы Fieldbus Data Link (FDL) уровня 2 эталонной модели ISO/OSI (PROFIBUS).

Характеристики S5–совместимых коммуникаций по сети PROFIBUS

OPC-сервер SIMATIC NET характеризуется следующими свойствами:

- Для организации связи в программируемом контроллере используются управляющие блоки SIMATIC S5 и SIMATIC S7.
- Возможна связь между двумя ПК-станциями.
- Возможна быстрая передача пакетов данных.
- Возможна структуризация обращений к данным в пределах пакета данных.
- Возможно отображение и контроль состояния соединений.

4.7.9 Идентификатор протокола

Для протокола FDL используется идентификатор (Protocol ID) **FDL**.

4.7.10 Имена соединений

Имя соединения – это имя, сконфигурированное в STEP 7 с целью однозначной идентификации соединения. Для OPC-сервера следует выбрать уникальное имя FDL-соединения.

Примеры имен соединений

Типичный пример имени:

FDL_Connection

4.7.11 Сервисы для работы с буферами

С помощью буфер-ориентированных сервисов можно организовать передачу блоков данных, управляемую программным образом. Буферы принимаемых данных и передаваемые буферы данных сопоставляются с соответствующими OPC-переменными.

Фиксированные имена переменных

Для каждого соединения указываются следующие фиксированные имена переменных:

- RECEIVE
- SEND
- SendSDA
- SendSDN

Синтаксис переменных процесса для буфер-ориентированных сервисов (FDL)

Имеются следующие формы синтаксиса:

FDL:[<connectionname>]**receive**{,}<type><address>{,<quantity>}

FDL:[<connectionname>]**send**{<n>}{,}<type><address>{,<quantity>}

Примечания:

FDL

Протокол FDL для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

receive

Последний буфер данных, принятый от коммуникационного партнера.

Структура буфера данных может быть произвольной. Поэтому буфер всегда передается в виде массива, состоящего из байтов.

Примечание

Переменная RECEIVE соответствует приемному буферу. Эта переменная, поэтому, может быть только прочитана. В случае обращения к устройству с целью чтения в коммуникационной системе явным образом подготавливается приемный буфер. Если в течение определенного времени в этот буфер не поступают данные, формируется уведомление о превышении времени. Следовательно, эти переменные требуется лишь контролировать или читать их из кеша.

Тип данных OLE	Тип в Visual Basic
T_ARRAY VT_UI1	Массив из элементов типа byte (байт).

send

Буфер передачи, присутствующий постоянно, который может быть передан коммуникационному партнеру.

Принимаемое по умолчанию значение размера буфера передачи указывается на этапе конфигурирования. Буфер всегда передается в виде массива, состоящего из байтов.

Запись в эту переменную и в переменные, получаемые из нее, приводит к передаче буфера коммуникационному партнеру.

Чтение и активизацию этой переменной и переменных, получаемых из нее, производить не следует.

В зависимости от сочетания адреса станции и точек SAP, указанных в конфигурации соединения, при записи передаваемого элемента используется соответствующий специальный сервис FDL:

Локальная точка SAP	Удаленная станция	Удаленная точка SAP	Значение/Сервис, используемый для передачи
0..62, 255	0..126	0..62, 255	Передача и прием /SDA
0..62, 255	0..126	63	Только передача / SDA
63	0..126	0..62, 255	Только прием (без передачи)
0..62, 255	127	63	Широковещание: только передача всем узлам / SDN

Локальная точка SAP	Удаленная станция	Удаленная точка SAP	Значение/Сервис, используемый для передачи
63	127	0..62, 255	Прием в режиме широковещания (без передачи)
0..62, 255	127	0..62, 255	Групповая передача: передача только тем участникам, которые активизировали удаленную точку SAP / SDN

Тип данных OLE	Тип в Visual Basic
VT_ARRAY VT_UI1	Массив из элементов типа byte (байт).

<n>

Размер буфера передачи.

Буфер-ориентированные сервисы позволяют пользователю передавать и принимать буферы данных. Буферы принимаемых данных и буферы передаваемых данных сопоставляются с OPC-переменными. Массив достаточно большого размера, к примеру, может вместить полностью весь буфер приема. Также имеется возможность назначения отдельных частей буфера отдельным переменным.

Примечание

Для переменных, относящихся к буферам передачи различных размеров, используются отдельные области памяти!

<type>

Тип данных.

Тип данных преобразуется в соответствующий тип данных OLE в OPC-сервере.

Тип данных	Описание	Тип данных OLE	Тип в Visual Basic
X	Бит (логический тип)	VT_BOOL	Boolean
B или BYTE	Байт (целое без знака, 8 битов)	VT_UI1	Byte
CHAR	Байт (целое со знаком, 8 битов)	VT_I1	Integer
W или WORD	Слово (целое без знака, 16 битов)	VT_UI2	Long
INT	Слово (целое со знаком, 16 битов)	VT_I2	Integer
D или DWORD	Двойное слово (целое без знака, 32 бита)	VT_UI4	Double
DINT	Двойное слово (целое без знака, 32 бита)	VT_I4	Long
REAL	Число с плавающей запятой, в представлении IEEE	VT_R4	Single
S5REAL	Число с плавающей запятой, в представлении S5	VT_R4	Single

<address>

Адрес байта, определяемый параметром *byte number* (номер байта).

В случае обращения к биту данных указывается номер байта и номер бита:

Номер байта.бит

<quantity>

Количество адресуемых переменных указанного типа, начиная с адреса, указанного параметром *address* (адрес).

Массивы битов не поддерживаются. Для типа данных X (логический тип) количество либо не указывается, либо устанавливается равным 1.

Синтаксис переменных процесса для буфер-ориентированных сервисов с применением SDA и SDN (FDL)

В некоторых случаях может оказаться выгодным отсутствие зависимости используемого сервиса от комбинации адреса партнерской станции и точек SAP.

Имена **SendSDA** и **SendSDN** позволяют выбрать для передачи только соответствующие сервисы **SDA** или **SDN**. Эти специальные имена не отображаются в окне OPC-браузера (OPC Browser).

Синтаксис:

Имеются две формы синтаксиса:

FDL:[<connectionname>]**SendSDA**{<n>}{,}<type><address>
{,<quantity>}

FDL:[<connectionname>]**SendSDN**{<n>}{,}<type><address>
{,<quantity>}

Пояснения

FDL

Протокол FDL для обращения к переменной процесса.

<connectionname>

Имя соединения, зависящее от протокола. Имя соединения указывается в конфигурации.

SendSDA

SendSDN

Для передачи используются только сервисы SDA или SDN.

<n>

Размер буфера передачи.

Буфер-ориентированные сервисы позволяют передавать и принимать буферы данных. Буферы передаваемых данных и буферы принимаемых данных сопоставляются с OPC-переменными. Массив достаточно большого размера, например, может вместить весь буфер приема. Также имеется возможность назначения отдельных частей буфера отдельным переменным.

<type>

Тип данных.

Тип данных приводится к соответствующему типу данных OLE в OPC-сервере.

Тип данных	Описание	Тип данных OLE	Тип в Visual Basic
X	Бит (логический тип)	VT_BOOL	Boolean
B или BYTE	Байт (целое без знака, 8 битов)	VT_UI1	Byte
CHAR	Байт (целое со знаком, 8 битов)	VT_I1	Integer
W или WORD	Слово (целое без знака, 16 битов)	VT_UI2	Long
INT	Слово (целое со знаком, 16 битов)	VT_I2	Integer
D или DWORD	Двойное слово (целое без знака, 32 бита)	VT_UI4	Double
DINT	Двойное слово (целое без знака, 32 бита)	VT_I4	Long
REAL	Число с плавающей запятой, в представлении IEEE	VT_R4	Single
S5REAL	Число с плавающей запятой, в представлении S5	VT_R4	Single

<address>

Адрес байта, определяемый параметром *byte number* (номер байта).

В случае обращении к биту данных указывается номер байта и номер бита:

Номер байта.бит

<quantity>

Количество адресуемых переменных указанного типа, начиная с адреса, указанного параметром *address* (адрес).

Массивы битов не поддерживаются. Для типа данных X (логический тип) количество либо не указывается, либо устанавливается равным 1.

Примеры переменных процесса для буфер-ориентированных сервисов (FDL)

Ниже приведены примеры синтаксиса имен FDL-переменных.

Переменные для чтения

FDL:[MyConnection]receive,w0,6

6 слов данных в буфере приема, начиная с байта 0

FDL:[MyConnection]receive,dword7

Двойное слово, начиная с байта 7

FDL:[MyConnection]Receive,REAL0,2

Два вещественных значения (тип real) в буфере приема, начиная с байта 0

Переменные для записи

FDL:[MyConnection]send30,dword7

Двойное слово в буфере передачи длиной в 30 байтов, начиная с байта 7.

FDL:[MyConnection]SendSDN,B5,20

Массив из 20 байтов в буфере передачи с принимаемым по умолчанию размером, начиная с байта 5. Принимаемый по умолчанию размер буфера указывается на этапе конфигурирования. Независимо от того, какая точка SAP указана в конфигурации, используется сервис PROFIBUS–FDL SDN.

FDL:[MyConnection]Send8,DINT0

Двойное слово со знаком в буфере передачи размером в 8 байтов, начиная с адреса 0.

4.7.12 Специальные информационные переменные для FDL

OPC-сервер, предназначенный для применения протокола SEND/RECEIVE в сети PROFIBUS (FDL), предоставляет переменные, позволяющие получить информацию о состоянии и характеристиках коммуникационной сети.

Синтаксис специальных информационных переменных для FDL

FDL:[|<CPname>]&<informationparameter>()

Пояснения

FDL

Протокол FDL для обращения к переменной процесса.

CPname

Имя коммуникационного процессора (CP) указывается на этапе конфигурирования. Поддерживаются только те имена CP, которые позволяют сконфигурировать FDL-соединение.

Информационные параметры

Предусмотрено пять следующих вариантов:

- busparameter
- defaultsap
- identify•ts
- lifelist

Эти параметры описаны ниже.

Busparameter

Чтение параметров шины (сети) PROFIBUS, используемой для указанного соединения. Эти значения возвращаются в виде массива байтов и соответствуют результатам, возвращаемым FDL-сервисом FDL_READ_VALUE. Более подробные сведения можно найти в руководстве по интерфейсу программирования FDL.

Содержание	Описание	Тип данных	Длина (байтов)
HSA	Наивысший адрес шины PROFIBUS: 2...126.	BYTE	1
TS	PROFIBUS-адрес локальной станции: 0...HSA или 126.	BYTE	1
Station_Type	Тип локальной станции.	INTEGER	2
Baudrate	Скорость передачи.	INTEGER	2
Medium_red	Резервирование.	INTEGER	2
Retry_Ctr	Количество повторных вызовов для не отвечающей станции (удаленной):0...7.	UWORD	2
Default_SAP	Номер принимаемой по умолчанию точки SAP станции (локальной):0...63.	BYTE	1
network_con_SAP	Резерв.	BYTE	1
TSL	Время ожидания приема (SLOT).	UWORD	2
TQUI	Время "молчания" передатчика / время переключения репитера.	UWORD	2
TSET	Время реакции (Setup time).	UWORD	2
MIN_TSDR	Минимальное время задержки станции.	UWORD	2
MAX TSDR	Максимальное время задержки станции.	UWORD	2
TTR	Заданное время обращения маркера.	DWORD	4
GAP	GAP- коэффициент обновления.	BYTE	1
in_Ring_desired	Запрос на включение в кольцо.	BOOLEAN	1
physical_layer	Выбираемые физические характеристики шины.	INTEGER	2
ident	Наименование производителя, тип контроллера, версии аппаратных и программных средств.	STRING	211

Тип данных OLE	Тип в Visual Basic
VT_ARRAY VT_UI1	Byte() (Байт)

defaultsap

Возвращается значение принимаемой по умолчанию точки SAP (SAP = Точка Доступа к Сервисам). Если явно не задана другая точка SAP, используется точка SAP, принимаемая по умолчанию.

Тип данных OLE	Тип в Visual Basic
VT_UI1	Byte (Байт)

identify

Чтение идентификатора станции для указанного соединения в виде массива, состоящего из 4 строк:

Возвращаемое значение содержат элементы:

- Производитель
- Контроллер
- Аппаратная версия
- Программная версия

Тип данных OLE	Тип в Visual Basic
VT_ARRAY VT_BSTR	String() (Строка)

ts

Возвращает локальный адрес станции указанного модуля.

Тип данных OLE	Тип в Visual Basic
VT_UI1	Byte (Байт)

lifelist

Сведения о доступных узлах шины.

Массив из 127 элементов содержит информацию обо всех имеющихся станциях. Каждому адресу станции соответствует свой индекс в массиве.

В элементах массива возвращаются следующие значения:

- **FDL_STATION_NON_EXISTENT**
Узлы отсутствуют (значение 0x10)
- **FDL_STATION_PASSIVE**
Пассивный узел (значения 0x00)
- **FDL_STATION_READY_FOR_RING**
Узел готов к включению в маркерное кольцо PROFIBUS (значение 0x30).
- **FDL_STATION_ACTIVE**
Активный узел (значение 0x20)

Тип данных OLE	Тип в Visual Basic
Массив VT_ARRAY из элементов типа VT_UI1	Массив из 127 элементов типа byte()

Примечание

Для чтения этой информации используется FDL-сервис. "FDL_LIFE_LIST_CREATE_REMOTE". Этот сервис оказывает существенную нагрузку на шину, поэтому следует как можно реже пользоваться этой переменной.

4.8 SNMP-коммуникации по сети Industrial Ethernet

Переменные процесса, используемые для SNMP

SNMP OPC-сервер семейства SIMATIC NET предоставляет следующие переменные:

- Переменные процесса
- Информационные переменные
- Переменные прерываний (Trap)

4.8.1 Идентификатор протокола

Идентификатор протокола

Для протокола SNMP используется идентификатор протокола (protocol ID) **SNMP**.

4.8.2 Типы данных, поддерживаемые протоколом SNMP

Приведение к поддерживаемым типам данных

При объявлении MIB-объектов в протоколе SNMP используются собственные типы данных протокола SNMP. SNMP OPC-сервер приводит типы данных протокола SNMP к поддерживаемым типам данных OPC-интерфейса следующим образом:

Тип данных SNMP Код	Канонический тип данных OLE	Тип в Visual Basic	Значение
ASN_INTEGER 02h	VT_I4	Long	Двойное слово со знаком (4 байта)
ASN_INTEGER32< 02h	VT_I4	Long	Двойное слово со знаком (4 байта)
ASN_UNSIGNED32 47h	VT_UI4	Double	Двойное слово без знака (4 байта)
ASN_OCTETSTRING 04h	VT_BSTR	String	Для непечатаемых символов используется представление: "ostring:xx.xx.xx.xx"
ASN_OBJECTIDENTIFIER 06h	VT_BSTR	String	В представлении MIB .a.b.c.d.e.
ASN_IPADDRESS 40h	VT_BSTR	String	В десятичном представлении: 129.168.0.4
ASN_COUNTER32 41h	VT_UI4	Double	Двойное слово без знака (4 байта)
ASN_GAUGE32 42h	VT_UI4	Double	Двойное слово без знака (4 байта)
ASN_TIMETICKS 43h	VT_UI4	Double	Двойное слово без знака (4 байта)
ASN_OPAQUE 44h	Массив VT_ARRAY из элементов типа VT_UI1	Массив VT_ARRAY из элементов типа VT_UI1	Байт без знака

4.8.3 Переменные процесса для сервисов обслуживания переменных протокола SNMP

Переменные и профили устройств

SNMP-переменные – это переменные коммуникационного партнера. Чтобы эти переменные стали "известны" OPC-серверу, используются профили устройств. Профили устройств можно создавать с помощью MIB-компилятора (MIB compiler) из MIB-файла устройства. Затем созданная конфигурация загружается в ПК, на котором функционирует OPC-сервер, после чего переменные становятся доступными OPC-серверу.

Синтаксис

Примечание

В отличие от синтаксиса, используемого для других протоколов, для SNMP применяется следующий синтаксис.

SNMP : [<devicename>] <objectname>

Пояснения

SNMP

Протокол SNMP для обращения к переменным процесса (MIB-объектам) и переменным SNMP-прерываний.

<Teilnehmername>

Имя устройства указывается на этапе конфигурирования системы и является уникальным.

<Objektname>

Символьное имя MIB-объекта партнерского устройства.

Пример

SNMP:[OSM]sysName

Обращение к MIB-объекту sysName. В конфигурации для этого устройства было выбрано имя OSM.

4.8.4 Специальные информационные переменные протокола SNMP

Введение

Далее описываются переменные, предоставляемые SNMP OPC-сервером.

- Переменные, содержащие информацию и состояния коммуникационной системы и соединений
- Переменные SNMP OPC-сервера

Информация о коммуникационной системе

OPC-сервер предоставляет переменные, с помощью которых можно получить информацию о коммуникационной системе SNMP и состояниях соединений.

Можно получить следующую информацию:

- Комментарии, введенные в результате конфигурирования
- IP-адрес партнерского устройства
- Состояние соединения с партнерским устройством
- Дополнительные сведения об устройствах, не поддерживающих SNMP

Синтаксис

SNMP:[<devicename>]<informationparameter>

Пояснения

SNMP

Протокол SNMP для обращения к переменным процесса (MIB-объектам) и переменным SNMP-прерываний.

<devicename>

Имя устройства указывается на этапе конфигурирования системы.

<informationparameter>

Определены следующие информационные параметры:

&description()	Текст, введенный в поле комментария на этапе конфигурирования. Тип VT_BSTR, только для чтения.
&ipaddress()	IP-адрес партнерского устройства. Тип VT_BSTR, только для чтения.
&statepath()	<p>Состояние соединения с коммуникационным партнером. Значение в текстовой форме (Тип VT_BSTR, только для чтения):</p> <p>DOWN Вызовы не доходят до устройства, либо отсутствует активная задача чтения или записи.</p> <p>UP Устройство доступно для успешного выполнения задач.</p> <p>RECOVERY Вызовы не доходят до устройства, либо последнее задание было завершено с ошибкой, связанной с прекращением связи, при этом имеется еще активное задание для устройства.</p>
&statepathval()	<p>Состояние соединения с коммуникационным партнером. Значения в виде целых чисел. Тип: VT_UI1, только чтение.</p> <p>1 Вызовы не доходят до устройства, либо отсутствует активная задача чтения или записи.</p> <p>2 Устройство доступно для успешного выполнения задач.</p> <p>3 Вызовы не доходят до устройства, либо последнее задание было завершено с ошибкой, связанной с прекращением связи, при этом имеется еще активное задание для устройства.</p>

Ниже перечислены информационные параметры, которые "видны" только в том случае, если на этапе конфигурирования было выбрано "No SNMP" для устройства, не поддерживающего SNMP, но поддерживающего вызовы (ping):

sysContact	Текст, введенный в поле sysContact на этапе конфигурирования; тип VT_BSTR, только чтение.
sysLocation	Текст, введенный в поле sysLocation на этапе конфигурирования; тип VT_BSTR, только чтение.
sysName	Текст, введенный в поле sysName на этапе конфигурирования; тип VT_BSTR, только чтение.

Примечание

SNMP:[OSM]&ipaddress()

Возвращается значение IP-адреса, сконфигурированное для узла с именем OSM.

Информация о SNMP OPC-сервере

OPC-сервер предоставляет переменные, с помощью которых можно получить информацию о SNMP OPC-сервере семейства SIMATIC NET.

Можно получить следующую информацию:

- Версия SNMP OPC-сервера SIMATIC NET
- Версия Winsocket
- Сведения о получении прерываний

Синтаксис

SNMP:[SYSTEM]&<informationparameter>()

Пояснения

SNMP

Протокол SNMP для обращения к информационным переменным для локальной системы.

SYSTEM

Идентификатор локальной системы (фиксированное значение).

<informationparameter>

Определены следующие информационные параметры:

version	Идентификатор версии SNMP OPC-сервера SIMATIC NET. Тип VT_BSTR, только чтение.
&winsockversion()	Идентификатор версии Winsocket. Тип VT_BSTR, только чтение.
&traplists()	<p>Указывает, смог ли SNMP OPC-сервер SIMATIC NET зарегистрироваться для получения прерываний. Тип VT_BOOL; только чтение.</p> <p>FALSE SNMP OPC-сервер не смог зарегистрироваться для получения прерываний.</p> <p>TRUE SNMP OPC-сервер успешно зарегистрировался для получения прерываний.</p>

Пример

SNMP:[SYSTEM]&version()

Возвращается версия SNMP OPC-сервера семейства SIMATIC NET, например, "SIMATIC NET Core Server SNMP V6.1.1000.2815 Copyright © SIEMENS AG".

4.8.5 Специальные прерывания протокола SNMP

Обработка SNMP-прерываний OPC-сервером

SNMP-прерывания – это события, которые формируются устройством и передаются OPC SNMP-серверу в произвольные моменты времени. OPC-сервер обрабатывает эти события следующим образом:

- Прерывание сопоставляется с простым событием интерфейса Alarms & Events.
- Для каждого сконфигурированного прерывания в интерфейсе Data Access создаются две переменные: значение одной переменной увеличивается на один всякий раз, когда поступает соответствующее прерывание, а в другой переменной хранится описание прерывания.

Синтаксис

Первая переменная предназначена для подсчета случаев возникновения события:

SNMP:[<devicename>]<trapname>

Вторая переменная предназначена для описания прерывания:

SNMP:[<devicename>]<trapname>_description

Пояснения

SNMP

Протокол SNMP для обращения к переменным процесса (MIB-объектам) и переменным SNMP-прерываний.

<Teilnehmername>

Имя устройства, указанное на этапе конфигурирования системы.

<trapname>

Имя прерывания.

Пример

Первая переменная возвращает количество событий (прерываний) "холодный запуск", инициируемых партнерским устройством:

SNMP:[OSM]coldStart

Вторая переменная возвращает описание прерывания:

SNMP:[OSM]coldStart_description

4.9 Ограничение доступа к OPC-переменным

Конфигурирование прав доступа в STEP 7 или NCM

Права доступа к переменным можно ограничить с помощью диалогового окна свойств OPC-сервера в STEP 7 или NCM. По умолчанию для переменных разрешено обращение для чтения/записи (RW), то есть, установлен соответствующий флажок, но в случае необходимости можно разрешить обращение только для чтения (R), записи (W) или запретить обращение вовсе (None).

В диалоговом окне "Item-specific Access Rights" (Права доступа к элементу) можно определить тип доступа отдельно для каждого элемента. Права, сконфигурированные в этом диалоговом окне, будут превалировать над правами доступа, принимаемыми по умолчанию.

Назначение прав доступа отдельному OPC-элементу

Установим права доступа для одного или нескольких OPC-элементов. Принимаемые по умолчанию права доступа, установленные для других OPC-элементов, не будут играть роли. Используется следующий синтаксис:

<OPCItem>=<rights>

<OPCItem>

Указывается один или несколько OPC-элементов в соответствии с синтаксисом, определяемым в документации по OPC. Возможно использование alias-имен (символьных имен). Можно использовать следующие метки:

- * любое количество символов
- ? ровно один символ

<rights>

- RW доступ для чтения и записи
- R доступ только для чтения
- W доступ только для записи
- NONE доступ для чтения и записи запрещен

Принцип выбора прав доступа

Установленные таким образом индивидуальные права доступа обладают приоритетом по отношению к правам доступа, принимаемым по умолчанию, которые были назначены для всей области. Индивидуальные права доступа, выбранные таким образом, применяются в том порядке, в котором они были заданы. Если для OPC-элемента было установлено несколько прав доступа, всегда применяется последнее введенное значение.

Пример:

```
DP:[CP 5611]Slave040_AB*=RW
DP:[CP 5611]Slave040_AB1=R
DP:[CP 5611]Slave040_AB2=W
DP:[CP 5611]Slave040_AB1=W
DP:[CP 5611]Slave040_AB1*=R
```

В результате действительными будут следующие сконфигурированные права доступа:

```
DP:[CP 5611]Slave040_AB2=W
DP:[CP 5611]Slave040_AB1=R
```


6 Использование OPC-сервера

В этой главе поясняются различные возможности, открывающиеся перед пользователем при работе с OPC-сервером.

6.1 SIMATIC Computing

Программный пакет SIMATIC Computing, являющийся частью программного обеспечения SIMATIC NET PC, позволяет создавать простые приложения, с помощью которых можно обращаться к данным процесса, не затрачивая значительных усилий на программирование. Чтение и запись данных процесса можно осуществлять через любую коммуникационную систему SIMATIC NET.

SIMATIC Computing предоставляет пользователю различные элементы ActiveX. Для решения конкретных задач эти ActiveX-элементы можно соответствующим образом конфигурировать и соединять друг с другом.

Совместимость с OPC

Элементы ActiveX из комплекта SIMATIC NET работают с OPC-интерфейсом напрямую.

Приложениями, которые были созданы с использованием элементов ActiveX из комплекта SIMATIC NET, также можно управлять с помощью OPC-серверов других производителей. Последнее, однако, может потребовать установку соответствующей лицензии.

Совместимость с предшествующими версиями

В предыдущих версиях SIMATIC NET и SIMATIC Computing для WinAC и WinLC был предусмотрен элемент управления данными (data control) "S7Data". Эта версия также включает новый, более удобный управляющий элемент SIMATIC NET OPC Data control.

Функции элемента S7Data по-прежнему поддерживаются, однако дальнейшее развитие элемента прекращено, и в этой документации он не описывается. В будущем следует использовать новый управляющий элемент SIMATIC NET OPC Data control.

Примечание

Подключая переменную процесса через элемент ActiveX, Вы устанавливаете связь с процессом. Изменение значения переменной процесса в элементе ActiveX может привести непосредственно к изменению физической величины в самом процессе.

Изменение физических величин процесса может привести к непредсказуемому развитию процесса, в результате чего может быть причинен ущерб здоровью персонала или повреждено оборудование.

С учетом этого все операции с ActiveX-элементами следует выполнять крайне осторожно. Например, следует ограничить права доступа к переменным процесса. На самом оборудовании должны быть предусмотрены устройства аварийного отключения.

6.1.1 Что такое элементы ActiveX Control?

SIMATIC Computing предоставляет пользователю различные элементы ActiveX control. Пользователь может работать с элементами визуализации (SIMATIC display control) или связывать другие элементы ActiveX (элементы других производителей или элементы, созданные самим пользователем) с элементом SIMATIC NET OPC Data control.

Элемент SIMATIC NET OPC Data control

Наиболее важным элементом является элемент управления данными SIMATIC NET OPC Data control. Без этого элемента никакие другие элементы управления SIMATIC не могут обращаться к данным процесса. Этот элемент управления данными обращается к данным процесса, которые поступают на OPC-сервер.

Управляющий элемент	Пиктограмма	Краткое описание
SIMATIC NET OPC Data control		Устанавливает соединение с OPC-сервером. Элемент управления данными скрыт от пользователя при работе программы.

Управляющие элементы для визуализации

Эти управляющие элементы (display control) предназначены для визуализации данных процесса. Данные на них поступают через управляющий элемент SIMATIC NET OPC Data control, а не непосредственно через OPC или другой интерфейс.

В SIMATIC предусмотрены следующие управляющие элементы визуализации:

Управляющий элемент	Внешний вид	Краткое описание
Кнопка		<p>Обращается к отдельным битам OPC-сервера. Элемент "Кнопка" обращается только к битовым данным и оперирует только двумя значениями:</p> <p>Off (ВЫКЛ) = 0</p> <p>On (ВКЛ) = 1</p> <p>Изменение состояния элемента "Кнопка" также приводит к изменению состояния переменной, назначенной элементу.</p> <p>Если для элемента "Кнопка" установлена защита от записи, он выполняет функцию индикатора.</p>
Число		<p>Обращается к переменным процесса. Пользователь может обращаться к байту, слову или двойному слову и изменять отдельные биты внутри этих данных.</p> <p>Если в элемент "Число" вводится новое значение, изменяется и значение соответствующей переменной процесса.</p>
Ползунок		<p>Обращается к переменным процесса. Пользователь может обращаться к байту, слову и двойному слову.</p> <p>Изменение положения "Ползунка" приводит к изменению значения переменной процесса.</p>

6.1.2 Как управляющие элементы обращаются к данным процесса?

Чтобы управляющие элементы SIMATIC ("Кнопка", "Число", "Ползунок") могли обратиться к данным процесса, должно быть выполнено следующее:

- Должно быть установлено соединение между элементом управления данными SIMATIC NET OPC Data control и OPC-сервером.
- Свойство переменной процесса OPC-сервера (значение, код качества, метка времени) должно быть связано со свойством управляющего элемента визуализации (например, **value (значение)**, **background color (цвет заднего фона)** и т.п.).

Элемент управления данными (Data control) конфигурирует OPC-сервер таким образом, чтобы постоянно осуществлялся мониторинг назначенных переменных. Если значение изменяется, новое значение поступает в управляющий элемент, откуда оно передается другим управляющим элементам.

На следующем рисунке показана взаимосвязь между управляющим элементом SIMATIC NET OPC Data control и другими управляющими элементами SIMATIC.

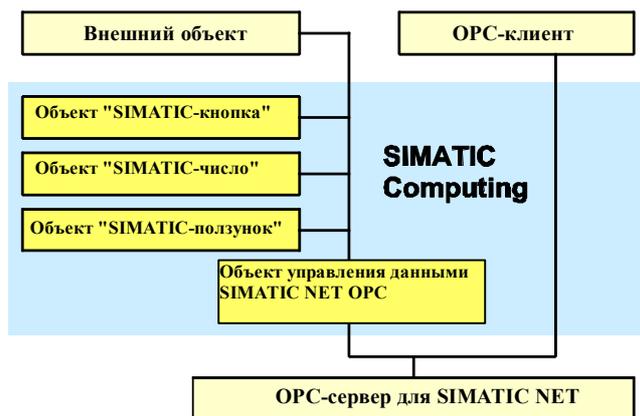


Рисунок 6–1 Связь между элементом управления данными SIMATIC NET OPC Data control и модулем для взаимодействия (интерфейса) с процессом

6.1.3 Как осуществляется доступ с помощью DCOM?

Управляющий элемент SIMATIC NET OPC Data control позволяет установить связь между программой и OPC-сервером другого компьютера.

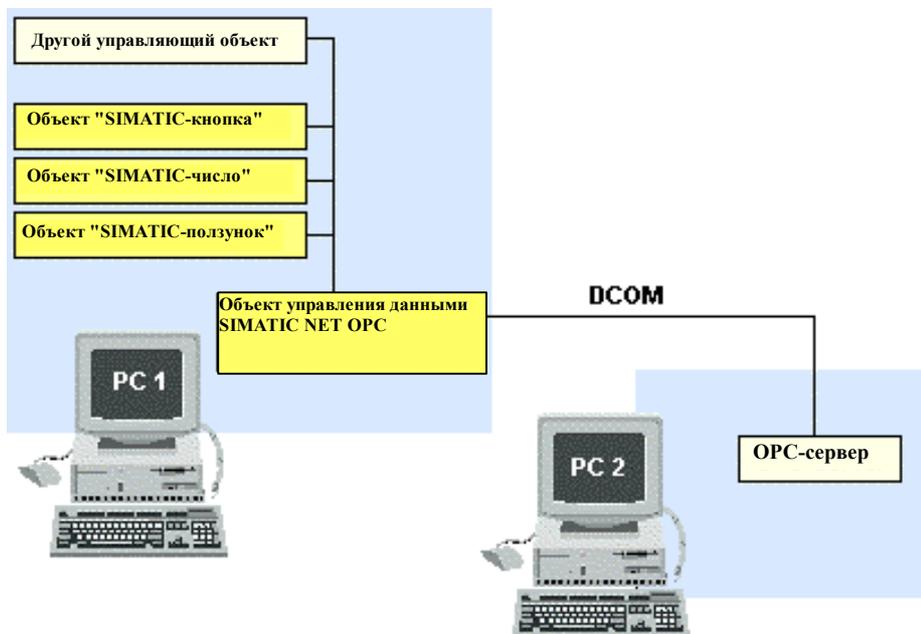


Рисунок 6–2 Связь с OPC-сервером через DCOM

6.1.4 Как используются управляющие элементы SIMATIC NET в инструментах разработки?

Чтобы управляющие элементы SIMATIC NET можно было использовать в средстве разработки, они должны быть ему известны.

Ниже описана последовательность действий для Visual Basic. Если используется другая среда разработки, некоторые шаги, возможно, придется изменить.

1. Запустите Visual Basic и создайте новый проект.
2. Откройте меню **Project (Проект)** и выберите команду **Components (Компоненты)**.

Откроется окно **Components (Компоненты)**. Во вкладке **Controls (Управляющие элементы)** отображаются все элементы ActiveX, имеющиеся на компьютере.

3. Выберите управляющие элементы SIMATIC NET.



Управляющие элементы SIMATIC NET

4. Щелкните по кнопке **ОК**.

Окно **Components (Компоненты)** будет закрыто, управляющие элементы SIMATIC NET будут вставлены в среду разработки.

6.1.5 Конфигурирование элемента управления данными SIMATIC NET OPC Data control

Элемент управления данными SIMATIC NET OPC Data control устанавливает соединение между ActiveX-элементами и OPC-сервером.

Элемент управления данными обладает свойствами, которые можно конфигурировать в приложениях-контейнерах (например, в Visual Basic).



Рисунок 6–3 Элемент управления данными

Как это работает?

Чтобы сконфигурировать элемент SIMATIC NET OPC Data control, необходимо выполнить следующие действия:

- Откройте окно **Properties (Свойства)** для элемента SIMATIC NET OPC Data control
- Выберите OPC-сервер
- После этого можно идти двумя путями:
 - Связать свойства (атрибуты) управляющих элементов визуализации со свойствами OPC-переменных процесса
 - Назначить OPC-переменные процесса событиям и запрограммировать обработку сообщений об изменениях состояний
- Дополнительные задачи

Создание программы

На этапе создания проекта управляющий элемент SIMATIC NET OPC Data control представляется в виде пиктограммы. Управляющий элемент связывается с OPC-сервером лишь с целью опроса пространства имен OPC, при условии, что OPC-сервер доступен локально или на удаленном компьютере и работает.

Управляющий элемент SIMATIC NET OPC Data control можно сконфигурировать в окне **Property Pages (Страницы свойств)**.

Выполнение программы

Когда созданная программа выполняется, управляющий элемент SIMATIC NET OPC Data control работает в фоновом режиме. Для пользователя он не "виден".

6.1.6 Как открыть окно свойств для элемента SIMATIC NET OPC Data control?

Конфигурирование элемента SIMATIC NET OPC Data control осуществляется в окне **Properties Pages (Страницы свойств)**, которое можно открыть в своем приложении-контейнере.

Ниже описана последовательность действий для Visual Basic. Если используется другая среда разработки, процедуру, возможно, потребуется скорректировать.

1. Щелкните по элементу управления данными (Data control) правой кнопкой мыши.
Откроется всплывающее меню.
2. Выберите меню **Properties (Свойства)**.
Откроется окно **Property Pages (Страницы свойств)**.

6.1.7 Выбор OPC-сервера

Являясь OPC-клиентом для интерфейса OPC Data Access 2.0, элемент SIMATIC NET OPC Data control может использовать любой OPC-сервер. OPC-сервер может быть локальным, либо располагаться на удаленном компьютере, с доступом через сеть.

1. Выберите закладку "OPC server" (OPC-сервер) в окне Property Pages (Страницы свойств).
2. В поле "Server" (Сервер) введите имя OPC-сервера, например, OPC.SimaticNET.
OPC-сервер также можно выбрать, нажав кнопку Browse (Обзор).
3. Если необходимо подключиться к OPC-серверу по сети с помощью DCOM, в поле Computer (Компьютер) следует ввести имя адресуемого компьютера.
В этом случае следует выполнить настройку DCOM на всех компьютерах.
4. Щелкните по кнопке **ОК**.
Выбранные параметры будут сохранены.

6.1.8 Подключение переменных процесса к элементам визуализации

Свойства (атрибуты) элементов визуализации можно связать со свойствами OPC-переменных процесса.

На этапе конфигурирования устанавливается связь между переменной процесса и элементом визуализации. На этапе выполнения программы содержание переменной визуализируется.

Как это работает?

Подключение к управляющим элементам визуализации происходит следующим образом:

Выберите свойство (атрибут) управляющего элемента, предназначенного для визуализации или ввода данных процесса

Можно также выполнить следующие дополнительные задачи:

- Удалить привязку переменной к свойству
- Настроить фильтр свойств ActiveX-элемента
- Сконфигурировать общие свойства

Выберите свойство управляющего элемента, предназначенного для отображения или ввода данных процесса

Чтобы данные процесса можно было отображать или вводить, необходимо связать свойство другого элемента ActiveX со свойством OPC-переменной через элемент SIMATIC NET OPC Data control.

Свойствами (атрибутами) OPC-переменной являются значение, код качества и метка времени.

Примеры

- Свойство **Text (Текст)** текстового поля в Visual Basic связывается со значением слова памяти 5.
После этого в текстовом поле отображается содержимое слова памяти 5.
- Свойство **BackColor (Цвет заднего фона)** элемента "SIMATIC Number" (Число) связывается с кодом качества слова памяти 5. Значениям кода качества GOOD (Хорошо), BAD (Плохо) и UNCERTAIN (Неопределенное состояние) назначаются, соответственно, зеленый, красный и желтый цвета.
Цвет заднего фона элемента "Число" изменяется, когда изменяется код качества переменной процесса, например, из-за обрыва провода.

Выбор свойства управляющего элемента визуализации

1. Выберите вкладку **Items (Элементы)** в окне **Property Pages (Страницы свойств)**.
2. Щелкните по кнопке **Assign New (Назначить новое)**.
Откроется окно **Assign Items (Назначить элементы)**.
3. В списке **Controls (Управляющие элементы)** выберите свойства требуемого управляющего элемента.
4. Выберите свойство, которое требуется связать с переменной процесса.
5. В верхнем поле, в котором отображается пространство имен сервера, выберите переменную процесса, которую следует связать с выбранным свойством.

В окне **Assigned Item (Назначенный элемент)** появится идентификатор (ItemID) выбранной переменной.

Значение ItemID также можно ввести непосредственно в самом поле.

6. Выберите свойства OPC-элементов, которые будут использоваться для визуализации.

Можно выбрать следующие свойства:

Свойство переменной процесса	Значение
Value (Значение)	С выбранным свойством элемента ActiveX будет связано значение переменной процесса.
Time stamp (Метка времени)	С выбранным свойством элемента ActiveX будет связана метка времени переменной процесса. Метка передается в формате строки. Формат отображения метки времени можно выбрать.
Quality (Код качества)	С выбранным свойством элемента ActiveX будет связан код качества переменной процесса. Можно выбрать, какое значение будет использоваться для кода качества. Можно, например, выбрать определенный цвет или текстовое сообщение.

7. Повторяйте действия 3 - 6, пока не будут назначены все требуемые свойства.
8. Щелкните по кнопке **OK**.
Окно **Assign Items (Назначить элементы)** будет закрыто. После этого значения выбранных свойств будут определяться переменными процесса.

6.1.9 Назначение события переменной процесса

С переменными процесса можно связывать события. Если в режиме выполнения переменная изменяется, управляющий элемент (Data control) формирует событие, параметр которого содержит имя события. Программа пользователя может реагировать на событие и, следовательно, на изменение переменной.

Назначение события переменной процесса

События, формируемые управляющим элементом (Data control), можно конфигурировать, назначая каждому событию определенную переменную процесса.

1. Выберите вкладку **Events (События)** в окне **Property Pages (Страницы свойств)**.
2. Выберите кнопку **Add... (Добавить...)**.
Откроется окно **Add Events (Добавить события)**.
3. Введите для каждого события имя в поле **User ID (Идентификатор пользователя)**.
4. В верхнем окне, в котором отображается пространство имен сервера, выберите переменную процесса, которую следует соединить с выбранным событием.
В поле **Item ID (Идентификатор элемента)** будет отображен идентификатор выбранной переменной.
Значение ItemID также можно ввести непосредственно в самом поле.
5. Щелкните по кнопке **Add... (Добавить...)**.
Переменная процесса будет назначена событию.
6. Повторяйте действия 4 - 7, пока не будут назначены все требуемые события.
7. Щелкните по кнопке **ОК**.
Окно **Add Events (Добавить события)** будет закрыто.

Примечание

Чтобы программа могла реагировать на события, сконфигурированные в соответствии с описанной выше последовательностью, необходимо применить метод ValueChanged (Изменение значения) управляющего элемента (Data control).

6.1.10 Дополнительные задачи

Для элемента Data control можно также выполнить следующие операции:

Удаление привязки переменной к свойству

Связь между значением переменной и свойством можно удалить одним из следующих способов:

- Удалить элемент ActiveX в приложении-контейнере.
- Удалить привязку в окне **Property Pages (Страницы свойств)**.

Если элемент ActiveX для визуализации удаляется из самого приложения-контейнера, соединение в элементе остается сконфигурированным. В следующий раз, когда элемент будет вставлен в приложение, он автоматически получит имя удаленного элемента. Сконфигурированные свойства соединения удаленного элемента также будут скорректированы соответствующим образом.

В окне **Property Pages (Страницы свойств)** привязку можно удалить полностью.

1. Выберите вкладку **Items (Элементы)** в окне **Property Pages (Страницы свойств)**.
2. Щелкните по кнопке **Assign New (Назначить новое)**.
Откроется окно **Assign Items (Назначить элементы)**.
3. Отобразите свойства элемента ActiveX, для которого требуется удалить назначение переменной.
4. Выберите свойство, назначение которого требуется удалить.
5. Удалите значение в **Assigned Item (Назначенный элемент)**.
Привязка переменной процесса к выбранному свойству будет удалена.
6. Щелкните по кнопке **ОК**.
Окно **Assign Items (Назначить элементы)** будет закрыто.

Примечание

Соединение может быть удалено при условии, что оно не используется в приложении-контейнере.

Настройка фильтра свойств элемента ActiveX

Управляющий элемент визуализации позволяет редактировать только выбранные свойства, например, **Enabled (Разрешено)** или **Value (Значение)**. Для этого необходимо настроить фильтр свойств.

1. Выберите закладку **Items (Элементы)** в окне **Property Pages (Страницы свойств)**.
2. Щелкните по кнопке **Assign New (Назначить новое)**.
Откроется окно **Assign Items (Назначить элементы)**.
3. Выберите кнопку **Filter... (Фильтр...)**.
Откроется окно **Property Filter (Фильтр свойств)**.
4. Выберите кнопку **Add... (Добавить...)**.
Откроется окно **Add (Добавить)**.
5. Укажите свойство, которое должно быть выбрано.
6. Щелкните по кнопке **ОК**.
Окно **Add (Добавить)** будет закрыто.
7. Выбранное свойство можно изменить в окне **Property Filter (Фильтр свойств)** с помощью кнопки **Edit (Изменить)**.
С помощью кнопки **Delete (Удалить)** выбранное свойство можно удалить из фильтра.
8. Щелкните по кнопке **ОК**.
Окно **Property Filter (Фильтр свойств)** будет закрыто.
9. Установите флажок **Filter (Фильтр)** в окне **Assign Items (Назначить элементы)**.
Теперь для всех элементов ActiveX будут отображаться только свойства, выбранные фильтром.
Если снять флажок **Filter (Фильтр)**, вновь будут отображаться все свойства.

Конфигурирование общих свойств

Можно сконфигурировать следующие свойства:

- **Autosconnect (автоматическое соединение)**
Если выбрано свойство **Autosconnect**, управляющий элемент автоматически соединяется с переменными OPC-сервера после конфигурирования.
Если свойство **Autosconnect** не выбрано, управляющий элемент соединяется с переменными только в результате выполнения команд в программе.
- **AutoConnect Timeout (Превышение времени автоматического соединения)**
Это свойство определяет, сколько миллисекунд будет ожидать управляющий элемент после соединения с OPC-сервером, прежде чем приступить к записи данных.

- **Default Dead Band (Зона нечувствительности по умолчанию)**
Это свойство определяет, какую величину должны иметь изменения, чтобы OPC-сервер записал новое значение в управляющий элемент (Data control).
 - **Default Update Rate (Скорость обновления по умолчанию)**
Это свойство определяет периодичность в миллисекундах, с которой OPC-сервер проверяет переменные процесса.
 - **Show Error Boxes (Показывать окна с сообщениями об ошибках)**
Если выбрано свойство Show Error Boxes, возникновение ошибок пользователя будет сопровождаться отображением стандартных сообщений об ошибках.
В SIMATIC Computing все сообщения об ошибках отображаются на английском языке. Если требуется, чтобы сообщения отображались на других языках, следует отключить опцию Show Error Boxes (Отображать окна с сообщениями об ошибках) и написать дополнительную программу для отображения сообщений об ошибках.
1. Выберите вкладку **Options (Опции)** в окне **Property Pages (Страницы свойств)**.
 2. Введите в полях требуемые значения.
 3. Щелкните по кнопке **ОК**.
Свойства сконфигурированы.

6.1.11 Конфигурирование элемента "Кнопка"

С помощью элемента "Кнопка" (Button) можно создавать кнопки для работы с битовыми переменными процесса (тип Boolean). В режиме выполнения состояние переменной отображается с помощью цвета и надписи на элементе "Кнопка". Щелчок по этой "кнопке" в режиме выполнения приводит к изменению значения битовой переменной процесса (тип Boolean).

Привязка кнопки к переменной процесса осуществляется путем назначения OPC-переменной свойству **Value (Значение)** элемента "Кнопка".

Элемент "Кнопка" может принимать два состояния:

- 0 или off (выкл)
- 1 или on (вкл)



Рисунок 6–4 Элемент "Кнопка"

Как это работает?

Чтобы сконфигурировать элемент "Кнопка", требуется выполнить следующие действия:

6.1.12 Свойства элемента "Кнопка"

Элемент "Кнопка" (Button) обладает следующими свойствами (атрибутами):

Свойство	Вкладка в окне Properties (Свойства) элемента "Siemens Button Control"	Краткое описание	Подробное описание
Enabled (Разрешено)	General (Общие свойства)	Указывает, передает ли элемент изменения своего состояния в параметр Value (Значение) и запускает ли он события	Активизация элемента "Кнопка"
FalseCaption (Надпись в состоянии "выкл")	General (Общие свойства)	Определяет надпись на элементе для случая, когда параметр Value (Значение) содержит значение "ложь", 0 или off (выкл)	Выбор надписи
FalseColor (Цвет в состоянии "выкл")	Color (Цвет)	Определяет цвет элемента для случая, когда параметр Value (Значение) содержит значение "ложь", 0 или off (выкл)	Выбор цвета элемента "Кнопка"
Font (Шрифт)	Font (Шрифт)	Определяет шрифт для отображения текста на элементе	Выбор шрифта надписи
TrueCaption (Текст в состоянии "вкл")	General (Общие свойства)	Определяет надпись на элементе для случая, когда параметр Value (Значение) содержит значение "верно", 1 или on (вкл)	Выбор надписи
TrueColor (Цвет в состоянии "вкл")	Color (Цвет)	Определяет цвет элемента для случая, когда параметр Value (Значение) содержит значение "верно", 1 или on (вкл)	Выбор цвета элемента "Кнопка"
Value (Значение)		Значение, отображаемое или вводимое с помощью элемента "Кнопка". Это свойство связывается с переменной процесса посредством элемента Data control.	Соединение элемента "Кнопка" с переменной процесса

6.1.13 Вызов окна свойств для элемента "Кнопка"

Окно **Property Pages (Страницы свойств)** открывается в приложении-контейнере.

Ниже показана последовательность действий для Visual Basic. Если используется другая среда разработки, некоторые действия, возможно, потребуются скорректировать.

1. Выберите элемент "Кнопка" (Button), который необходимо сконфигурировать.
2. Щелкните по элементу правой кнопкой мыши.
Будет открыто всплывающее меню.
3. Выберите команду **Properties (Свойства)**.
Будет открыто окно **Property Pages (Страницы свойств)**.

6.1.14 Выбор надписи

Для элемента "Кнопка" можно выбрать две различных надписи:

- Текст надписи на кнопке для случая, когда бит содержит значение **"верно", 1** или **on (вкл)**.
 - Текст надписи на кнопке для случая, когда бит содержит значение **"ложь", 0** или **off (выкл)**.
1. Выберите вкладку General (Общие свойства) в окне Property Pages (Страницы свойств).
 2. В поле **TrueCaption (Текст для состояния "верно")** введите текст для надписи, которая будет отображаться на кнопке, когда бит содержит значение **"верно", 1** или **on (вкл)**.
 3. В поле **FalseCaption (Текст для состояния "ложь")** введите текст для надписи, которая будет отображаться на кнопке, когда бит содержит значение **"ложь", 0** или **off (выкл)**.

6.1.15 Выбор шрифта надписи

Для элемента "Кнопка" можно настроить следующие параметры шрифта:

- Font (Шрифт)
 - Font size (Размер шрифта)
 - Effects (Эффекты)
1. Выберите вкладку **Font (Шрифт)** в окне **Property Pages (Страницы свойств)**.
 2. Выберите шрифт для надписи из списка **Font (Шрифт)**.
 3. Выберите размер шрифта в списке **Size (Размер)**. Требуемое значение также можно ввести в самом поле.
 4. Выберите требуемые эффекты в списке **Effects (Эффекты)**.
В поле **Sample Text (Образец надписи)** отображается образец надписи, соответствующей текущим настройкам.

6.1.16 Выбор цвета для элемента "Кнопка"

Для каждого из двух состояний элемента "Кнопка" можно выбрать соответствующий цвет, выбрав его из палитры цветов или непосредственно введя его значение.

1. Выберите вкладку **Color (Цвет)** в окне **Property Pages (Страницы свойств)**.
2. Если требуется сконфигурировать состояние **off (выкл)**, выберите в окне **Properties (Свойства)** параметр **FalseColor (Цвет в состоянии "ложь")**. Если требуется определить состояние **on (вкл)**, выберите **TrueColor (Цвет в состоянии "верно")**.
3. Выберите требуемую палитру цветов в списке **Color Set (Настройка цвета)**.
Цвета выбранной палитры цветов будут отображены в окне **Color Palette (Палитра цветов)**.
Нажав кнопку **Edit Custom Color... (Изменение цветов пользователя...)**, можно сконфигурировать собственные цвета.
4. Выберите требуемый цвет.

6.1.17 Активизация элемента "Кнопка"

По умолчанию элемент "Кнопка" активен сразу после его вставки в проект. Элемент "Кнопка" можно отключить или вновь включить в любое время с помощью окна **Siemens Button Control Properties (Свойства элемента Siemens Button)**:

1. Выберите вкладку **General (Общие свойства)** в окне **Siemens Button Control Properties (Свойства элемента Siemens Button)**.
Элемент "Кнопка" активизируется, когда устанавливается флажок **Enable (Разрешить)**.
Элемент "Кнопка" деактивизируется, когда флажок снимается.
2. В случае необходимости измените состояние флажка (активизируйте или деактивизируйте кнопку).

6.1.18 Назначение элемента "Кнопка" переменной процесса

Если предполагается визуализация или изменение значений переменной процесса с помощью элемента "Кнопка", необходимо привязать свойство **Value (Значение)** элемента "Кнопка" к переменной процесса в SIMATIC NET OPC Data control.

Свойство **Value (Значение)** элемента "Кнопка" может быть привязано только к биту. В случае применения переменных другого типа (не Boolean) значения переменных будут конвертироваться в тип Boolean OPC-сервером, если это возможно.

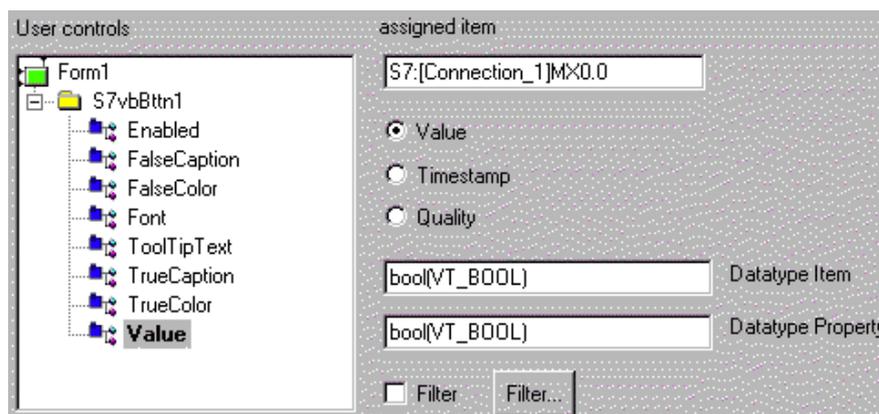


Рисунок 6–5 Подключение элемента "Кнопка" к переменной процесса

Примечание

После того, как переменная назначена элементу "Кнопка", этот элемент получает доступ к данным процесса. Если значения, визуализируемые элементом, изменяются, одновременно изменяется и значение физической величины процесса.

Изменение данных процесса может привести к непредсказуемой реакции процесса, которая может завершиться нанесением серьезного ущерба здоровью персонала или повреждением оборудования. Конструкция технологического оборудования всегда должна содержать элементы аварийной защиты и отключения.

6.1.19 Конфигурирование элемента "Число"

С помощью элемента "Число" можно отображать и изменять переменную процесса в числовом формате.



Рисунок 6–6 Элемент "Число"

В режиме выполнения значение переменной отображается в поле "Отображение". В этом поле можно ввести новое значение. В зависимости от конфигурации новое значение вступает в силу сразу же сразу же или передается OPC-серверу после щелчка по кнопке "Ввод".

Как это работает?

Чтобы сконфигурировать элемент "Число", необходимо выполнить следующие действия:

6.1.20 Свойства и методы элемента "Число"

Элемент "Число" (Number) обладает следующими свойствами (атрибутами) и методами:

Свойство или метод	Вкладка в окне свойств элемента "Siemens Number"	Краткое описание	Подробное описание
Alignment (Выравнивание)	General (Общие свойства)	Выбор способа выравнивания значений, отображаемых элементом "Число"	Настройка отображения данных
Appearance (Внешний вид)	Style (Стиль)	Выбор трех- или двухмерного отображения элемента	Выбор стиля
BackColor (Цвет заднего фона)	Color (Цвет)	Выбор цвета заднего фона	Выбор цвета для элемента "Число"
BorderStyle (Стиль границы)	Style (Стиль)	Выбор стиля границ элемента	Выбор стиля

Свойство или метод	Вкладка в окне свойств элемента "Siemens Number"	Краткое описание	Подробное описание
Caption <i>n</i> (Надпись <i>n</i>)	Label (Надпись)	Выбор текста первого (<i>n</i> =1) или второго (<i>n</i> =2) заголовка	Выбор надписи
Caption <i>n</i> Alignment (Выравнивание надписи <i>n</i>)	Label (Надпись)	Выбор типа выравнивания для первого или второго заголовка	Выбор надписи
Caption <i>n</i> BackColor (Цвет заднего фона надписи <i>n</i>)	Color (Цвет)	Выбор цвета заднего фона первого или второго заголовка	Выбор цвета для элемента "Число"
CaptionFont (Шрифт надписи)	Font (Шрифт)	Выбор шрифта для двух заголовков	Выбор шрифта для надписи
Caption <i>n</i> ForeColor (Цвет шрифта надписи <i>n</i>)	Color (Цвет)	Выбор цвета шрифта первого или второго заголовка	Выбор цвета для элемента "Число"
Caption <i>n</i> Size (Размер шрифта надписи <i>n</i>)	Label (Надпись)	Выбор размера шрифта первого или второго заголовка	Выбор надписи
ConvertedValue (Преобразованное значение)	read-only (только чтение)	Значение переменной процесса	
DataType (Тип данных)	General (Общие свойства)	Выбор типа данных для преобразованных значений	Настройка отображения данных
Enabled (Разрешено)	Style (Стиль)	Определяет, должен ли элемент передавать изменения в параметр Value (Значение) и формировать события	Активизация элемента "Число"

Свойство или метод	Вкладка в окне свойств элемента "Siemens Number"	Краткое описание	Подробное описание
Factor (Множитель)	Scaling (Масштабирование)	Выбор коэффициента масштабирования, используемого, если выбрана опция scale-by-formula (Масштабирование по формуле)	Настройка масштаба для отображения значений
Font (Шрифт)	Font (Шрифт)	Выбор шрифта для элемента	Выбор шрифта для надписи
ForeColor (Цвет шрифта)	Color (Цвет)	Выбор цвета шрифта для элемента	Выбор цвета для элемента "Число"
LayoutType (Расположение надписи)	Label (Надпись)	Выбор расположения отображаемой надписи	Выбор надписи
LimitCheck (Проверка граничного значения)	General (Общие свойства)	Указывает, должно ли значение находиться в пределах диапазона с заданными границами, и устанавливает область в пределах этого диапазона, в которой значения действительны	Настройка отображения данных
Locked(Защита от записи)	Style (Стиль)	Выбор режима "только чтение"	Активизация элемента "Число"
LowerLimit (Нижняя граница)	General (Общие свойства)	Указывает значение нижней границы для обращения к переменной	Настройка отображения данных

Свойство или метод	Вкладка в окне свойств элемента "Siemens Number"	Краткое описание	Подробное описание
Offset (Смещение)	Scaling (Масштабирование)	Определение смещения (Offset), которое должно применяться для опции scale-by-formula (Масштабирование по формуле)	Настройка масштаба для отображения значений
Precision (Точность)	General (Общие свойства)	Определяет степень точности вещественных чисел	Настройка отображения данных
Radix (Представление)	General (Общие свойства)	Определяет тип представления отображаемых чисел: двоичный, восьмеричный, десятичный или шестнадцатеричный	Настройка отображения данных
RangeRawMax (Верхняя граница входного диапазона)	Scaling (Масштабирование)	Установка максимального входного значения	Настройка масштаба для отображения значений
RangeRawMin (Нижняя граница входного диапазона)	Scaling (Масштабирование)	Установка минимального входного значения	Настройка масштаба для отображения значений
RangeScaledMax (Верхняя граница конечного диапазона)	Scaling (Масштабирование)	Определение максимального значения приведенного (масштабированного) числа	Настройка масштаба для отображения значений
RangeScaledMin (Нижняя граница конечного диапазона)	Scaling (Масштабирование)	Определение минимального значения приведенного (масштабированного) числа	Настройка масштаба для отображения значений
ScaleMode (Режим масштабирования)	Scaling (Масштабирование)	Выбор режима масштабирования значений	Настройка масштаба для отображения значений

Свойство или метод	Вкладка в окне свойств элемента "Siemens Number"	Краткое описание	Подробное описание
ShowButtons (Показывать кнопки)	General (Общие свойства)	Указывает, должны ли отображаться кнопки для ввода и удаления значений	Настройка отображения данных
ShowErrorBoxes (Показывать сообщения об ошибках)	General (Общие свойства)	Указывает, должны ли отображаться стандартные сообщения об ошибках в случае возникновения ошибок пользователя	Настройка отображения данных
UpperLimit (Верхняя граница)	General (Общие свойства)	Определяет верхнее граничное значение для обращения к переменной	Настройка отображения данных
Value (Значение)	Значение, отображаемое элементом "Число". Это свойство соединяется с переменной процесса посредством элемента Data control.		Привязка переменной процесса к элементу визуализации
Write (Записать)		Записывает ConvertedValue (Преобразованное значение) в Value (Значение)	
WriteMode (Режим записи)	Style (Стиль)	Указывает, должны ли новые значения записываться автоматически (т.е., не вручную)	Выбор стиля
ZeroPad (Заполнение нулями)	General (Общие свойства)	Указывает, должно ли число, отображаемое элементом, заполняться нулями спереди, до заполнения всех свободных разрядов (количество определяется типом данных)	Настройка отображения данных

6.1.21 Вызов окна свойств для элемента "Число"

Окно **Property Pages (Страницы свойств)** открывается в приложении-контейнере.

Ниже будет представлена последовательность действий для Visual Basic. Если используется другая среда разработки, некоторые действия, возможно, придется скорректировать.

1. Выберите элемент "Число" (Number), который требуется сконфигурировать.
2. Щелкните правой кнопкой мыши по элементу.
Будет открыто всплывающее меню.
3. Выберите команду **Properties (Свойства)**.
Откроется окно **Property Pages (Страницы свойств)**.

6.1.22 Настройка отображения данных

Здесь можно указать, как элемент "Число" будет отображать значения переменных процесса. Можно выбрать следующее:

- Размер и стиль отображения данных.
В следующей таблице перечислены размерности соответствующих типов данных для элемента "Число".

Тип данных	Степень	Длина	Описание
Boolean (Логический тип)	0	1 бит	Значение типа Boolean (битовая переменная)
Byte (Байт)	1	1 байт	Однобайтовое значение без знака
Word (Слово)	2	2 байта	Двухбайтовое значение без знака
Integer (Целое число)	3	2 байта	Двухбайтовое целое значение со знаком
Double word (Двойное слово)	4	4 байта	Четырехбайтовое значение без знака
Double integer (Двойное целое число)	5	4 байта	Четырехбайтовое значение со знаком
Real (Вещественное число)	6	4 байта	Четырехбайтовое целое значение со знаком

- Выбор представления значения: двоичное, восьмеричное или шестнадцатеричное представление.
- Количество десятичных разрядов после запятой.
- Отображаемые значения прижимаются к левой границе, прижимаются к правой границе или отображаются по центру элемента "Число".
- Граничные значения.
- Заполнение нулями спереди.

- Отображение кнопок для ввода значений.
- Отображение сообщений об ошибке.

Последовательность действий

1. Выберите вкладку **General (Общие свойства)** в окне **Property Pages (Страницы свойств)**.
2. Выберите тип данных в поле **Data type (Тип данных)**. Выберите тип данных, который соответствует типу данных переменной процесса.
3. В поле **Radix (Представление)** выберите тип представления значения: двоичное, восьмеричное, десятичное или шестнадцатеричное представление.
4. Если в поле **Data type (Тип данных)** выбрано **Real (Вещественное значение)**, введите в поле **Precision (Точность)** количество разрядов после запятой.
Если значение не вводится, по умолчанию устанавливается 3.
5. С помощью кнопок **Alignment (Выравнивание)** выберите тип выравнивания значения: по правой границе, по левой границе или по центру элемента "Число".
6. Если требуется отображать нули спереди значения, установите флажок **Zero Pad (Заполнение нулями)**.
По умолчанию флажок не установлен.
7. Если необходимо отображать кнопку для ввода и удаления значений, установите флажок **Show Buttons (Показывать кнопки)**.
Если выбрано отображение кнопок, в режиме выполнения потребуется подтвердить ввод каждого нового значения.
Щелчок по кнопке "Ввод" приводит к записи значения в OPC-сервер.
Щелчок по кнопке "Удалить" приводит к удалению нового значения и замене его предыдущим значением.
По умолчанию флажок не установлен.
Если выбрано отображение кнопок в режиме выполнения, также можно использовать следующие клавиши:

Кнопка	Клавиша
 Ввод	Клавиша "Ввод"
 Удалить	Клавиша "Escape"

8. Если необходимо, чтобы в режиме выполнения отображались стандартные сообщения об ошибках, установите флажок **Show Error Boxes (Отображать сообщения об ошибках)**.
По умолчанию флажок установлен.

6.1.23 Выбор стиля отображения

Для элемента "Число" можно выбрать стиль отображения.

- Двухмерное или трехмерное отображение
- Внешняя граница элемента
- Должно ли измененное значение свойства **Value** записываться в OPC-переменную автоматически или из программы, с помощью метода **Write (Записать)**

Последовательность действий

1. Выберите вкладку **Style (Стиль)** в окне **Property Pages (Страницы свойств)**.
2. В списке **Appearance (Стиль)** выберите двухмерное или трехмерное отображение элемента.
Трехмерное отображение можно выбрать только в том случае, если в списке **Border Style (Стиль границы)** выбрано значение **Fixed Single (Непрерывная одиночная граница)**.
3. В списке **Border Style (Стиль границ)** можно выбрать прямоугольную границу или отсутствие границы.
4. В списке **Write Mode (Режим записи)** выберите, должны ли значения элемента "Число" записываться в переменную процесса автоматически или программой.

6.1.24 Выбор надписи

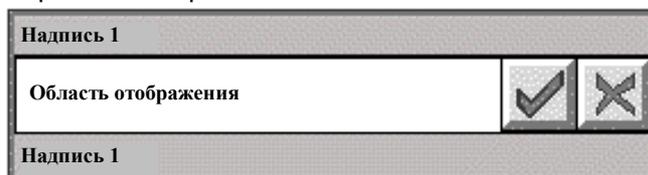
Для элемента "Число" можно выбрать надпись:

- Горизонтальное или вертикальное расположение надписей

Горизонтальное расположение



Вертикальное расположение



- Выравнивание по левой границе, по правой границе или по центру
- Ширина или высота надписи
- Текст надписи
- Цвет надписи
Цвет надписи выбирается во вкладке **Color (Цвет)**.
- Шрифт надписи
Шрифт надписи выбирается во вкладке **Font (Шрифт)**.

Последовательность действий

1. Выберите вкладку **Label (Надпись)** в окне **Property Pages (Страницы свойств)**.
2. В поле **Layout Type (Тип расположения)** выберите горизонтальное или вертикальное расположение надписей.
3. Если необходимо, можно указать ширину и высоту элемента "Число" в полях **Size 1 (Размер 1)** и **Size 2 (Размер 2)**.
Если выбрано горизонтальное расположение, необходимо указать ширину элемента "Число" с учетом ширины первой и второй надписей. Если выбрано вертикальное расположение, необходимо указать высоту элемента "Число" с учетом высоты первой и второй надписей.
По умолчанию установлено значение **1**. В этом случае размер надписи автоматически подстраивается под длину текста и размер шрифта, выбранный для надписи.
Если необходимо настроить несколько элементов "Число", имеет смысл указать такой размер, чтобы элементы "Число" выглядели одинаково.
В этом случае вводится размер надписей в единицах измерения "twip". Один "twip" составляет, приблизительно, 0.018 мм. 567 единиц "twip" соответствуют одному сантиметру экрана.
4. Введите тексты для первой и второй надписей в текстовых полях **Caption 1 (Надпись 1)** и **Caption 2 (Надпись 2)**.
5. С помощью кнопок выравнивания можно указать тип выравнивания первой и второй надписей: по левой границе, по правой границе или по центру.

6.1.25 Настройка масштаба для отображения значений

Для отображения переменных процесса в элементе "Число" можно выбрать масштаб. Функция масштабирования позволяет привести диапазон значений переменной процесса к диапазону, который используется для отображения. Коэффициент масштабирования используется в режиме выполнения и для чтения, и для записи значений.

Можно выбрать следующие варианты масштабирования:

- Без масштабирования
- Масштабирование по формуле
- Приведение диапазона

Последовательность действий

1. Выберите вкладку **Scaling (Масштабирование)** в окне **Property Pages (Страницы свойств)**.
2. Выберите вариант масштабирования в списке **Scale mode (Режим масштабирования)**.

По умолчанию установлено **No scaling (Без масштабирования)**.

3. Если в поле **Scale mode (Режим масштабирования)** выбрано **Scale by formula (Масштабирование по формуле)**, необходимо также ввести формулу.

Введите в поле **Scale (Масштаб)** коэффициент масштабирования.

В поле **Offset (Смещение)** введите константу, которая будет добавляться к значению после масштабирования.

Масштабированное значение рассчитывается по следующей формуле:

$(\text{Значение в ПЛК} \times \text{Масштаб}) + \text{Смещение} = \text{Отображение}$

$\text{Значение в ПЛК} = \text{значение переменной процесса}$

$\text{Масштаб} = \text{Коэффициент масштабирования}$

$\text{Смещение} = \text{Константа}$

$\text{Отображение} = \text{значение, отображаемое в элементе "Число"}$

4. Если в поле **Scale mode (Режим масштабирования)** выбрано **Scale by range transformation (Масштабирование путем приведения к диапазону)**, также необходимо указать верхнюю и нижнюю границы диапазонов для значения переменной процесса и значения, которое будет отображаться в элементе "Число".

6.1.26 Выбор шрифта надписи

Можно настроить следующие параметры шрифта для надписи на элементе "Число":

- Font (Шрифт)
- Font size (Размер шрифта)
- Effects (Эффекты)

Последовательность действий

1. Выберите вкладку **Font (Шрифт)** в окне **Property Pages (Страницы свойств)**.
2. Выберите шрифт для надписи в списке **Font (Шрифт)**.
3. Выберите размер шрифта в поле **Size (Размер)** или запишите значение непосредственно в поле.
4. Выберите требуемые эффекты в списке **Effects (Эффекты)**.
В поле **Sample Text (Пример текста)** отображается пример текста, соответствующий текущим параметрам.

6.1.27 Выбор цвета для элемента "Число"

Можно выбрать следующие цвета:

- Цвет заднего фона и цвет отображения значения
- Цвет заднего фона и цвет надписей первого и второго заголовков

Последовательность действий

Цвета можно выбрать из палитры цветов или определить собственные цвета:

1. Выберите вкладку **Color (Цвет)** в окне **Property Pages (Страницы свойств)**.
2. Выберите из списка элемент, для которого необходимо установить цвет.
3. Выберите в списке **Color Set (Набор цветов)** требуемую палитру цветов.
Цвета выбранной палитры отображаются в поле **Color Palette (Палитра цветов)**.
Если требуется задать собственные цвета, следует щелкнуть по кнопке **Edit Custom Color... (Настройка цветов пользователя...)**.
4. Выберите требуемый цвет.

6.1.28 Активизация элемента "Число"

По умолчанию элемент "Число" активизируется сразу же после конфигурирования и получает доступ к переменной процесса. Элемент "Число" можно деактивизировать или повторно активизировать в окне **Property Pages (Страницы свойств)** в любое время:

1. Выберите вкладку **Style (Стиль)** в окне **Property Pages (Страницы свойств)**.
Элемент "Число" активизирован, если установлен флажок **Enabled (Разрешено)**. В противном случае он деактивизирован.
2. В случае необходимости измените состояние флажка.
3. Если следует ограничить доступ к переменным процесса интерфейсного модуля, следует установить флажок **Locked (Только чтение)**.
По умолчанию эта опция не выбрана.

6.1.29 Назначение переменной процесса элементу "Число"

Если требуется отображать числовые значения процесса или вводить новые значения, необходимо связать свойство **Value (Значение)** элемента "Число" (Number) в элемент SIMATIC NET OPC Data control с переменной процесса.

Кроме того, можно связать свойство **BackColor (Цвет заднего фона)** с кодом качества переменной процесса. В этом случае цвет заднего фона элемента "Число" будет сигнализировать действительность или недействительность отображаемого значения.

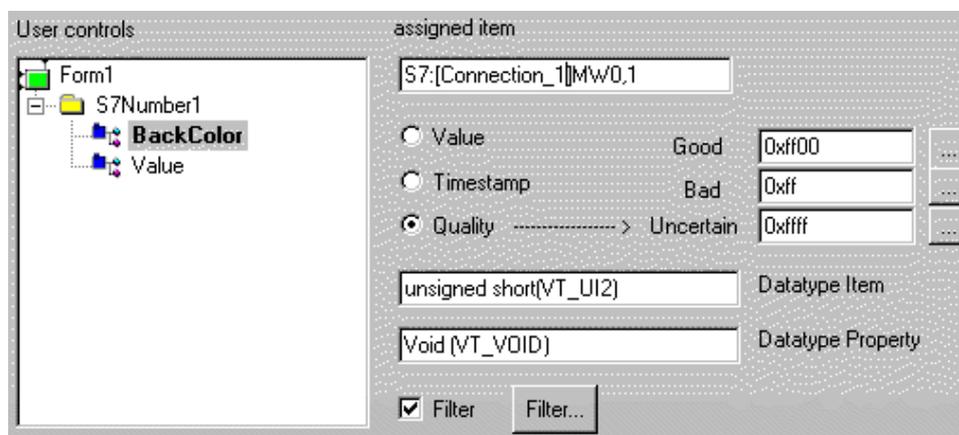


Рисунок 6–7 Назначение переменной процесса элементу "Число"

Примечание

После того, как переменная назначена элементу "Число", этот элемент получает доступ к данным процесса. Если значения, визуализируемые элементом, изменяются, одновременно изменяются и значения физических величин процесса.

Изменение величин процесса может привести к непредсказуемой реакции процесса, нанесением серьезного ущерба здоровью персонала или повреждению оборудования. Конструкция технологического оборудования всегда должна содержать элементы аварийной защиты и отключения.

6.1.30 Конфигурирование элемента "Ползунок"

Переменные процесса можно отображать и изменять с помощью элемента "Ползунок" (Slider). Перемещение ручки ползунка приводит к изменению значения переменной процесса. И наоборот, изменение соответствующей переменной процесса приводит к автоматическому изменению положения ручки ползунка.

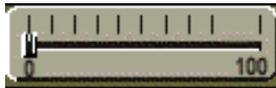


Рисунок 6–8 Элемент "Ползунок"

Как это работает?

Чтобы сконфигурировать элемент "Ползунок", требуется выполнить следующие действия:

6.1.31 Свойства элемента "Ползунок"

Элемент "Ползунок" (Slider) обладает следующими свойствами (атрибутами):

Свойство	Вкладка в окне свойств элемента "Siemens Slider"	Краткое описание	Подробное описание
Direction (Положение)	General (Общие свойства)	Выбор положения элемента: горизонтальное или вертикальное	Настройка отображения данных

Свойство	Вкладка в окне свойств элемента "Siemens Slider"	Краткое описание	Подробное описание
Enabled (Разрешено)	General (Общие свойства)	Указывает, должен ли элемент реагировать на изменения свойств значения	Активизация элемента "Ползунок"
LargeChange (Шаг грубой настройки)	General (Общие свойства)	Выбор размера шага, на который смещается ползунок активированного элемента, когда происходит нажатие клавиши "PageUp" или "PageDown"	Настройка отображения данных
Max (Максимальное значение)	General (Общие свойства)	Выбор значения, соответствующего верхнему положению ползунка	Настройка отображения данных
Min (Минимальное значение)	General (Общие свойства)	Выбор значения, соответствующего нижнему положению ползунка	Настройка отображения данных
SmallChange (Шаг точной настройки)	General (Общие свойства)	Выбор размера шага, на который смещается ползунок активированного элемента, когда нажимается клавиша перемещения курсора (up/down или right/left)	Настройка отображения данных
Ticks (Метки шкалы)	General (Общие свойства)	Выбор количества штрихов (меток шкалы)	Настройка отображения данных
Value (Значение)	Значение, устанавливаемое или отображаемое элементом "Ползунок". Это свойство связывается с переменной процесса через элемент Data control.		Назначение переменной процессу элементу "Ползунок"

6.1.32 Вызов окна свойств для элемента "Ползунок"

Окно **Property Pages (Страницы свойств)** открывается в приложении-контейнере.

Ниже будет представлена последовательность действий для Visual Basic. Если используется другая среда разработки, некоторые действия, возможно, придется скорректировать.

Последовательность действий

1. Выберите элемент "Ползунок" (Slider), который требуется сконфигурировать.
2. Щелкните правой кнопкой мыши по элементу.
Будет открыто всплывающее меню.
3. Выберите команду **Properties (Свойства)**.
Откроется окно **Property Pages (Страницы свойств)**.

6.1.33 Настройка отображения данных

Здесь можно указать, как элемент "Ползунок" будет отображать значения переменных процесса.

- Горизонтальное или вертикальное расположение элемента



Горизонтальное расположение



Вертикальное расположение

- Количество меток шкалы между минимальным и максимальным значениями
- Минимальное и максимальное отображаемые значения
- Шаг смещения ползунка при нажатии клавиш управления курсором
- Должны ли отображаться минимальное и максимальное значения

Последовательность действий

1. Выберите окно **Property Pages (Страницы свойств)**.
2. В полях **Direction (Положение)** выберите, как должен располагаться элемент "Ползунок": горизонтально или вертикально.
3. Если требуется, чтобы элемент "Ползунок" отображал минимальное и максимальное значения, установите флажок **Show Min and Max Value (Показывать минимальное и максимальное значения)**.
4. В поле **Ticks (Метки шкалы)** укажите количество меток, которое должно отображаться между минимальным и максимальным значениями. В режиме выполнения промежуточные значения отображаются в виде коротких меток на шкале. Ползунок может перемещаться только по этим меткам.
5. В полях **Min (Минимальное значение)** и **Max (Максимальное значение)** введите минимальное и максимальное значения.
6. В полях **SmallChange (Шаг точной настройки)** и **LargeChange (Шаг грубой настройки)** введите размер шага, на который будет перемещаться ползунок в режиме выполнения при нажатии клавиш управления курсором или клавиш PageUp или PageDown.

6.1.34 Активизация элемента "Ползунок"

По умолчанию элемент "Ползунок" активизируется сразу же после конфигурирования. Элемент "Ползунок" можно деактивизировать или повторно активизировать в окне **Property Pages (Страницы свойств)** в любое время:

1. Откройте окно **Property Pages (Страницы свойств)**.
Элемент "Ползунок" активизирован, если установлен флажок **Enabled (Разрешено)**. В противном случае он деактивизирован.
2. В случае необходимости установите или сбросьте флажок.

6.1.35 Назначение переменной процесса элементу "Ползунок"

Если требуется отображать или изменять значение переменной процесса с помощью элемента "Ползунок" (Slider), следует связать свойство **Value (Значение)** элемента "Ползунок" с переменной процесса в элементе SIMATIC NET OPC Data control.

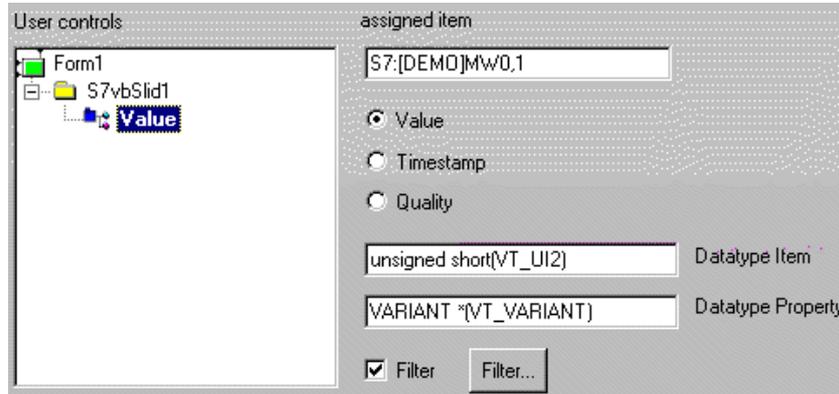


Рисунок 6–9 Назначение переменной процесса элементу "Ползунок"

Примечание

После того, как переменная назначена элементу "Ползунок", этот элемент получает доступ к данным процесса. Если значения, визуализируемые элементом, изменяются, одновременно изменяется и значение физической величины процесса.

Изменение величин процесса может привести к непредсказуемой реакции процесса и нанесению серьезного ущерба здоровью персонала или повреждению оборудования. Конструкция технологического оборудования всегда должна содержать элементы аварийной защиты и отключения.

6.1.36 Использование других элементов ActiveX

Управляющий элемент SIMATIC NET OPC Data control также можно использовать с элементами других производителей или элементами ActiveX собственной разработки.

Собственные элементы ActiveX и элементы других производителей должны предоставлять ряд методов, которые могли бы реагировать на изменение назначенных им переменных.

В качестве приложения-контейнера можно использовать, к примеру, Visual Basic. Если используется другое контейнерное приложение, оно должно соответствовать требованиям, предъявляемым к расширенным управляющим элементам (Extended Controls). В противном случае необходимо будет создать программу, которая бы выполняла эти функции.

6.1.37 Правила работы с контейнерными приложениями?

Управляющие элементы с расширенными функциями (Extended Controls)

Чтобы элемент SIMATIC NET OPC Data control мог работать с приложением-контейнером, приложение должно поддерживать функцию Property Browsing (Обзор свойств). Кроме того, приложение-контейнер должно поддерживать функции расширенных управляющих элементов, что соблюдается в случае использования приложений-контейнеров производства Microsoft.

Расширенный управляющий элемент – это подэлемент, в котором содержится другой управляющий элемент, поддерживающий специальные свойства, методы и события контейнера. Более подробные сведения можно найти в интерактивной справочной системе Microsoft.

Свойства и методы для расширенных элементов управления

Чтобы использовать функции расширенных управляющих элементов, приложение-контейнер должно поддерживать следующие методы:

- IOleClientSize::GetContainer
- IOleContainer::EnumObjects
- IOleControlSite::GetExtendedControl

Приложение должно распознавать свойство **Name (Имя)**.

Отсутствие расширенных управляющих элементов

Visual Basic поддерживает расширенные управляющие элементы. Однако приложения-контейнеры других производителей могут не поддерживать расширенные управляющие элементы (например, Borland's Delphi V3.0).

Если приложение-контейнер пользователя не поддерживает расширенные управляющие элементы, необходимо написать программу, которая выполняла бы отсутствующие функции. Центр технической поддержки фирмы Siemens будет рад предложить вам специальные приложения-контейнеры. Центр поддержки предоставит вам примеры программ, содержащие дополнительные функции.

6.1.38 Правила работы с элементами ActiveX

Можно создавать собственные элементы ActiveX и использовать их с элементами Data control из семейства SIMATIC. Созданные управляющие элементы должны предоставлять одно свойство, которое соответствовало бы свойству **Value (Значение)** элементов SIMATIC.

6.1.39 Каким образом элемент Data Control читает данные?

Если приложение-контейнер поддерживает расширенные управляющие элементы, элемент управления данными (Data control) обнаружит новый созданный элемент ActiveX и его свойства автоматически.

Если это не так, следует назначить переменную процесса свойству, которое соответствует свойству **Value (Значение)** (так же, как это делалось во вкладке **Items (Элементы)** окна **Property Pages (Страницы свойств)**). Если значение переменной в модуле связи с процессом изменяется, элемент Data control автоматически изменяет значение свойства используемого управляющего элемента.

Управляющий элемент, разработанный вами, должен содержать подпрограмму для обработки данных элемента управления данными (Data control).

6.1.40 Как данные записываются в элемент Data Control?

Если вы создали собственный элемент ActiveX, вам потребуется подпрограмма для обработки данных.

Ниже представлен пример подпрограммы для чтения значения свойства и внесения в него изменений.

```
Программа на Visual Basic
-----
Public Property Get Value() As Long
    Value = Object1.Value
End Property
-----
Public Property Let Value(ByVal New_Value As Long)
    Object1.Value() = New_Value
    PropertyChanged "Value"
End Property
-----
Private Sub Value_Change()
    PropertyChanged "Value"
End Sub
```

6.1.41 Свойства и методы управляющих элементов SIMATIC

Если вам требуется получить более подробную информацию о свойствах и методах управляющих элементов (Control) семейства SIMATIC, следует обратиться к настоящему разделу.

Управляющие элементы семейства SIMATIC обладают следующими свойствами и методами:

Свойство или метод	Выполняемая функция
Activated (Активизация)	Активизация всех сконфигурированных соединений
AddNotification (Добавить уведомление)	Добавление события в таблицу событий элемента OPC Data control. Когда переменная, указанная в параметрах вызова, изменяется, вызывается функция
Alignment (Выравнивание)	Выбор способа выравнивания (расположения) параметров
Appearance (Стиль отображения)	Определение внешнего вида управляющего элемента
AutoConnect (Автоматическое соединение)	Определяет, устанавливаются ли сконфигурированные соединения в режиме выполнения автоматически
AutoConnectTimeout (Превышение времени при автоматическом соединении)	Выбор значения времени превышения
BackColor (Цвет заднего фона)	Определение цвета заднего фона элемента
BorderStyle (Стиль границы)	Определяет, должен ли элемент отображаться с прямоугольной границей
Caption n (Надпись n)	Определение текста первого или второго заголовка
Caption n Alignment (Выравнивание надписи n)	Определение типа выравнивания первого и второго заголовков
Caption n BackColor (Цвет заднего фона надписи n)	Выбор цвета заднего фона для заголовков
Caption n Font (Шрифт надписи n)	Выбор шрифта для заголовков
Caption n ForeColor (Цвет надписи n)	Выбор цвета надписей для заголовков
Caption n Size (Размер надписи n)	Выбор ширины заголовков
Connect (Соединить)	Установление всех сконфигурированных соединений
ConvertedValue (Преобразованное значение)	Установка значения свойства Value (Значение) , преобразованного в соответствии с формулой масштабирования

Свойство или метод	Выполняемая функция
DataType (Тип данных)	Выбор типа данных для преобразованных значений
DefaultDeadband (Зона нечувствительности по умолчанию)	Определение зоны нечувствительности, которая используется OPC-сервером
DefaultUpdateRate (Скорость обновления по умолчанию)	Определение скорости обновления, которую использует элемент SIMATIC Data control, если скорость обновления не задана
Direction (Положение)	Выбор положения (горизонтальное или вертикальное)
Disconnect (Отсоединить)	Разрыв всех установленных соединений
Enabled (Разрешено)	Определяет, реагирует ли управляющий элемент (control) на изменение свойства Value (Значение)
Factor (Коэффициент)	Выбор коэффициента масштабирования, используемого для опции scale-by-formula (Масштабирование по формуле)
FalseCaption (Надпись в состоянии "ложь")	Выбор текста, отображаемого на элементе, когда свойство Value (Значение) содержит значение False ("Ложь")
FalseColor (Цвет в состоянии "ложь")	Выбор цвета элемента для случая, когда свойство Value (Значение) содержит значение False ("Ложь")
Font (Шрифт)	Выбор шрифта
ForeColor (Цвет надписи)	Выбор цвета текста
GetItemProperties (Прочитать свойства элемента)	Возвращает значения свойств OPC-элементов, указанных в вызове метода
LargeChange (Шаг грубой настройки)	Выбор величины шага, на который смещается "ручка" выбранного (находящегося в "фокусе") ползунка, когда нажимается клавиша "PageUp" или "PageDown"
LayoutType (Расположение)	Выбор способа расположения заголовков на элементе относительно области отображения
LimitCheck (Проверка предельных значений)	Выбор проверки попадания введенного значения в диапазон
Locked (Только чтение)	Указывает, что к элементу разрешен доступ только для чтения
LowerLimit (Нижняя граница)	Выбор нижней границы значения, записываемого в OPC-сервер и возвращаемого OPC-сервером

Свойство или метод	Выполняемая функция
Max (Максимальное значение)	Определение максимального значения для ползунка
Min (Минимальное значение)	Определение минимального значения для ползунка
NodeName (Имя узла)	Указывает идентификатор удаленного компьютера в сети (для установки соединения через DCOM).
Offset (Смещение)	Определение смещения для опции scale-by-formula (Масштабирование по формуле)
Precision (Точность)	Выбор степени точности числа
QueryAvailableItemProperties (Опросить имеющиеся свойства элемента)	Возвращает список всех OPC-свойств, включая описание свойств OPC-элемента.
Radix (Представление)	Выбор способа представления отображаемого числа (двоичное, восьмеричное, десятичное или шестнадцатеричное представление)
RangeRawMax (Максимальное входное значение)	Определение верхней границы входного диапазона
RangeRawMin (Минимальное входное значение)	Определение нижней границы входного диапазона
RangeScaledMax (Максимальное значение приведенного диапазона)	Определение верхней границы диапазона преобразованных значений
RangeScaledMin (Минимальное значение приведенного диапазона)	Определение нижней границы диапазона преобразованных значений
ReadMultiVariables (Читать несколько переменных)	Чтение значений указанных OPC-элементов из партнерского устройства
ReadVariable (Читать переменную)	Чтение значения переменной OPC-сервера из партнерского устройства
ScaleMode (Режим масштабирования)	Выбор режима масштабирования значений
ServerName (Имя сервера)	Выбор OPC-сервера для связи с процессом
ShowButtons (Показывать кнопки)	Указывает, должны ли на элементе отображаться кнопки для ввода и удаления значений
ShowErrorBoxes (Отображать сообщения об ошибках)	Указывает, должны ли в случае возникновения ошибок пользователя отображаться сообщения об ошибках
SmallChange (Шаг точной настройки)	Выбор величины шага, на который смещается "ручка" выбранного (находящегося в "фокусе") ползунка, когда нажимается клавиша управления курсором (вверх/вниз или вправо/влево)

Свойство или метод	Выполняемая функция
Ticks (Метки шкалы)	Выбор количества меток на шкале элемента
TrueCaption (Надпись в состоянии "верно")	Выбор текста, отображаемого на элементе, когда свойство Value (Значение) содержит значение True ("Верно")
TrueColor (Цвет в состоянии "верно")	Выбор цвета элемента для случая, когда свойство Value (Значение) содержит значение True ("Верно")
UpperLimit (Верхняя граница)	Определяет верхнюю границу значения, записываемого в OPC-сервер
Value (Значение)	Соединяет элемент SIMATIC NET OPC Data control с OPC-сервером
Write (Запись)	Записывает значение свойства ConvertedValue (Преобразованное значение) в свойство Value (Значение)
WriteMode (Режим записи)	Выбор типа срабатывания элемента, когда пользователь вводит новое значение
WriteMultiVariables (Запись нескольких переменных)	Записывает новые значения в несколько переменных OPC-сервера
WriteVariable (Запись переменной)	Записывает новое значение в указанную переменную OPC-сервера
ZeroPad (Заполнение нулями)	Определяет, должно ли число, отображаемое элементом, заполняться спереди нулями, количество которых зависит от типа данных

6.1.42 Activated (Активизация)

SIMATIC NET OPC Data control

Активизирует или деактивизирует все взаимосвязи с переменными процесса. Если элемент SIMATIC NET OPC Data control деактивизируется, также прекращается обмен данными между элементами визуализации и элементом SIMATIC OPC Data Control.

Синтаксис

`object.Activated [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

value (значение)

Логическое выражение, которое определяет, будет ли реагировать элемент на события, формируемые пользователем.

Можно выбрать одно из следующих значений:

True (Верно)	Все соединения активизированы (по умолчанию).
False (Ложь)	Все соединения деактивизированы.

Если значение параметра **Activated (Активизация)** не указывается, все соединения с переменными процесса будут активны.

Примечание

Соединения с переменными процесса остаются установленными даже в случае их деактивизации.

6.1.43 Alignment (Выравнивание)

■ SIMATIC Number Control (элемент "Число")

Определяет способ выравнивания числа, отображаемого элементом "Число".

Синтаксис

object.Alignment [= значение]

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число").

value (значение)

Значение или константа, которая определяет тип выравнивания.

Можно выбрать одно из следующих значений:

0 (по левой границе)	Число выравнивается по левой границе поля (по умолчанию).
1 (по правой границе)	Число выравнивается по правой границе поля.
2 (по центру)	Число выравнивается по центру поля.

Если параметр **Alignment (Выравнивание)** не указывается, число выравнивается по левой границе поля.

6.1.44 Appearance (Стиль отображения)

■ SIMATIC Number Control (элемент "Число")

Определяет, должен ли элемент "Число" отображаться в 3D-режиме.

Синтаксис

`object.Appearance [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, которая определяет внешний вид элемента.

Возможны следующие значения:

0 (двухмерное (плоское) изображение)	Элемент отображается без 3D-эффекта.
1 (трехмерное (3D) изображение)	Элемент отображается с 3D-эффектом (по умолчанию).

Примечание

Настройка типа изображения играет роль только, если для BorderStyle (Стиль границы) выбрано FixedSingle (Непрерывная одиночная граница).

6.1.45 AutoConnect (Автоматическое соединение)

■ SIMATIC NET OPC Data control

Указывает, будут ли сконфигурированные соединения с OPC-сервером устанавливаться в режиме выполнения автоматически.

Синтаксис

`object.AutoConnect [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

value (значение)

Логическое выражение, которое определяет, будет ли элемент реагировать на события, формируемые пользователем.

Возможны следующие значения:

True (Верно)	Все сконфигурированные соединения устанавливаются в режиме выполнения автоматически (по умолчанию)
False (Ложь)	Все соединения устанавливаются только после вызова метода Connect (Соединить) .

Если параметр **AutoConnect** не указывается, все соединения устанавливаются в режиме выполнения автоматически.

6.1.46 **AutoConnectTimeout (Превышение времени при автоматическом соединении)**

■ SIMATIC NET OPC Data control

Определяет значение времени превышения.

Если для свойства **AutoConnect (Автоматическое соединение)** выбрано значение **True (Верно)**, элемент SIMATIC NET OPC Data control вызывает метод **Connect (Соединить)** по истечении указанного времени.

Синтаксис

```
object.AutoConnectTimeout [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

value (значение)

Значение времени превышения (тип *long*).

6.1.47 BackColor (Цвет заднего фона)

■ SIMATIC Number Control (элемент "Число")

Определяет цвет заднего фона элемента "Число".

Синтаксис

`object.BackColor [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число").

value (значение)

Значение или константа, которая определяет цвет заднего фона элемента.

Можно выбрать следующее:

Стандартные цвета	Выбор цветов с помощью палитры RGB-цветов.
Цвета системы Windows	Выбор цветов с помощью системных цветов. Системные цвета зависят от контейнера.

6.1.48 BorderStyle (Стиль границы)

■ SIMATIC Number Control (элемент "Число")

Определяет, должен ли элемент "Число" отображаться с границей.

Синтаксис

`object.BorderStyle [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, которая определяет, должна ли использоваться граница.

Можно выбрать одно из следующих значений:

0 или None (Нет)	Нет границы.
1 или FixedSingle (Непрерывная одиночная граница)	Граница в виде одиночной линии.

Если параметр **BorderStyle (Стиль границы)** не указывается, граница у элемента будет отсутствовать.

6.1.49 Caption n (Надпись n)

 SIMATIC Number Control (элемент "Число")

Определяет текст заголовка.

Синтаксис

Первый заголовок:

`object.Caption1 [= значение]`

Второй заголовок:

`object.Caption2 [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Строка с текстом заголовка.

6.1.50 Caption n Alignment (Выравнивание надписи n)

 SIMATIC Number Control (элемент "Число")

Определяет способ выравнивания заголовков.

Синтаксис

Первый заголовок:

`object.Caption1Alignment` [= значение]

Второй заголовок:

`object.Caption2Alignment` [= значение]

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, которая определяет способ выравнивания заголовков.

Можно выбрать следующие значения:

0 или Left (по левой границе)	Заголовок выравнивается по левой границе (по умолчанию).
1 или Right (по правой границе)	Заголовок выравнивается по правой границе (по умолчанию).
2 или Center (по центру)	Заголовок центрируется.

6.1.51 CaptionnBackColor (Цвет заднего фона надписи n)

 SIMATIC Number Control (элемент "Число")

Определяет цвет заднего фона для заголовков.

Синтаксис

Первый заголовок:

`object.Caption1BackColor` [= значение]

Второй заголовок:

`object.Caption2BackColor` [= значение]

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, которая определяет цвет заднего фона заголовка.

Можно выбрать одно из следующих значений:

Стандартные цвета	Выбор цветов с помощью палитры RGB-цветов.
Цвета системы Windows	Выбор цветов с помощью системных цветов. Системные цвета зависят от контейнера.

6.1.52 **Caption n Font (Шрифт надписи n)**

■ SIMATIC Number Control (элемент "Число")

Определяет шрифт для первого или второго заголовка.

Синтаксис

Первый заголовок:

`object.Caption1Font` [= значение]

Второй заголовок:

`object.Caption2Font` [= значение]

Пояснения

`object` (объект)

Описывает элемент в списке назначений и идентифицирует определенный управляющий элемент SIMATIC.

`value` (значение)

Значение или константа, определяющая шрифт.

Следует знать возможности, предоставляемые используемой средой разработки для выбора шрифта.

6.1.53 **Caption n ForeColor (Цвет надписи n)**

■ SIMATIC Number Control (элемент "Число")

Определяет цвет текста заголовков.

Синтаксис

Первый заголовок:

`object.Caption1ForeColor` [= значение]

Второй заголовок:

`object.Caption2ForeColor` [= значение]

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, которая определяет цвет текста заголовка.

Возможны следующие значения:

Стандартные цвета	Выбор цветов с помощью палитры RGB-цветов.
Цвета системы Windows	Выбор цветов с помощью системных цветов. Системные цвета зависят от контейнера.

6.1.54 CaptionnSize (Размер текста n)

➡ SIMATIC Number Control (элемент "Число")

В случае горизонтального расположения определяется ширина, а в случае вертикального расположения – высота заголовка. Высота и ширина указываются в единицах размерности "twip" (см. описание в предыдущих разделах).

Синтаксис

Первый заголовок:

`object.Caption1Size` [= значение]

Второй заголовок:

`object.Caption2Size` [= значение]

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, которая определяет размер шрифта надписи.

Возможные значения:

-1	Автоматический размер (по умолчанию)
>=1	Значение в единицах "twip"

Если параметр **CaptionnSize (Размер надписи n)** не указывается, по умолчанию используется значение **-1**. В этом случае размер поля определяется в зависимости от длины текста.

6.1.55 ConvertedValue (Преобразованное значение)

■ SIMATIC Number Control (элемент "Число")

Записывает приведенное (масштабированное) значение в свойство **Value (Значение)**. **ConvertedValue (Преобразованное значение)** уже приведено к типу данных, указанному в свойстве **DataType (Тип данных)**.

Синтаксис

```
object.ConvertedValue [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение типа variant, определяющее значение элемента.

Примечание

Если в параметр Enabled (Разрешено) записано значение False (Ложь), двунаправленная связь между параметром Value (Значение) и Converted Value (Преобразованное значение) разрывается. Изменения, вносимые в свойство Converted Value (Преобразованное значение), не влияют на свойство Value (Значение).

6.1.56 DataType (Тип данных)

■ SIMATIC Number Control (элемент "Число")

Определяет тип данных для преобразованных значений.

Если число с используемым типом данных слишком велико и не может быть отображено, значение обрезается. Тип данных, выбранный для элемента "Число", должен соответствовать типу данных, выбранному в поле **Assigned Variable (Назначенная переменная)** элементов SIMATIC NET OPC Data control.

Синтаксис

```
object.DataType [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, которая определяет тип данных.

Можно выбрать следующие значения:

Тип данных	Степень	Размер	Описание
Boolean (Логический тип)	0	1 бит	Значение типа Boolean (битовое значение)
Byte (Байт)	1	1 байт	Однобайтовое значение без знака
Word (Слово)	2	2 байта	Двухбайтовое значение без знака
Integer (Целое число)	3	2 байта	Двухбайтовое целое значение со знаком
Double word (Двойное слово)	4	4 байта	Четырехбайтовое значение без знака (по умолчанию)
Double integer (Двойное целое число)	5	4 байта	Четырехбайтовое целое значение со знаком
Real (Вещественное число)	6	4 байта	Четырехбайтовое вещественное значение со знаком (с плавающей запятой)
Timers (Таймеры)	7	2 байта	Двухбайтовое значение без знака
Counter (Счетчик)	8	2 байта	Двухбайтовое значение без знака

Если параметр **DataType (Тип данных)** не указывается, по умолчанию используется тип данных "двойное слово".

Примечание

Параметр **DataType (Тип данных)** определяет, играет ли роль параметр **Precision (Точность)**.

Если длина данных, возвращаемых элементом **data control**, превышает длину данных, отображаемых в соответствующем элементе **SIMATIC control**, или длина OPC-переменной процесса выходит за диапазон значений, соответствующий типу данных, выбранному для элемента "Число", в этом случае вместо значения отображаются три точки (...). Если значение требуется изменить, необходимо удалить эти три точки.

6.1.57 **DefaultDeadband (Зона нечувствительности по умолчанию)**

■ SIMATIC NET OPC Data control

Определяет зону нечувствительности в процентах, которая будет использоваться OPC-сервером SIMATIC NET.

Если определен параметр **DefaultDeadband (Зона нечувствительности по умолчанию)**, OPC-сервер записывает изменившиеся значения переменных процесса в элемент OPC data control только в том случае, если уровень изменения значения превышает указанный процент от диапазона значений.

Синтаксис

```
object.DefaultDeadBand [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

value (значение)

Значение типа single (однобайтовое значение). Значение не должно быть отрицательным.

Примечание

Параметр **DefaultDeadband** можно использовать лишь после того, как будет определен диапазон значений для переменных с помощью конфигуратора символьных файлов (Symbol File Configurator). Параметр **DefaultDeadband** применяется ко всем переменным, назначенным элементу SIMATIC NET OPC Data control. Если для разных элементов визуализации в вашем приложении используются разные значения параметра **DefaultDeadband**, в этом случае необходимо использовать несколько элементов SIMATIC NET OPC Data control с различными параметрами **DefaultDeadband**, назначив им соответствующие элементы визуализации.

6.1.58 DefaultUpdateRate (Скорость обновления по умолчанию)

■ SIMATIC NET OPC Data control

Определяет период, с которым OPC-сервер обновляет данные.

Синтаксис

```
object.DefaultUpdateRate [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

value (значение)

Значение типа long (двойное слово).

Можно выбрать одно из следующих значений:

0	Используется минимальная скорость обновления, установленная OPC-сервером.
> 0	Все изменения назначенных переменных передаются сразу же по истечении этого интервала.

Примечание

В SIMATIC NET наименьший допустимый интервал обновления составляет 50 мс. Параметр DefaultUpdateRate применяется ко всем переменным, назначенным элементу SIMATIC NET OPC Data control. Если в вашем приложении для разных элементов визуализации используются разные значения параметра DefaultUpdateRate, в этом случае необходимо использовать несколько элементов SIMATIC NET OPC Data control с другими значениями параметра DefaultUpdateRate и назначить им соответствующие элементы визуализации.

6.1.59 Direction (Положение)

➤ SIMATIC Slider Control (элемент "Ползунок")

Определяет положение элемента "Ползунок":

- горизонтальное
- вертикальное

Синтаксис

`object.Direction [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC Slider Control (элемент "Ползунок").

value (значение)

Значение или константа, которая определяет ориентацию элемента.

Можно выбрать одно из следующих значений:

0	Горизонтальное расположение
1	Вертикальное расположение

6.1.60 Enabled (Разрешено)

➤ SIMATIC Button Control (элемент "Кнопка")

➤ SIMATIC Number Control (элемент "Число")

➤ SIMATIC Slider Control (элемент "Ползунок")

Данный параметр определяет, будет ли элемент реагировать на изменения свойства **Value (Значение)** и генерировать события.

Синтаксис

`object.Enabled [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC display control
(Элемент "Кнопка/Число/Ползунок").

value (значение)

Логическое выражение, которое определяет, будет ли элемент реагировать на события, формируемые пользователем.

Можно выбрать одно из следующих значений:

True (Верно)	Элемент реагирует на события.
False (Ложь)	Элемент не реагирует на события. Исключением является событие "Ошибка".

Если параметр **Enabled (Разрешено)** не указывается, элемент реагирует на события.

6.1.61 Factor (Коэффициент)

 SIMATIC Number Control (элемент "Число")

Определяет коэффициент масштабирования, который используется, если выбрана опция **ScaleByFormula (Масштабирование по формуле)**. Кроме коэффициента масштабирования для масштабирования также должен быть определен параметр **Offset (Смещение)**.

Для масштабирования значений можно использовать формулу:

(Значение в ПЛК * Коэффициент масштабирования) + **Смещение** =
Отображаемое значение

Значение в ПЛК – значение параметра **Value (Значение)**, когда элемент назначен OPC-серверу.

Коэффициент масштабирования – значение параметра **Factor (Коэффициент масштабирования)**.

Смещение – значение параметра **Offset (Смещение)**.

Отображаемое значение – содержимое параметра **Text (Текст)**.

Синтаксис

object.**Factor** [= значение]

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение типа float (число с плавающей запятой), которое определяет коэффициент для формулы масштабирования.

Если параметр **Factor (Коэффициент масштабирования)** не указывается, используется значение **1.0**.

Примечание

Для использования ScaleByFormula (Масштабирование по формуле) должен быть задан параметр ScaleMode (Режим масштабирования). В противном случае параметр Factor (Коэффициент масштабирования) не действует.

6.1.62 **FalseCaption (Текст в состоянии "Ложь")**

■ SIMATIC Button Control (элемент "Кнопка")

Данный параметр содержит текст, который отображается на элементе "Кнопка", если **Value (Значение)** содержит **False (Ложь)**.

Синтаксис

```
object.FalseCaption [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Button Control (элемент "Кнопка").

value (значение)

Строка с отображаемым текстом.

6.1.63 **FalseColor (Цвет в состоянии "Ложь")**

■ SIMATIC Button Control (элемент "Кнопка")

Определяет цвет элемента "Кнопка", если **Value (Значение)** содержит False (Ложь).

Синтаксис

```
object.FalseColor [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Button Control (элемент "Кнопка").

value (значение)

Значение или константа, которая определяет цвет заднего фона или цвет текста элемента.

Можно выбрать одно из следующих значений:

Стандартные цвета	Выбор цветов с помощью палитры RGB-цветов.
Цвета системы Windows	Выбор цветов с помощью системных цветов. Системные цвета зависят от контейнера.

6.1.64 Font (Шрифт)

■ SIMATIC Button Control (элемент "Кнопка")

■ SIMATIC Number Control (элемент "Число")

Определяет шрифт для элемента визуализации.

Синтаксис

object.**Font**

Пояснения

object (объект)

Экземпляр элемента SIMATIC Button Control (элемент "Кнопка") или экземпляр элемента SIMATIC Number Control (элемент "Число")

6.1.65 ForeColor (Цвет надписи)

■ SIMATIC Number Control (элемент "Число")

Определяет цвет надписи на элементе.

Синтаксис

object.**ForeColor** [= значение]

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, которое определяет цвет надписи на элементе.

Можно выбрать одно из следующих значений:

Стандартные цвета	Выбор цветов с помощью палитры RGB-цветов.
Цвета системы Windows	Выбор цветов с помощью системных цветов. Системные цвета зависят от контейнера.

6.1.66 LargeChange (Шаг грубой настройки)

■ SIMATIC Slider Control (элемент "Ползунок")

Определяет величину шага, на которую перемещается "Ползунок", когда происходит щелчок мышью по одной из стрелок "Ползунка".

Синтаксис

`object.LargeChange [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC Slider Control (элемент "Ползунок").

value (значение)

Значение или константа, которая определяет величину изменения (величину шага).

Примечание

Нажатие клавиши "page up" или щелчок по верхней (правой) стрелке "Ползунка" приводит к увеличению значения на величину LargeChange.

Нажатие клавиши "page down" или щелчок по нижней (левой) стрелке "Ползунка" приводит к уменьшению значения на величину LargeChange.

6.1.67 **LayoutType (Расположение)**

■ SIMATIC Number Control (элемент "Число")

Определяет способ расположения надписей элемента визуализации.

Синтаксис

`object.LayoutType [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, определяющая способ расположения.

Можно выбрать одно из следующих значений:

Горизонтальное расположение (0)	Первая надпись располагается слева, а вторая надпись – справа от поля со значением переменной.
Вертикальное расположение (1)	Первая надпись располагается сверху, а вторая – снизу от поля с отображением значения.

Если параметр **LayoutType (Тип расположения)** не указывается, надписи располагаются слева и справа от поля со значением переменной.

6.1.68 **LimitCheck (Проверка граничных значений)**

■ SIMATIC Number Control (элемент "Число")

Указывает, должно ли проверяться превышение граничных значений для введенного значения. Диапазон значений устанавливается параметрами UpperLimit (Верхняя граница) и LowerLimit (Нижняя граница).

Проверка определяет, находится ли значение, предоставляемое элементом, в пределах указанных границ. Значение, поступающее в элемент "Число" из OPC-сервера, не проверяется.

Если введенное значение превышает допустимый диапазон, элемент "Число" не записывает это значение в свойство **Value (значение)**. В этом случае формируется событие LimitCheckFailed (Ошибка проверки граничных значений).

Синтаксис

`object.LimitCheck [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, определяющая тип проверки граничных значений.

Можно выбрать одно из следующих значений:

Без проверки или 0	Введенное значение не проверяется.
Нижняя граница или 1	Проверяется только нижняя граница введенного значения
Верхняя граница или 2	Проверяется только верхняя граница введенного значения.
Нижняя и верхняя границы или 3	Проверяются и верхняя, и нижняя границы введенного значения.

Если параметр LimitCheck (Проверка граничных значений) не меняется, введенное число не проверяется.

Примечание

Параметр LimitCheck (Проверка граничных значений) должен быть активизирован, иначе параметры UpperLimit (Верхняя граница) и LowerLimit (Нижняя граница) не имеют силы.

6.1.69 Locked (Только чтение)

 SIMATIC Number Control (элемент "Число")

Устанавливает для элемента статус "Только чтение". В этом состоянии значение изменять нельзя, текущее значение только отображается.

Синтаксис

`object.Locked [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Логическое выражение, которое определяет, можно или нельзя менять значение.

Можно выбрать одно из следующих значений:

True (Верно)	Текст в элементе визуализации можно читать и выбирать, но нельзя изменить. Программа может изменить текст, изменив параметр Value (Значение) . Все изменения отображаются элементом.
False (Ложь)	Текст элемента можно изменять.

Если параметр Locked (Только чтение) не указывается, разрешается изменение значений.

6.1.70 LowerLimit (Нижняя граница)

■ SIMATIC Number Control (элемент "Число")

Указывает нижнюю границу для значений, вводимых пользователем. Значение проверяется, если активизирован параметр LimitCheck (Проверка граничных значений).

Если введено значение, меньшее, чем нижнее граничное значение, элемент не записывает это значение в свойство **Value (Значение)**. В этом случае формируется событие **LimitCheckFailed (Ошибка проверки граничных значений)**.

Синтаксис

```
object.LowerLimit [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение типа float (число с плавающей запятой), которое определяет нижнюю границу диапазона вводимых значений.

Примечание

Для LimitCheck (Проверка граничных значений) должно быть выбрано Lower (Нижняя граница) или Both (Обе границы), иначе LowerLimit (Нижняя граница) не имеет силы.

6.1.71 Max (Максимальное значение)

■ SIMATIC Slider Control (элемент "Ползунок")

Определяет максимальное значение для элемента "Ползунок". Если параметр **Value (Значение)** превышает или равен значению **Max (Максимальное значение)**, ручка "Ползунка" находится в крайнем правом или крайнем верхнем положении.

Синтаксис

```
object.Max [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Slider Control (элемент "Ползунок").

value (значение)

Значение или константа, которая определяет наибольшее отображаемое число.

6.1.72 Min (Минимальное значение)

■ SIMATIC Slider Control (элемент "Ползунок")

Определяет наименьшее значение для элемента "Ползунок". Если параметр **Value (Значение)** содержит значение, которое меньше или равно **Min (Минимальное значение)**, ручка "Ползунка" находится в крайнем левом или в крайнем нижнем положении.

Синтаксис

```
object.Min [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Slider Control (элемент "Ползунок").

value (значение)

Значение или константа, которая определяет наименьшее отображаемое число.

6.1.73 **nodeName (Имя узла)**

■ SIMATIC NET OPC Data control

Определяет имя удаленного компьютера, на котором работает OPC-сервер. Указывая имя узла, можно установить соединение по сети с помощью DCOM.

Синтаксис

`object.Node [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

value (значение)

Строка, которая определяет путь или идентификатор удаленного компьютера (ПК) для установления соединения.

6.1.74 **Offset (Смещение)**

■ SIMATIC Number Control (элемент "Число")

Определяет смещение, которое используется для опции **ScaleByFormula (Масштабирование по формуле)**, если она выбрана. Для масштабирования требуется задать не только **смещение**, но также и параметр **Factor (Коэффициент масштабирования)**.

Для масштабирования значений можно использовать формулу:

$(\text{Значение в ПЛК} * \text{Коэффициент масштабирования}) + \text{Смещение} = \text{Отображаемое значение}$

Значение в ПЛК – это значение свойства **Value (Значение)**, если элемент связан с OPC-сервером.

Коэффициент масштабирования – значение свойства **Factor (Коэффициент масштабирования)**.

Смещение – значение свойства **Offset (Смещение)**.

Отображаемое значение – содержимое свойства **Text (Текст)**.

Синтаксис

`object.Offset [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение типа float (число с плавающей запятой), которое определяет **смещение** для формулы масштабирования. Если параметр **Offset (Смещение)** не указывается, применяется значение **0.0**.

Примечание

Для параметра ScaleMode (Режим масштабирования) должно быть выбрано значение ScaleByFormula (Масштабирование по формуле = 1), иначе параметр Offset не будет действителен.

6.1.75 Precision (Точность)

➡ SIMATIC Number Control (элемент "Число")

Это свойство определяет точность, т.е., количество разрядов после запятой для вещественных чисел. Число округляется до указанного разряда.

Синтаксис

object.Precision [= значение]

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение типа integer (целое число), которое определяет точность числа. Если параметр **Precision (Точность)** не указан, используется значение **3**.

6.1.76 Radix (Представление)

➡ SIMATIC Number Control (элемент "Число")

Определяет способ представления отображаемого числа:

- двоичное представление
- восьмеричное представление
- десятичное представление
- шестнадцатеричное представление

Каждое значение должно вводиться с использованием выбранного способа представления.

Синтаксис

```
object.Radix [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, которая определяет способ представления отображаемого числа.

Можно выбрать одно из следующих значений:

Двоичное или 0	Двоичный режим. Допустимые значения: 0 и 1
Восьмеричное или 1	Восьмеричный режим. Допустимые значения: 0 ... 7
Десятичное или 2	Десятичный режим: Допустимые значения: 0 ... 9, '+', '-' и '.'
Шестнадцатеричное или 3	Шестнадцатеричный режим: Допустимые значения: 0 ... 9, a ... f и A ... F)

Если параметр Radix (Представление) не указывается, используется значение Decimal (Десятичное представление).

6.1.77 RangeRawMax (Максимальное входное значение)

■ SIMATIC Number Control (элемент "Число")

Вместе с параметром **RangeRawMin (Минимальное входное значение)** этот параметр определяет диапазон входных (первичных) значений. Под входными (первичными) значениями понимаются значения, которые поступают непосредственно из OPC-сервера. Диапазон входных значений можно привести к диапазону отображаемых значений с помощью элемента визуализации "Число". Диапазон отображаемых значений определяется параметрами **RangeScaledMin (Минимальное отображаемое значение)** и **RangeScaledMax (Максимальное отображаемое значение)**.

Параметр RangeRawMax определяет верхнюю границу диапазона первичных значений.

Преобразование диапазона значений путем указания границ диапазонов называют приведением диапазона. Для этого указываются следующие параметры:

- Диапазон входных (первичных) значений - для значений, поступающих от OPC-сервера
- Конечный диапазон - для значений, отображаемых элементом визуализации и вводимых в него

Значения одного диапазона автоматически приводятся к значениям другого диапазона.

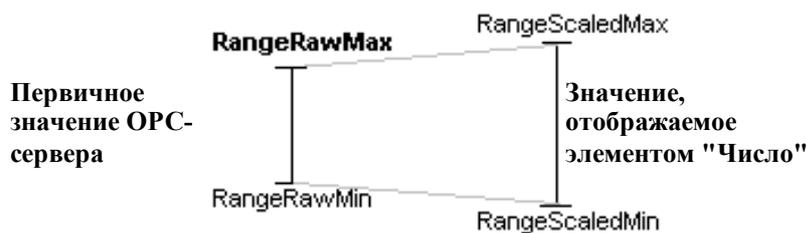


Рисунок 6–10 Приведение диапазонов

Синтаксис

```
object.RangeRawMax [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение типа float (число с плавающей запятой), которое определяет границы диапазона для преобразования диапазона.

Если параметр **RangeRawMax (Максимальное входное значение)** не указан, используется значение **255.0**.

Примечание

Для ScaleMode (Режим масштабирования) должно быть выбрано ScaleByRange (Масштабирование путем приведения диапазона), иначе RangeRawMax не будет иметь силы.

6.1.78 RangeRawMin (Минимальное входное значение)

■ SIMATIC Number Control (элемент "Число")

Вместе с параметром **RangeRawMax (Максимальное входное значение)** этот параметр определяет диапазон для входных (первичных) значений. Под входными (первичными) значениями понимаются значения, которые поступают непосредственно из OPC-сервера. Диапазон входных (первичных) значений можно привести к диапазону отображаемых значений с помощью элемента визуализации "Число". Диапазон отображаемых значений определяется параметрами **RangeScaledMin (Минимальное отображаемое значение)** и **RangeScaledMax (Максимальное отображаемое значение)**.

Параметр RangeRawMin определяет нижнюю границу диапазона входных (первичных) значений.

Преобразование диапазона значений путем указания границ диапазонов называют приведением диапазона. Указываются следующие параметры:

- Диапазон первичных значений - для значений, поступающих из OPC-сервера
- Конечный диапазон - для значений, отображаемых элементом визуализации и вводимых в него

Значения одного диапазона автоматически приводятся к значениям другого диапазона.

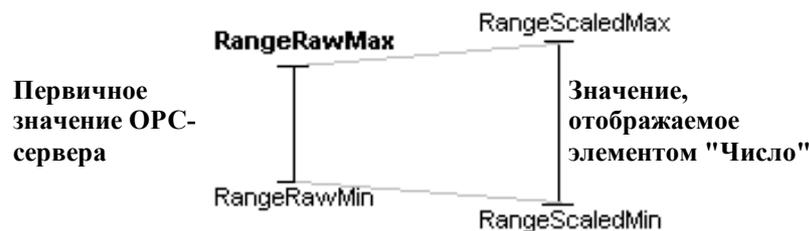


Рисунок 6–11

Приведение диапазонов

Синтаксис

```
object.RangeRawMin [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение типа float (число с плавающей запятой), которое определяет границы диапазона для преобразования диапазона.

Если параметр **RangeRawMin (Минимальное входное значение)** не указан, используется значение **0.0**.

Примечание

Для ScaleMode (Режим масштабирования) должно быть выбрано ScaleByRange (Масштабирование путем приведения диапазона), иначе RangeRawMin не будет иметь силы.

6.1.79 RangeScaledMax (Максимальное отображаемое значение)

■ SIMATIC Number Control (элемент "Число")

Вместе с параметром **RangeScaledMin (Минимальное отображаемое значение)** данный параметр определяет диапазон значений, отображаемых элементом "Число". Отображаемые значения рассчитываются по входным значениям, источником которых является OPC-сервер.

Параметр RangeScaledMax определяет наибольшее значение диапазона отображаемых чисел.

Масштабирование значений путем указания границ диапазонов называют "приведением диапазонов". Для этого указываются следующие параметры:

- Диапазон исходных значений - для значений, источником которых является OPC-сервер
- Конечный диапазон - для значений, которые отображаются и вводятся посредством элемента визуализации

Значения одного диапазона автоматически приводятся к значениям другого диапазона.

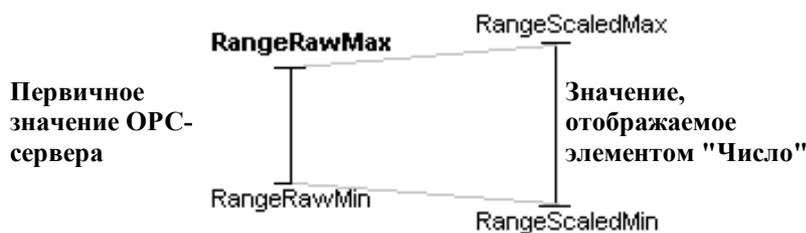


Рисунок 6–12 Приведение диапазонов

Синтаксис

```
object.RangeScaledMax [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение типа float (число с плавающей запятой), которое определяет границы конечного диапазона.

Если параметр **RangeScaledMax (Максимальное отображаемое значение)** не указывается, применяется значение **100.0**.

Примечание

Параметр ScaleMode (Режим масштабирования) должен содержать значение ScaleByRange (Масштабирование путем приведения диапазона), в противном случае RangeScaledMax не будет иметь силы.

6.1.80 RangeScaledMin (Минимальное отображаемое значение)

■ SIMATIC Number Control (элемент "Число")

Вместе с параметром **RangeScaledMax (Максимальное отображаемое значение)** данный параметр определяет диапазон значений, отображаемых элементом "Число". Отображаемые значения рассчитываются по входным (исходным) значениям, источником которых является OPC-сервер.

Параметр RangeScaledMin определяет наименьшее значение диапазона отображаемых чисел.

Масштабирование значений путем указания границ диапазонов называют "приведением диапазона". Для этого указываются следующие параметры:

- Диапазон исходных значений - для значений, источником которых является OPC-сервер
- Конечный диапазон - для значений, которые отображаются и вводятся посредством элемента визуализации

Значения одного диапазона автоматически приводятся к значениям другого диапазона.

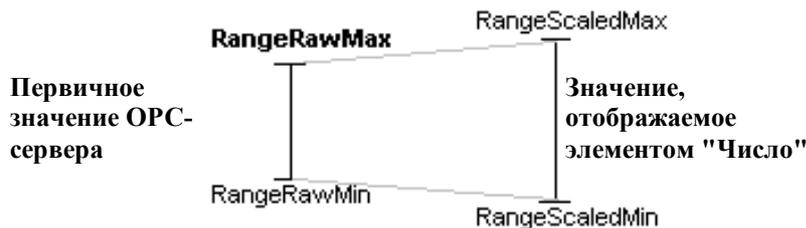


Рисунок 6–13 Приведение диапазонов

Синтаксис

`object.RangeScaledMin [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение типа float (число с плавающей запятой), которое определяет границы конечного диапазона.

Если параметр **RangeScaledMin (Минимальное отображаемое значение)** не изменен, применяется значение **0.0**.

Примечание

Параметр ScaleMode (Режим масштабирования) должен содержать значение ScaleByRange (Масштабирование путем приведения диапазонов), в противном случае RangeScaledMin не будет иметь силы.

6.1.81 ScaleMode (Режим масштабирования)

■ SIMATIC Number Control (элемент "Число")

Определяет режим масштабирования значений. Значение также можно указать с помощью параметра **ScaleProperty (Выбор масштабирования)**.

Можно выбрать один из следующих режимов масштабирования:

- По формуле
- Приведение диапазона

Масштабирование по формуле

(Значение в ПЛК * Коэффициент масштабирования) + **Смещение** =
Отображаемое значение

Значение в ПЛК – значение свойства **Value (Значение)**, когда элемент визуализации соединен с OPC-сервером.

Коэффициент масштабирования – значение свойства **Factor (Коэффициент масштабирования)**.

Смещение – значение свойства **Offset (Смещение)**.

Отображаемое значение – содержимое свойства **Text (Текст)**.

Приведение диапазона

Указывается диапазон значений процесса (диапазон входных значений) и диапазон отображаемых значений. Значения одного диапазона можно приводить к значениям другого диапазона (преобразование (приведение) диапазонов).

Синтаксис

```
object.ScaleMode [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, которая определяет режим масштабирования.

Можно выбрать одно из следующих значений:

ScaleNone (без масштабирования) или 0	Масштабирование не применяется.
ByFormula (по формуле) или 1	Для масштабирования значений применяется формула с параметрами Factor (Коэффициент масштабирования) и Offset (Смещение) .
ByRange (по диапазону) или 2	Для масштабирования значений используется метод приведения диапазона.

Если параметр **ScaleMode (Режим масштабирования)** не указан, масштабирование не осуществляется.

Примечание

Свойство ScaleMode (Режим масштабирования) определяет, будут ли действительны параметры Factor (Коэффициент масштабирования), Offset (Смещение), RangeRawMax (Максимальное входное значение), RangeRawMin (Минимальное входное значение), RangeScaledMax (Максимальное отображаемое значение) и RangeScaledMin (Минимальное отображаемое значение).

6.1.82 ServerName (Имя сервера)

■ SIMATIC NET OPC Data control

Параметр определяет имя (ProgID) OPC-сервера, являющегося источником требуемых переменных процесса.

Синтаксис

`object.ServerName [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

value (значение)

Строковая переменная, содержащая идентификатор (ProgID) OPC-сервера, который будет использоваться элементом.

Для SIMATIC NET значением может быть, например, **OPC.SimaticNET**.

6.1.83 ShowButtons (Показывать кнопки)

■ SIMATIC Number Control (элемент "Число")

Это свойство определяет, должны ли на элементе визуализации отображаться кнопки для ввода или удаления значений.

Синтаксис

`object.ShowButtons [= значение]`

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Логическое выражение, определяющее, будут ли отображаться кнопки.

Можно выбрать одно из следующих значений:

True (Верно)	Справа от поля значения отображаются кнопки. Кнопки активизируются при вводе нового значения. Если производится запись или изменение значения, новое значение должно быть подтверждено нажатием кнопки "Ввод". Чтобы отменить изменение, следует щелкнуть по кнопке "Удалить".
False (Ложь)	Кнопки не отображаются.

6.1.84 ShowErrorBoxes (Показывать сообщения об ошибках)

■ SIMATIC NET OPC Data control

■ SIMATIC Number Control (элемент "Число")

Параметр определяет, должны ли отображаться стандартные сообщения об ошибках в случае возникновения ошибок пользователя. Для каждой ошибки формируется событие **Error (Ошибка)**.

В случае возникновения любой ошибки, которая влияет на установление соединения с OPC-сервером, формируется событие **ConnectionError**.

Синтаксис

```
object.ShowErrorBoxes [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control или экземпляра элемента SIMATIC Number Control (элемент "Число")

value (значение)

Логическое выражение, которое определяет, должны ли отображаться сообщения об ошибках.

Можно выбрать одно из следующих значений:

True (Верно)	Элемент отображает указанные сообщения об ошибках.
False (Ложь)	Сообщения об ошибках не отображаются.

Если параметр **ShowErrorBoxes (Показывать сообщения об ошибках)** не указывается, элемент визуализации будет отображать стандартные сообщения об ошибках.

Примечание

В SIMATIC Computing сообщения об ошибках отображаются на английском языке. Если необходимо, чтобы сообщения отображались на других языках, опцию ShowErrorBoxes (Показывать сообщения об ошибках) следует отключить и написать собственную программу для обработки ошибок.

6.1.85 SmallChange (Шаг точной настройки)

■ SIMATIC Slider Control (элемент "Ползунок")

Параметр определяет величину шага, на которую перемещается указатель "Ползунка", когда элемент визуализации активен (установлен фокус) и происходит нажатие клавиш "Up/Down" или "Left/Right".

Нажатие клавиши "Right" (Вправо) или "Up" (Вверх) приводит к увеличению свойства **Value (Значение)** на величину SmallChange. Нажатие клавиш "Left" (Влево) или "Down" (Вниз) приводит к уменьшению параметра **Value (Значение)** на величину SmallChange.

Синтаксис

```
object.SmallChange [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Slider Control (элемент "Ползунок").

value (значение)

Значение или константа, которая определяет величину изменения.

6.1.86 Ticks (Метки шкалы)

■ SIMATIC Slider Control (элемент "Ползунок")

Параметр определяет количество меток на шкале элемента "Ползунок".

Если, к примеру, параметр **Ticks (Метки шкалы)** содержит значение **10**, шкала "Ползунка" будет разделена на **10** частей.

Синтаксис

```
object.Ticks [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Slider Control (элемент "Ползунок").

value (значение)

Константа, которая определяет количество отображаемых меток шкалы.

Если параметр **Ticks (Метки шкалы)** не указывается, используется принимаемое по умолчанию значение 10.

6.1.87 TrueCaption (Надпись в состоянии "верно")

■ SIMATIC Button Control (элемент "Кнопка")

Параметр определяет текст надписи, которая отображается на элементе визуализации, когда **Value (Значение)** содержит **True (Верно)**.

Синтаксис

```
object.TrueCaption [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Button Control (элемент "Кнопка").

value (значение)

Строка, содержащая текст.

6.1.88 TrueColor (Цвет в состоянии "верно")

■ SIMATIC Button Control (элемент "Кнопка")

Параметр определяет цвет элемента в случае, когда в **Value (Значение)** содержится **True (Верно)**.

Синтаксис

```
object.TrueColor [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Button Control (элемент "Кнопка").

value (значение)

Значение или константа, которая определяет цвет заднего фона или цвет шрифта текста на элементе визуализации.

Можно выбрать одно из следующих значений:

Стандартные цвета	Выбор цветов с помощью палитры RGB-цветов.
Цвета системы Windows	Выбор цветов с помощью системных цветов. Системные цвета зависят от контейнера.

6.1.89 UpperLimit (Верхняя граница)

➤ SIMATIC Number Control (элемент "Число")

Параметр определяет наибольшее значение, которое может быть введено пользователем. Вводимое значение проверяется, если оптимизирован параметр **LimitCheck (Проверка граничных значений)**.

Если вводится значение, которое превышает верхнее граничное значение, элемент визуализации не записывает это значение в параметр **Value (Значение)**. В этом случае формируется событие **Error (Ошибка)** и событие **LimitCheckFailed (Ошибка проверки граничного значения)**. Событие **Error (Ошибка)** формируется при условии, если активизирован параметр **ShowErrorBoxes (Показывать сообщения об ошибках)**.

Синтаксис

```
object.UpperLimit [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение типа Float (число с плавающей запятой), определяющее верхнюю границу для функции проверки граничных значений.

Если параметр **UpperLimit (Верхняя граница)** не указывается, используется значение **100.0**.

Примечание

Параметр LimitCheck (Проверка граничных значений) должен содержать значение Upper (Верхняя граница) или Both (Верхняя и нижняя границы), иначе UpperLimit (Верхняя граница) не вступит в силу.

6.1.90 Value (Значение)

➤ SIMATIC Button Control (элемент "Кнопка")

➤ SIMATIC Number Control (элемент "Число")

➤ SIMATIC Slider Control (элемент "Ползунок")

Данный параметр связывает элемент SIMATIC NET OPC Data control с переменной процесса OPC-сервера.

Value (Значение) может содержать значения с различными типами данных.

Синтаксис

```
object.Value (значение) [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Display control (Элемент визуализации).

value (значение)

Значение произвольного типа (тип variant), которое отображается или вводится посредством элемента визуализации.

Примечание

Если параметр Enabled (Разрешено) содержит значение "Ложь", связь между Value (Значение) и ConvertedValue (Преобразованное значение) перестает быть двунаправленной. Новые значения ConvertedValue (Преобразованное значение) не записываются в Value (Значение).

Когда изменяется Value (Значение), формируется событие ChangeEvent (Изменение значения).

6.1.91 WriteMode (Режим записи)

 SIMATIC Number Control (элемент "Число")

Параметр определяет, должны ли значения, которые вводятся в элемент "Число", автоматически записываться в свойство **Value (Значение)** и, следовательно, в элемент SIMATIC NET OPC Data control.

Синтаксис

```
object.WriteMode [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Значение или константа, которая определяет, будут ли вводимые значения автоматически записываться в свойство **Value (Значение)**.

Можно выбрать одно из следующих значений:

Automatic (Автоматически) или 0	Новое значение автоматически записывается в Value (Значение) и, следовательно, в OPC-сервер (режим по умолчанию).
Manual (Вручную) или 1	Новое значение не записывается, пока из программы не будет вызван метод Write (Запись).

Если параметр **WriteMode (Режим записи)** не указывается, значение записывается в **Value (Значение)** автоматически.

6.1.92 ZeroPad (Заполнение нулями)

■ SIMATIC Number Control (элемент "Число")

Параметр определяет, должны ли свободные разряды, располагающиеся слева от числа, отображаемого элементом визуализации, заполняться нулями. Количество нулей определяется типом данных.

Синтаксис

```
object.ZeroPad [= значение]
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

value (значение)

Логическое выражение, которое определяет, будет ли отображаемое число заполняться спереди нулями.

Можно выбрать одно из следующих значений:

True (Верно)	Свободные разряды, расположенные слева от числа, заполняются нулями, количество которых определяется типом данных числа.
False (Ложь)	Не заполнять число нулями.

Если параметр **ZeroPad (Заполнение нулями)** не указывается, число спереди нулями не заполняется.

6.1.93 AddNotification (Добавить уведомление)

■ SIMATIC NET OPC Data control

Данный параметр добавляет уведомление о событии в таблицу событий элемента OPC Data Control. Если переданная переменная процесса изменяется, формируется событие **ValueChanged (Значение изменилось)** с сообщенным идентификатором (ID).

Синтаксис

```
result=object.AddNotification(ItemIDs, UserIDs, Errors)
```

Пояснения

result (результат)

Значение типа long (Двойное слово), которое указывает, произошла ли ошибка. Если result = 0, ошибка не произошла.

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

ItemIDs (Идентификаторы элементов)

Идентификатор OPC-элемента типа String() (Строковая переменная), который определяет, какие именно переменные процесса контролируются на возникновение изменений.

UserID (Идентификатор пользователя)

Произвольный идентификатор типа String() (Строковая переменная), назначаемый каждому идентификатору элемента (ItemID). Этот идентификатор назначается пользователем. Он может использоваться в программе пользователя для идентификации события.

Errors (Ошибки)

Ответный параметр типа Variant (Произвольный тип), который содержит результат регистрации (успешно/неуспешно) для каждого идентификатора элемента (ItemID).

6.1.94 Connect (Соединить)

 SIMATIC NET OPC Data control

Параметр устанавливает сконфигурированное соединение с OPC-сервером.

Синтаксис

```
result = object.Connect
```

Пояснения

result (результат)

Значение типа long (Двойное слово), которое указывает, произошла ли ошибка. Если result = 0, ошибка не произошла.

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

Примечание

Если на этапе установления соединения произошла ошибка, это может быть определено с помощью события ConnectionError (Ошибка соединения), после чего запускается подпрограмма обработки ошибок.

6.1.95 Disconnect (Отсоединить)

■ SIMATIC NET OPC Data control

Параметр разрывает все соединения.

Синтаксис

```
result = object.Disconnect
```

Пояснения

result (результат)

Значение типа long (Двойное слово), которое указывает, произошла ли ошибка.

Если result = 0, ошибка не произошла.

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

6.1.96 GetItemProperties (Прочитать свойства элемента)

■ SIMATIC NET OPC Data control

Возвращает значения свойств (атрибутов) указанного OPC-элемента.

Синтаксис

```
result = object.GetItemProperties(ItemID, Count, PropertyIDs, PropertyValues, Errors)
```

Пояснения

result (Результат)

Значение типа long (Двойное слово), которое указывает, произошла ли ошибка.

Если result = 0, ошибка не произошла.

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

ItemID (Идентификатор элемента)

Идентификатор OPC-элемента типа String (Строковая переменная), связанного с соответствующей переменной процесса, для которого должно быть выполнено чтение текущих значений OPC-свойств.

Count (Количество параметров)

Значение типа long (Двойное слово), которое определяет количество свойств, значения которых должны быть прочитаны.

PropertyIDs (Идентификаторы свойств)

Параметр типа Variant (Произвольный тип), содержащий список идентификаторов свойств, которые должны быть прочитаны.

PropertyValues (Значения свойств)

Ответный параметр типа Variant (Произвольный тип), содержащий список запрошенных значений.

Errors (ошибки)

Ответный параметр типа Variant (Произвольный тип), содержащий результат регистрации (успешно/неуспешно) для каждого идентификатора свойства (PropertyID).

6.1.97 QueryAvailableItemProperties (Опросить имеющиеся свойства элемента)

■ SIMATIC NET OPC Data control

Параметр возвращает перечень OPC-свойств, а также их описание для указанного OPC-элемента.

Синтаксис

```
result=object.QueryAvailableItemProperties(ItemID,Count,PropertyIDs,Descriptions,DataTypes)
```

Пояснения

result (Результат)

Значение типа long (Двойное слово), которое указывает, произошла ли ошибка. Если result = 0, ошибка не произошла.

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

ItemID (Идентификатор элемента)

Идентификатор OPC-элемента типа String (Строковая переменная), связанного с соответствующей переменной процесса, для которого должна быть прочитана информация об OPC-параметрах.

Count (Количество параметров)

Ответный параметр типа long (Двойное слово), который указывает, сколько свойств существует у указанного OPC-элемента.

PropertyIDs (Идентификаторы свойств)

Ответный параметр типа Variant (Произвольный тип), который содержит перечень идентификаторов свойств (PropertyID), поддерживаемых указанным элементом.

Идентификатор свойства (PropertyID) используется как входной параметр для метода **GetItemProperties (Прочитать свойства элемента)**, который применяется для чтения текущего значения свойства.

Descriptions (Описания)

Ответный параметр типа Variant (Произвольный тип), который содержит список текстовых описаний для отдельных параметров.

DataTypes (Типы данных)

Ответный параметр типа Variant (Произвольный тип), который содержит список кодовых обозначений типов данных для отдельных свойств. Кодовые обозначения типов данных соответствуют кодовым обозначениям, установленным для COM.

6.1.98 ReadMultiVariables (Прочитать нескольких переменных)

■ SIMATIC NET OPC Data control

Параметр служит для чтения значений переменных, подключенных к OPC-серверу.

Синтаксис

```
result = object.ReadMultiVariables (VarItemIDs, VarValues,  
VarErrors {,VarQualities} {,VarTimeStamps})
```

Пояснения

result (Результат)

Значение типа long (Двойное слово), которое указывает, произошла ли ошибка.

Если result = 0, ошибка не произошла.

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

VarItemIDs (Идентификаторы элементов, связанных с переменными)

Переменная типа Variant (Произвольный тип), которая определяет массив идентификаторов OPC-элементов (OPC ItemID), устанавливающий, какие переменные OPC-сервера должны быть прочитаны.

VarValues (Значения переменных)

Ответная переменная типа Variant (Произвольный тип), содержащая массив соответствующих значений указанных переменных OPC-сервера.

VarErrors (Ошибки переменных)

Переменная типа Variant (Произвольный тип), содержащая массив статусов типа Long для возвращаемых переменных.

VarQualities (Коды качества переменных)

Ответная переменная типа Variant (Произвольный тип), содержащая массив соответствующих кодов качества для указанных переменных.

VarTimeStamps (Метки времени переменных)

Ответная переменная типа Variant (Произвольный тип), содержащая массив соответствующих меток времени указанных переменных.

6.1.99 ReadVariable (Прочитать переменную)

■ SIMATIC NET OPC Data control

Параметр служит для чтения указанной переменной из OPC-сервера.

Синтаксис

```
result = object.ReadVariable (ItemID, Value, Quality,
Timestamp)
```

Пояснения

result (Результат)

Значение типа long (Двойное слово), которое указывает, произошла ли ошибка.

Если result = 0, ошибка не произошла.

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

ItemID (Идентификатор элемента)

Строка с идентификатором элемента (ItemID), определяющая переменную OPC-сервера, которая должна быть прочитана.

Value (значение)

Ответная переменная типа Variant (Произвольный тип), содержащая значение указанной переменной OPC-сервера.

Quality (Код качества)

Ответное значение типа Long, содержащее код качества переменной.

TimeStamp (метка времени)

Ответная переменная типа Date (Дата), содержащая метку времени OPC.

6.1.100 Write (Записать)

■ SIMATIC Number Control (элемент "Число")

Параметр служит для записи значения **ConvertedValue (Преобразованное значение)** в **Value (Значение)**.

Это метод можно использовать только, если параметр **WriteMode (Режим записи)** содержит значение **Manual (Запись вручную)**.

Синтаксис

```
object.Write
```

Пояснения

object (объект)

Экземпляр элемента SIMATIC Number Control (элемент "Число")

6.1.101 WriteMultiVariables (Записать несколько переменных)

■ SIMATIC NET OPC Data control

Параметр служит для записи новых значений в несколько переменных процесса, предоставляемых OPC-сервером.

Синтаксис

```
result = object.WriteMultiVariables (VarItemIDs, VarValues, VarErrors)
```

Пояснения

result (Результат)

Значение типа long (Двойное слово), которое указывает, произошла ли ошибка. Если result = 0, ошибка не произошла.

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

VarItemIDs (Идентификаторы переменных процесса)

Переменная типа Variant (Произвольный тип), устанавливающая массив идентификаторов OPC-элементов (OPC ItemID), определяющих переменные процесса.

VarValues (Значения переменных)

Переменная типа Variant (Произвольный тип), содержащая массив соответствующих значений, которые должны быть записаны в указанные переменные.

VarErrors (ошибки значений)

Переменная типа Variant (Произвольный тип), содержащая массив состояний типа Long для каждой переменной.

6.1.102 WriteVariable (Записать переменную)

■ SIMATIC NET OPC Data control

Параметр служит для записи нового значения в указанную переменную процесса OPC-сервера.

Синтаксис

```
result = object.WriteVariable (ItemID, Value )
```

Пояснения

result (результат)

Значение типа long (Двойное слово), который указывает, произошла ли ошибка.

Если result = 0, ошибка не произошла.

object (объект)

Экземпляр элемента SIMATIC NET OPC Data control.

ItemID (идентификатор элемента)

Строка, содержащая идентификатор OPC-элемента (OPC ItemID), идентифицирующий переменную процесса OPC.

Value (значение)

Переменная типа Variant (Произвольный тип), содержащая значение, которое будет записано в указанную переменную процесса.

6.1.103 События элементов семейства SIMATIC

Для управляющих элементов предусмотрен ряд фиксированных событий, на которые программа может реагировать в режиме выполнения.

6.1.104 События элемента Data Control

Для элемента Data control предусмотрены следующие события:

Событие	причина
ConnectionError (Ошибка соединения)	Произошла ошибка соединения.
ValueChanged (Значение изменилось)	Изменилось значение или код качества переменной процесса, зарегистрированной посредством метода AddNotification (Добавить уведомление) . Путем вызова события ValueChanged (Значение изменилось) приложение пользователя уведомляется об изменении.

6.1.105 События элемента "Кнопка"

Для элемента "Кнопка" предусмотрены следующие события:

Событие	причина
ChangeEvent (Изменение значения)	Изменилось значение свойства (атрибута).
ClickEvent (Щелчок)	По элементу щелкнули кнопкой мыши.

6.1.106 События элемента "Число"

Для элемента "Число" предусмотрены следующие события:

Событие	причина
Change (Изменение значения)	Изменилось значение свойства Value (Значение) .
Click (Щелчок)	По элементу визуализации щелкнули кнопкой мыши (нажали и отпустили кнопку).
DoubleClick (Двойной щелчок)	По элементу дважды щелкнули кнопкой мыши.
Error (Ошибка)	Была обнаружена ошибка элемента.
KeyDown (Нажатие клавиши)	Когда элемента был активен, была нажата клавиша.
KeyPress (Нажатие и отпускание клавиши)	Когда элемент был активен, была нажата и отпущена клавиша.
KeyUp (Отпускание клавиши)	Когда элемент был активен, была отпущена клавиша.
LimitCheckFailed (Ошибка проверки граничных значений)	Превышено граничное значение.
MouseDown (Нажатие кнопки мыши)	Кнопка мыши была нажата, когда указатель мыши находился на элементе.

MouseMove (Наведение указателя мыши)	На элемент был наведен указатель мыши.
MouseUp (Отпускание кнопки мыши)	Кнопка мыши была отпущена, когда указатель мыши находился на элементе.

6.1.107 События элемента "Ползунок"

Для элемента "Ползунок" предусмотрены следующие события:

Событие	причина
Change (Изменение значения)	Изменилось значение свойства Value (Значение) .
Click (Щелчок)	По элементу визуализации щелкнули кнопкой мыши (нажали и отпустили кнопку).
DbClick (Двойной щелчок)	По элементу дважды щелкнули кнопкой мыши.
Error (Ошибка)	Была обнаружена ошибка элемента.
KeyDown (Нажатие клавиши)	Когда элемент был активен, была нажата клавиша.
KeyPress (Нажатие и отпускание клавиши)	Когда элемент был активен, была нажата и отпущена клавиша.
KeyUp (Отпускание клавиши)	Когда элемент был активен, была отпущена клавиша.
MouseDown (Нажатие кнопки мыши)	Кнопка мыши была нажата, когда указатель мыши находился на элементе.
MouseMove (Наведение указателя мыши)	На элемент был наведен указатель мыши.
MouseUp (Отпускание кнопки мыши)	Кнопка мыши была отпущена, когда указатель мыши находился на элементе.

6.1.108 Сообщения об ошибках, предусмотренные для элементов визуализации SIMATIC

Элементы Data control (Элемент управления данными) и Number control (Элемент "Число") могут генерировать сообщения об ошибках.

Сообщения об ошибках элемента Data Control

Если происходит ошибка в элементе Data Control, этот элемент генерирует событие **ConnectionError (Ошибка соединения)**. Программа пользователя может принимать событие **ConnectionError (Ошибка соединения)** и реагировать на него. Событие **ConnectionError** может формироваться в случае возникновения стандартных ошибок COM, например, **E_FAIL** или **E_OUTOFMEM**.

Таблицы сообщений об ошибках

В таблицах перечислены только наиболее важные сообщения об ошибках.

6.1.109 Сообщения об ошибках элемента Data Control

В следующей таблице перечислены наиболее важные сообщения об ошибках:

Код ошибки	Значение
0x80004005	Общая ошибка COM.
0x8007000E	Свободной памяти недостаточно.
0x80070057	Ошибка синтаксиса переменной.
0xC0040007	Недопустимый тип переменной.
0xC004000B	Записываемое значение слишком велико.

6.1.110 Сообщения об ошибках элемента "Число"

В следующей таблице перечислены наиболее важные сообщения об ошибках:

Код ошибки	Значение
C0040002	<p>Масштабирование не может быть выполнено, поскольку формула масштабирования некорректна.</p> <p>Данная ошибка происходит, если указывается диапазон исходных значений (RangeRawMin, RangeRawMax) нулевой длины (Min = Max). Это приведет к делению на 0, поэтому масштабирование невозможно.</p> <p>Следует указать диапазон исходных значений, у которого RangeRawMin меньше, чем RangeRawMax.</p>
C0040003	<p>Значение параметра Value (Значение) некорректно.</p> <p>Значение, сообщаемое OPC-сервером или скриптом, который обращается к Value (Значение), не может быть интерпретировано.</p> <p>Проверьте значения, которые записываются в OPC-сервер.</p>
C0040004	<p>Недопустимое значение введенного текста (значения).</p> <p>Это ошибка часто происходит, когда в поле элемента визуализации введено некорректное значение. Как правило, она возникает в случае ввода недопустимого символа.</p> <p>Набор допустимых символов определяется форматом представления числа или используемым типом данных. Более подробную информацию можно найти в описании основных параметров значения</p> <p>Введите допустимое значение</p>

Код ошибки	Значение
C0040005	<p>Не был обнаружен объект COM.</p> <p>Неправильно установлен продукт SIMATIC или Windows. Управляющий элемент не может установить связь с другими компонентами, необходимыми для правильной работы с программой.</p> <p>Проверьте установку программных компонентов.</p>
C0040006	<p>Не могут быть созданы стандартные управляющие элементы Microsoft (Microsoft Standard Controls).</p> <p>Неправильно установлен продукт SIMATIC или Windows. Управляющий элемент не может установить связь с другими компонентами, необходимыми для правильной работы с программой.</p> <p>Проверьте установку программных компонентов.</p>
C0040010	<p>Проверка граничных значений не может быть выполнена, поскольку значение нижней границы превышает значение верхней границы.</p> <p>Эта ошибка возникает, когда предусмотрена проверка граничных значений элемента "Число". В этом случае причиной является то, что нижняя граница (LowerLimit) превышает верхнюю границу (UpperLimit).</p> <p>Задайте правильно диапазон для проверки граничных значений.</p>

6.2 Программирование интерфейса автоматизации

Интерфейс автоматизации (Automation Interface) является дополнением к пользовательскому интерфейсу (Custom Interface). Интерфейс автоматизации предоставляет пользователю удобства современных средств разработки и языков описания сценариев, позволяя использовать их для программирования OPC. Интерфейс автоматизации можно использовать как для обращения к переменным процесса (Data Access), так и для обработки событий и аварийных сообщений (Alarms & Events).

Применение

Интерфейс автоматизации применяется, когда требуется создать приложение, оперирующее не очень большим количеством переменных и не предъявляющее высоких требований к пропускной способности канала связи, базирующееся на офисном приложении или Visual Basic.

6.2.1 Программирование интерфейса автоматизации для доступа к данным

Для доступа к данным (Data Access) предусмотрена простая модель классов, в которой интерфейсы и принадлежащие им методы сгруппированы в отдельные классы.

Интерфейс автоматизации делает модель классов, применяемую для пользовательского интерфейса, более совершенной, предоставляя пользователю преимущества и возможности, которые заложены в таких системах структурированного программирования, как Visual Basic.

6.2.2 Что предоставляет пользователю объектная модель OPC Data Access?

В классах модели Data Access содержатся следующие объекты:

- OPCServer (OPC-сервер)
- OPCGroup (OPC-группа)
- OPCItem (OPC-элемент)

Для интерфейса автоматизации могут быть добавлены дополнительные объекты. Для управления объектами OPCGroup (OPC-группа) и OPCItem (OPC-элемент) предусмотрены отдельные собирательные объекты (семейства): OPCGroups (OPC-группы) и OPCItems (OPC-элементы). Собирательные объекты предоставляют функции для управления назначенными им объектами.

Также предусмотрен собирательный объект-семейство OPCBrowser (OPC-Браузер), предоставляющий функции обзора.

Объектная модель

На следующем рисунке показаны объекты и взаимосвязи между объектами.

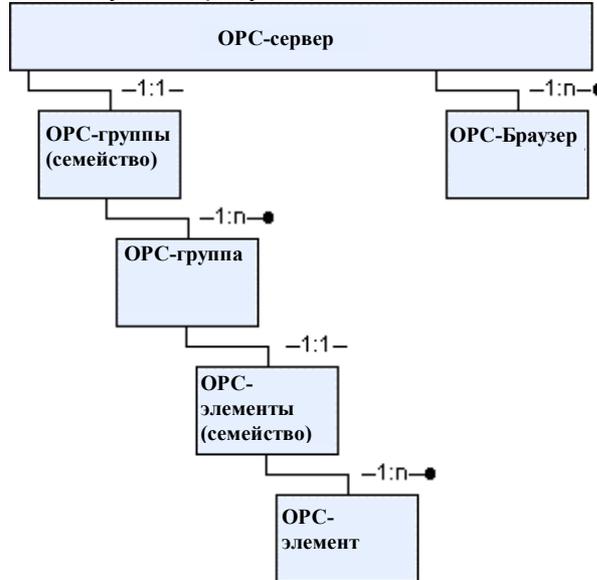


Рисунок 6–14 Объектная модель OPC Data Access

6.2.3 Что требуется учитывать при программировании

- При работе с интерфейсом автоматизации необязательные (опциональные) параметры должны передаваться как объекты произвольного типа (*variant*). В то же время в Visual Basic опциональные параметры должны объявляться как опциональные переменные определенного типа. Благодаря этому объекты произвольного типа (*variant*) содержат данные надлежащего типа.
- Асинхронные функции
Чтобы был доступен мониторинг переменных, специальное свойство группы **IsSubscribed** должно содержать **True (Верно)**.
- В Visual Basic объекты, которые будут принимать события, должны объявляться с дополнением **withEvents**.

Пример:

```
Dim WithEvents MyOPCGroup as OPCGroup
```

- В соответствии со спецификацией OPC массивы всегда начинаются с индекса 1. В программе это должно указываться так:

```
Option Base 1 'ARRAYs are always Option Base 1
```

6.2.4 Объекты интерфейса автоматизации для доступа к данным

Ниже представлен перечень свойств (атрибутов), методов и событий, предусмотренных для интерфейса Data Access (Доступ к данным). Описаны только свойства, характерные для SIMATIC NET. Подробное описание свойств, методов и событий следует искать в соответствующих спецификациях OPC.

Объекты

Существуют следующие объекты, предназначенные для интерфейса автоматизации для доступа к данным:

Объект OPCServer (OPC-сервер)

Объекты OPCServer класса "OPC server" создаются клиентом.

В свойствах объектов OPCServer содержится общая информация о сервере. Одновременно с созданием объекта также создается собирательный объект (семейство) OPCGroups (OPC-группы) в качестве свойства объекта OPCServer.

Свойства объекта OPCServer

Ниже представлен список свойств объекта OPCServer. Описаны только свойства, характерные для SIMATIC NET. Описание свойств можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Свойство	Назначение
Bandwidth	Возвращает "полосу пропускания" сервера. Это свойство не поддерживается OPC-сервером SIMATIC NET.
BuildNumber	Возвращает внутренний номер сервера
ClientName	Указывает имя клиента. Имя клиента (ClientName) используется, главным образом, для тестирования.
CurrentTime	Возвращает текущее время (UTC).
LastUpdateTime	Возвращает время (UTC), когда сервер последний раз передавал данные клиенту
LocaleID	Указывает язык для отображения надписей. Для SIMATIC NET может быть выбран немецкий или английский язык.
MajorVersion	Возвращает основной номер версии сервера
MinorVersion	Возвращает второстепенный номер версии сервера

Свойство	Назначение
OPCGroups	Указывает семейство объектов OPCGroup (OPC-группа)
PublicGroupNames	Возвращает имя общей группы (public group) OPC-сервера. Опциональные общие группы OPC-сервером SIMATIC NET не поддерживаются.
ServerName	Возвращает имя подключенного OPC-сервера
ServerNode	Возвращает имя узла сети, в котором расположен OPC-сервер
ServerState	Возвращает состояние сервера
StartTime	Возвращает время (UTC), когда был запущен сервер
VendorInfo	Возвращает сведения о производителе. OPC-сервер SIMATIC NET возвращает следующую информацию: SIMATIC NET OPC Data control-Server .

Методы объекта OPCServer

Ниже перечислены методы объекта OPCServer. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Метод	Назначение
Connect	Устанавливает соединение с OPC-сервером. В качестве <i>ProgID</i> для OPC-сервера можно указать, например, OPC.SimaticNET .
CreateBrowser	Создает объект-семейство OPCBrowser (OPC-Браузер)
Disconnect	Разрывает соединение с OPC-сервером. OPC-сервер SIMATIC NET разрывает все коммуникационные соединения с OPC-клиентом после того, как будет удалено последнее соединение.
GetErrorString	Читает сообщение об ошибке с указанным кодом ошибки. OPC-сервер SIMATIC NET поддерживает сообщения об ошибках на немецком и английском языках. Сообщения об ошибках операционной системы Windows отображаются на языке, который используется в операционной системе.
GetItemProperties	Возвращает список, содержащий значения запрошенных свойств

Метод	Назначение
GetOPCServers	Возвращает имена зарегистрированных OPC-серверов. Например, OPC-сервер SIMATIC NET имеет имя OPC.SimaticNET .
LookupItemIDs	Возвращает список с идентификаторами элементов (ItemID), которые соответствуют идентификаторам свойств (PropertyID). OPC-сервер SIMATIC NET не возвращает идентификаторы свойств, которые могли бы отображаться в качестве идентификаторов элементов.
QueryAvailableLocaleIDs	Возвращает коды доступных языков. OPC-сервер SIMATIC NET поддерживает сообщения об ошибках на немецком и английском языках. Сообщения об ошибках операционной системы Windows отображаются на языке, который используется операционной системой.
QueryAvailableProperties	Возвращает коды свойств и свойства OPC-элемента.

События объекта OPCServer

Ниже приведено описание события, предусмотренного для объекта OPCServer. Описано только событие, характерное для SIMATIC NET. Описание события можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Событие	Назначение
ServerShutDown	Событие запускается, когда OPC-сервер выключается. OPC-сервер SIMATIC NET запускает это событие, когда из программы конфигурирования поступает команда отключения, либо когда на ПК-станцию поступают новые конфигурационные данные.

Объект-семейство OPCBrowser

С помощью собирательного объекта-семейства OPCBrowser (OPC-браузер) можно просматривать пространство имен OPC-сервера.

Объект класса OPCBrowser создается с помощью метода **CreateBrowser** объекта OPCServer. Для одного сервера можно создать несколько объектов OPCBrowser.

Свойства объекта OPCBrowser

Ниже приведен перечень свойств собирательного объекта-семейства OPCBrowser (OPC-браузер). Описаны только свойства, характерные для SIMATIC NET. Описание свойств можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Свойство	Назначение
AccessRights	Определяет права доступа для метода ShowLeafs
Count	Возвращает количество элементов
CurrentPosition	Возвращает текущее положение на дереве пространства имен
DataType	Возвращает тип данных для метода ShowLeafs
Filter	Определяет фильтр для методов ShowLeafs и ShowBranches
Organization	Возвращает организационную структуру пространства имен. Пространство имен OPC-сервера SIMATIC NET имеет иерархическую структуру.

Методы объекта OPCBrowser

Ниже перечислены методы собирательного объекта-семейства OPCBrowser. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Метод	Назначение
GetAccessPaths	Чтение пути доступа для указанного идентификатора элемента (ItemID). Для OPC-сервера SIMATIC NET не следует использовать параметр AccessPath.
GetItemID	Этот метод возвращает идентификатор элемента (ItemID) в пространстве имен OPC-сервера, если это возможно.
Item	Указывает имя элемента
MoveDown	Переход на один уровень вниз в пространстве имен.

Метод	Назначение
MoveTo	Переход на указанную позицию в пространстве имен.
MoveToRoot	Переход в начало пространства имен
MoveUp	Переход на один уровень вверх в пространстве имен
ShowBranches	Указывает имена ветвей в текущей позиции обзора
ShowLeafs	Указывает имена объектов ("листья") в текущей позиции обзора

Объект-семейство OPCGroups

Собирательный объект OPCGroups (OPC-группы) – это семейство объектов OPCGroup (OPC-группа). Стандартные значения для всех вновь создаваемых OPC-групп определяются свойствами семейства OPCGroups.

Если объект OPCServer успешно выполняет вызов **Connect (Соединить)**, в качестве свойства объекта OPCServer создается объект-семейство OPCGroups.

Примечание

Опциональные общие группы (public group) не поддерживаются OPC-сервером SIMATIC NET.

Свойства объекта OPCGroups

Ниже перечислены свойства собирательного объекта-семейства OPCGroups (OPC-группы). Описаны только свойства, характерные для SIMATIC NET. Описание свойств можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Свойство	Значение
Count	Возвращает количество групп
DefaultGroupDeadband	Указывает исходное значение свойства Deadband для вновь создаваемых объектов OPCGroup
DefaultGroupsActive	Указывает исходное значение свойства ActiveState для вновь создаваемых объектов OPCGroup

Свойство	Значение
DefaultGroupLocaleID	Указывает исходное значение свойства LocaleID для вновь создаваемых объектов OPCGroup
DefaultGroupTimeBias	Указывает исходное значение свойства TimeBias для вновь создаваемых объектов OPCGroup
DefaultGroupUpdateRate	Указывает исходное значение свойства UpdateRate для вновь создаваемых объектов OPCGroup
Parent	Возвращает ссылку на соответствующий объект OPCServer (OPC-сервер)

Методы объекта OPCGroups

Ниже перечислены методы собирательного объекта-семейства OPCGroups (OPC-группы). Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Метод	Назначение
Add	Создает новый объект OPCGroup и добавляет его в семейство
GetOPCGroup	Указывает ссылку на имя или дескриптор сервера объекта OPCGroup
Item	Возвращает ссылку объект семейства с указанным индексом
Remove	Удаляет группу, принадлежащую серверу
RemoveAll	Удаляет все группы сервера

События объекта OPCGroups

Ниже описано событие, предусмотренное для собирательного объекта-семейства OPCGroups (OPC-группы). Описано только событие, характерное для SIMATIC NET. Подробное описание события можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Событие	Назначение
GlobalDataChange	Уведомляет об изменении значения и статуса активных элементов всех активных групп

Объект OPCGroup

Класс OPCGroup (OPC-группа) предназначен для управления отдельными переменными процесса, OPC-элементами. С помощью объекта OPCGroup клиент может объединять отдельные OPC-элементы в единую группу по ряду семантических признаков и обращаться с этой группой как с единым объектом.

Мониторинг переменных, а также чтение и запись относятся ко всей группе. Например, мониторинг всех OPC-элементов группы можно активизировать, вызвав одну единственную функцию.

Таким образом, все переменные процесса, которые, например, отображаются на одной экранной форме станции операторского управления и визуализации, следует размещать в одну и ту же группу и активизировать мониторинг переменных при открытии окна одной операцией.

Примечание

Опциональные общие группы (public group) не поддерживаются OPC-сервером SIMATIC NET.

Свойства объекта OPCGroup

Ниже перечислены свойства объекта OPCGroup (OPC-группа). Описаны только свойства, характерные для SIMATIC NET. Описание свойств можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Свойство	Назначение
ClientHandle	Указывает дескриптор для локализации данных
DeadBand	Указывает "зону нечувствительности", в пределах которой не производится уведомление об изменении значений.
IsActive	Определяет статус группы. Обязательно запишите в свойство IsActive значение True (Верно) , если необходим мониторинг переменных этой группы.
IsPublic	Указывает, является ли группа общей. Опциональные общие группы OPC-сервером SIMATIC NET не поддерживаются.
IsSubscribed	Указывает, будут ли контролироваться переменные группы. Если требуется мониторинг переменных этой группы, обязательно запишите в IsSubscribed значение True (Верно) .

Свойство	Назначение
LocaleID	Указывает язык текстовой строки, возвращаемой сервером.
Name	Указывает имя группы
OPCItems	Указывает объект-семейство, предназначенный для управления элементами группы
Parent	Возвращает ссылку на соответствующий объект OPCServer (OPC-сервер)
ServerHandle	Возвращает уникальный дескриптор группы
TimeBias	Возвращает величину приращения времени для преобразования метки времени в локальное время
UpdateRate	Определяет периодичность, с которой клиент должен получать уведомления об изменении значений или изменении статуса элементов

Методы объекта OPCGroup

Ниже перечислены методы объекта OPCGroup (OPC-группа). Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Automation Interface
 (Интерфейс автоматизации для доступа к данным)
 Версия 2.02
 Февраль 4, 1999

Метод	Назначение
AsyncCancel	Отменяет асинхронное задание
AsyncRead	Передает команду асинхронного чтения
AsyncRefresh	Создает событие для каждого активного OPC-элемента с текущим значением из кеша.
AsyncWrite	Передает команду асинхронного чтения
SyncRead	Запускает синхронное чтение значений, сведений о состояниях и меток времени для одного или нескольких элементов группы.
SyncWrite	Запускает синхронную запись значений для одного или нескольких элементов группы.

События объекта OPCGroup

OPC-интерфейс автоматизации возвращает изменения значений активных элементов и результаты асинхронных операций посредством событий.

В Visual Basic объекты, которые предназначены для приема событий, должны объявляться с дополнением **withEvents**.

Ниже приведен перечень событий, предусмотренных для объекта OPCGroup (OPC-группа). Описаны только события, характерные для SIMATIC NET. Описание событий можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Событие	Назначение
AsyncCancelComplete	Запускается по завершению задания на отмену.
AsyncReadComplete	Запускается по завершению задания на чтение
AsyncWriteComplete	Запускается по завершению задания на запись
DataChange	Запускается по изменению значения или кода качества одного или нескольких элементов.

Собирательный объект-семейство OPCItems

Объект-семейство OPCItems (OPC-элементы) – это семейство объектов OPCItem (OPC-элемент), предназначенное для создания и управления OPC-элементами. Стандартные значения для всех создаваемых OPC-элементов определяются свойствами объекта OPCItems.

Если создается объект OPCGroup (OPC-группа), одновременно автоматически создается объект-семейство OPCItems. OPCItems всегда существует как свойство группы и используется для мониторинга OPC-переменных.

Свойства объекта OPCItems

Ниже перечислены свойства объекта OPCItems (OPC-элементы). Описаны только свойства, характерные для SIMATIC NET. Описание свойств можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Свойство	Назначение
Count	Возвращает количество элементов в группе.
DefaultAccessPath	Определяет начальное значение свойства AccessPath для вновь создаваемых OPC-элементов. Для OPC-сервера SIMATIC NET свойство DefaultAccessPath должно быть пустым.
DefaultIsActive	Определяет начальное значение свойства ActiveState для вновь создаваемых OPC-элементов
DefaultRequestedDataType	Определяет начальное значение свойства RequestedDataType для вновь создаваемых OPC-элементов
Parent	Возвращает ссылку на соответствующий объект OPCGroup (OPC-группа)

Методы объекта OPCItems

Ниже перечислены методы объекта OPCItems (OPC-элементы). Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Метод	Назначение
AddItem	Добавляет новый OPC-элемент в семейство.
AddItems	Добавляет несколько OPC-элементов в семейство
GetOPCItem	Содержит ссылку на дескриптор сервера, созданный с помощью AddItem
Item	Содержит ссылку на элемент семейства
Remove	Удаляет один или несколько элементов из группы
SetActive	Делает активным один или несколько элементов группы
SetClientHandles	Изменяет дескриптор клиента для одного или нескольких элементов
SetDataTypes	Устанавливает тип данных для одного или нескольких элементов.
Validate	Проверяет действительность для одного или нескольких OPC-элементов

Объект OPCItem

Объект класса OPC Item (OPC-элемент) представляет переменную процесса, например, модуль ввода программируемого контроллера. Переменная процесса – это записываемый и/или читаемый элемент данных, представляющий точку ввода/вывода процесса, например, температуру в резервуаре. Каждая переменная процесса характеризуется значением, кодом качества и меткой времени.

Свойства объекта OPCItem

Ниже перечислены свойства объекта OPCItem (OPC-элемент). Описаны только свойства, характерные для SIMATIC NET. Описание свойств можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Свойство	Назначение
AccessPath	Возвращает путь доступа к элементу. Для OPC-сервера SIMATIC NET свойство Access Path должно быть пустым.
AccessRights	Возвращает права доступа к переменным.
CanonicalDataType	Возвращает исходный тип данных элемента.
ClientHandle	Возвращает дескриптор для упрощенного присвоения переменных процесса внутренним структурам данных клиента.
EUInfo	Возвращает сведения о единицах измерения значения (опция). OPC-сервер SIMATIC NET не поддерживает единицы измерения (инженерные единицы).
EUType	Возвращает единицу измерения для возвращенного значения (опция). OPC-сервер SIMATIC NET не поддерживает единицы измерения (инженерные единицы).
IsActive	Определяет, формируются ли для элемента события-уведомления.
ItemID	Возвращает уникальное имя элемента.
Parent	Возвращает ссылку на родительский объект OPCGroup (OPC-группа)
Quality	Возвращает код качества для последнего прочитанного значения.
RequestedDataType	Определяет запрошенный тип данных для значения элемента.
ServerHandle	Возвращает дескриптор сервера, предназначенный для идентификации элемента
TimeStamp	Возвращает время последнего чтения значения.
Value (значение)	Возвращает последнее действительное значение элемента

Методы объекта OPCItem

Ниже перечислены методы объекта OPCItem (OPC-элемент). Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Automation Interface
(Интерфейс автоматизации для доступа к данным)
Версия 2.02
Февраль 4, 1999

Метод	Назначение
Read	Выполняет синхронное чтение значения, кода качества и/или метки времени переменной.
Write	Устанавливает синхронно значение переменной.

6.2.5 Программирование интерфейса автоматизации для доступа к аварийным сообщениям и событиям

Для доступа к аварийным сообщениям и событиям (Alarms & Events) предусмотрена простая модель классов, в которой интерфейсы и принадлежащие им методы сгруппированы в отдельные классы.

Интерфейс автоматизации делает модель классов, применяемую для пользовательского интерфейса, более совершенной, предоставляя пользователю преимущества и возможности, которые заложены в таких системах структурированного программирования, как Visual Basic.

6.2.6 Что предоставляет объектная модель OPC Alarms & Events?

В классах модели Alarms & Events содержатся следующие объекты:

- OPCEventServer (OPC-сервер событий)
- OPCEventSubscriptions (OPC-подписки на события)
- OPCEventSubscription (OPC-подписка на событие)
- OPCEventAreaBrowsers (OPC-браузеры областей событий)
- OPCEventAreaBrowser (OPC-браузер областей событий)
- OPCEvents (OPC-события)
- OPCEvent (OPC-событие)
- OPCEventCondition (OPC-условие события)

- OPCEventSubConditions (OPC-субусловия события)
- OPCEventSubCondition (OPC-субусловие события)

Для интерфейса автоматизации могут быть добавлены дополнительные объекты.

Примечание

Поскольку сервер событий SIMATIC NET является простым сервером событий (Simple Event Server), следующие объекты не поддерживаются: OPCEventAreaBrowsers, OPCEventAreaBrowser, OPCEventCondition, OPCEventSubConditions, OPCEventSubCondition.

Объектная модель

На следующем рисунке показаны объекты и взаимосвязи между объектами.

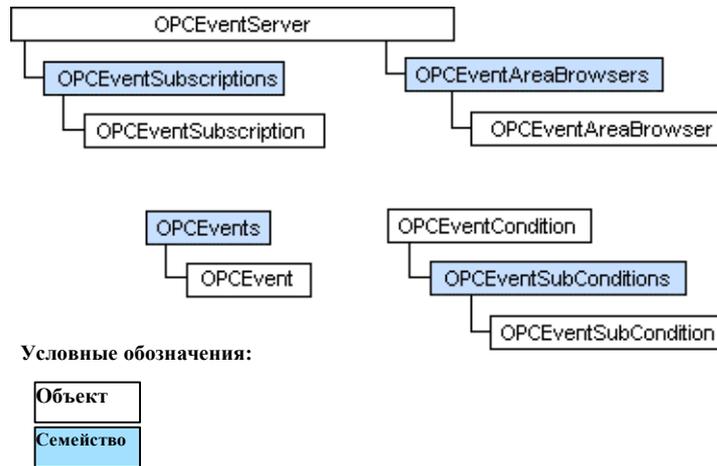


Рисунок 6–15 Объектная модель OPC Alarms & Events

6.2.7 О чем следует помнить при программировании

При работе с интерфейсом автоматизации необязательные (опциональные) параметры должны передаваться с произвольным типом (*variant*). В то же время в Visual Basic опциональные параметры должны объявляться с конкретным типом. Благодаря этому объекты произвольного типа (*variant*) содержат данные надлежащего типа.

- Асинхронные функции
Чтобы был возможен мониторинг переменных, в специальное свойство группы **IsSubscribed** должно быть записано значение **True (Верно)**.
- В Visual Basic объекты, которые будут принимать события, должны объявляться с дополнением **withEvents**.

Пример:

```
Dim WithEvents MyOPCGroup as OPCGroup
```

- В соответствии со спецификацией OPC массивы всегда начинаются с индекса 1. В программе это должно указываться так:

```
Option Base 1 'ARRAYs are always Option Base 1
```

6.2.8 Объекты интерфейса автоматизации для доступа к аварийным сообщениям и событиям

Ниже представлен перечень свойств (атрибутов), методов и событий, предусмотренных для интерфейса Alarms & Events (Доступ к аварийным сообщениям и событиям). Описаны только свойства, характерные для SIMATIC NET. Подробное описание свойств, методов и событий следует искать в соответствующих спецификациях OPC.

Объекты

Ниже перечислены объекты, предусмотренные для интерфейса автоматизации для доступа к аварийным сообщениям и событиям:

Объект OPCEventServer

Объекты OPCEventServer (OPC-сервер событий) класса OPC Event Server создаются клиентом. Прежде чем клиент сможет обращаться к другим объектам Alarms & Events, должен быть создан объект OPCEventServer.

Свойства объектов OPCEventServer содержат общую информацию о сервере событий. Когда создается объект, также создается собирательный объект-семейство OPCEventServerSubscription (OPC-подписка на события).

Для подключения объекта OPCEventServer к серверу событий используется метод **Connect (Соединить)**.

Свойства объекта OPCEventServer

Ниже перечислены свойства объекта OPCEventServer (OPC-сервер событий). Описаны только свойства, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Alarm & Events Automation Interface (Интерфейс автоматизации для доступа к аварийным сообщениям и событиям)

Версия 1.01

Декабрь 15, 1999

Свойство	Назначение
BuildNumber	Возвращает внутренний номер сервера.
ClientName	Указывает имя клиента.
CurrentTime	Возвращает текущее время (UTC).
FiltersByArea	Указывает, может ли сервер использовать фильтрацию по областям. OPC-сервер событий не поддерживает деление на области.
FiltersByCategory	Указывает, может ли сервер использовать фильтрацию по категориям событий.
FiltersByEventType	Указывает, может ли сервер использовать фильтрацию по типам событий.
FiltersBySeverity	Указывает, может ли сервер использовать фильтрацию по степени важности событий.
FiltersBySource	Указывает, может ли сервер использовать фильтрацию по источникам.
LastUpdateTime	Возвращает время (UTC), когда сервер последний раз передавал данные клиенту.
LocaleID	Указывает язык для отображения текстов.
MajorVersion	Возвращает основной номер версии сервера
MinorVersion	Возвращает вторичный номер версии сервера
OPCEventAreaBrowsers	Указывает семейство объектов OPCAutoEventAreaBrowser . OPC-сервер событий SIMATIC NET не поддерживает объект OPCEventAreaBrowsers .
OPCEventSubscriptions	Указывает семейство объектов OPCEventSubscription .
ServerName	Возвращает имя OPC-сервера, с которым установлено соединение.
ServerNode	Возвращает имя узла сети, в котором располагается OPC-сервер.
ServerState	Возвращает статус сервера.
StartTime	Возвращает время, когда был запущен OPC-сервер.
VendorInfo	Возвращает сведения о производителе.

Методы объекта OPCEventServer

Ниже перечислены методы объекта OPCEventServer (OPC-сервер событий). Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Alarm & Events Automation Interface (Интерфейс автоматизации для доступа к аварийным сообщениям и событиям)
Версия 1.01
Декабрь 15, 1999

Метод	Назначение
AckCondition	Подтверждает выполнение одного или нескольких условий на сервере событий.
Connect	Устанавливает соединение с OPC-сервером.
Disconnect	Разрывает соединение с OPC-сервером.
EnableConditionsByArea	Переводит все условия для всех областей (зон) в определенное состояние.
EnableConditionBySrc	Переводит все условия для всех источников в определенное состояние.
GetConditionState	Выполняет чтение текущей информации о статусе условия.
GetErrorString	Преобразует код ошибки в текстовое сообщение об ошибке.
GetOPCEventServers	Возвращает имена всех зарегистрированных серверов событий.
QueryAvailableLocaleIDs	Возвращает коды возможных языков.
QueryConditionNames	Возвращает имена условий, действительных для определенной категории событий.
QueryEventAttributes	Возвращает специальные атрибуты производителя.
QueryEventCategories	Возвращает категории, которые поддерживаются сервером событий.
QuerySourceConditions	Возвращает имена условий, связанных с определенным источником.
QuerySubConditionNames	Возвращает имена субусловий, связанных с определенным источником.

События объекта OPCEventServer

Ниже перечислены события, предусмотренные для объекта OPCEventServer (OPC-сервер событий). Описаны только события, характерные для SIMATIC NET. Описание событий можно найти в следующей спецификации OPC:

Alarm & Events Automation Interface (Интерфейс автоматизации для доступа к аварийным сообщениям и событиям)
Версия 1.01
Декабрь 15, 1999

Событие	Назначение
EventServerShutDown	Это событие запускается, когда OPC-сервер выключается. OPC-сервер SIMATIC NET запускает это событие, когда из программы конфигурирования поступает команда отключения, либо когда на ПК-станцию поступают новые конфигурационные данные.

Объект-семейство OPCEventSubscriptions

Собирательный объект OPCEventSubscriptions (OPC-подписки на события) – это семейство объектов OPCEventSubscription (OPC-подписка на события) и методов, предназначенных для создания, удаления и управления этими объектами.

Свойства объекта OPCEventSubscriptions определяют стандартные значения для всех вновь создаваемых объектов OPCEventSubscription.

Если вызов **Connect (Соединить)** объекта OPCEventServer (OPC-сервер событий) выполнен успешно, автоматически создается объект-семейство OPCEventSubscriptions.

OPCEventSubscriptions всегда существует в качестве свойства объекта OPCEventServer и используется для управления сообщениями о событиях.

Свойства объекта OPCEventSubscriptions

Ниже перечислены свойства объекта OPCEventSubscriptions (OPC-подписки на события). Описаны только свойства, характерные для SIMATIC NET. Описание свойств можно найти в следующей спецификации OPC:

Alarm & Events Automation Interface (Интерфейс автоматизации для доступа к аварийным сообщениям и событиям)
Версия 1.01
Декабрь 15, 1999

Свойство	Назначение
Count	Возвращает количество элементов
DefaultIsActive	Выбирает в качестве начального активное состояние для вновь создаваемых объектов OPCEventSubscription
DefaultbufferTime	Устанавливает первоначальное значение периодичности передачи уведомлений о событиях для вновь создаваемых объектов OPCEventSubscription
DefaultMaxSize	Устанавливает начальное значение максимального количества событий, которое может быть передано в одном уведомлении о событии, для вновь создаваемых объектов OPCEventSubscription

Методы объекта OPCEventSubscriptions

Ниже перечислены методы объекта OPCEventSubscriptions (OPC-подписки на события). Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Alarm & Events Automation Interface (Интерфейс автоматизации для доступа к аварийным сообщениям и событиям)
Версия 1.01
Декабрь 15, 1999

Метод	Назначение
Add	Создает новый объект OPCEventSubscription и добавляет его в семейство.
Item	Указывает ссылку на объект семейства с указанным индексом
Remove	Удаляет объект OPCEventSubscription
RemoveAll	Удаляет все объекты OPCEventSubscription

Объект OPCEventSubscription

Объект OPCEventSubscription (OPC-подписка на события) представляет подписку на ограниченный набор событий. Клиент запускает задание на периодическую отправку событий серверу событий.

Свойства объекта OPCEventSubscription

Ниже перечислены свойства объекта OPCEventSubscription (OPC-подписка на события). Описаны только свойства, характерные для SIMATIC NET. Описание свойств можно найти в следующей спецификации OPC:

Alarm & Events Automation Interface (Интерфейс автоматизации для доступа к аварийным сообщениям и событиям)
Версия 1.01
Декабрь 15, 1999

Свойство	Назначение
bufferTime	Определяет, как часто должны передаваться уведомления о событиях для объекта OPCEventSubscription.
IsActive	Устанавливает активное состояние объекта OPCEventSubscription
MaxSize	Определяет наибольшее количество событий, которое может быть передано в одном уведомлении о событии
Name	Указывает имя объекта OPCEventSubscription

Методы объекта OPCEventSubscription

Ниже перечислены методы объекта OPCEventSubscription. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Alarm & Events Automation Interface (Интерфейс автоматизации для доступа к аварийным сообщениям и событиям)
Версия 1.01
Декабрь 15, 1999

Метод	Назначение
GetFilter	<p>Возвращает текущий фильтр объекта OPCEventSubscription. Параметры S7 OPC-сервера событий SIMATIC NET имеют следующее назначение:</p> <p><i>EventType</i> (Тип события)</p> <p>Поддерживаются типы событий OPC_SIMPLE_EVENT (Простое событие OPC) и OPC_CONDITION_EVENT (Условное событие OPC).</p> <p><i>EventCategory</i> (Категория события)</p> <p>Категории событий описаны в разделе <i>Свойства OPC-сервера событий SIMATIC NET</i>.</p> <p><i>LowSeverity</i> (Незначительное событие)</p> <p><i>HighSeverity</i> (Значительное событие)</p> <p><i>Areas</i> (Области (Зоны))</p> <p>OPC-сервер событий не поддерживает разбиение на области (зоны).</p> <p><i>Sources</i> (Источники)</p> <p>В качестве источника можно ввести имя соединения.</p>
GetReturnedAttributes	Возвращает атрибуты для каждой категории событий, которые сервер передает вместе с уведомлениями о событиях.
Refresh	Обновляет все соединения.
RefreshCancel	Отменяет выполнение метода Refresh (Обновить). S7 OPC-сервер событий SIMATIC NET не поддерживает метод RefreshCancel (Отменить обновление).
SelectReturnedAttributes	Для каждой категории события этот метод указывает атрибуты, которые возвращаются вместе с уведомлением о событии методом OnEvent (По событию)

Метод	Назначение
SetFilter	<p>Устанавливает все фильтры таким образом, чтобы их свойства совпадали со свойствами созданных событий. Параметры S7 OPC-сервера события SIMATIC NET имеют следующее назначение:</p> <p><i>EventType (Тип события)</i></p> <p>Поддерживаются типы событий OPC_SIMPLE_EVENT (Простое событие OPC) и OPC_CONDITION_EVENT (Условное событие OPC).</p> <p><i>EventCategory (Категория события)</i> S7_PROCESS_ALARM</p> <p><i>LowSeverity (Незначительное событие)</i></p> <p><i>HighSeverity (Значительное событие)</i></p> <p><i>Areas (Области (Зоны))</i></p> <p>OPC-сервер событий не поддерживает разбиение на области (зоны).</p> <p><i>Sources (Источники)</i></p> <p>В качестве источника можно ввести имя соединения.</p>

События объекта OPCEventSubscription

Ниже перечислен список событий, предусмотренных для объекта OPCEventSubscription (OPC-подписка на события). Описаны только события, характерные для SIMATIC NET. Описание событий можно найти в следующей спецификации OPC:

Alarm & Events Automation Interface (Интерфейс автоматизации для доступа к аварийным сообщениям и событиям)
Версия 1.01
Декабрь 15, 1999

Событие	Назначение
ConditionEvent	Возникает, когда сервер отправляет уведомления о событиях, возникающих по условию.
RefreshCancel	Возникает, когда прерывается метод Refresh (Обновить) . OPC-сервер событий SIMATIC NET не поддерживает метод RefreshCancel (Отменить обновление) .

Событие	Назначение
RefreshComplete	Возникает, когда завершается метод Refresh (Обновить) . OPC-сервер событий SIMATIC NET не поддерживает метод RefreshCancel (Отменить обновление) .
RefreshConditionEvent	Возникает, когда сервер отправляет уведомления о событиях, связанных с обновлением условий.
SimpleEvent	Возникает, когда сервер отправляет группу простых событий.
TrackingEvent	Возникает, когда сервер отправляет события, связанные со слежением. OPC-сервер событий SIMATIC NET не поддерживает TrackingEvent .

Объект-семейство OPCAutoEventAreaBrowsers

Собирательный объект OPCAutoEventAreaBrowsers (OPC-браузеры области событий) – это семейство объектов OPCAutoEventAreaBrowser (OPC-браузер области событий) и методов, предназначенных для создания, удаления и управления этими объектами.

Примечание

OPC-сервер событий не поддерживает объект OPCAutoEventAreaBrowsers.

Объект OPCAutoEventAreaBrowser

С помощью объекта OPCAutoEventAreaBrowser (OPC-браузер области событий) клиент может просматривать области и источники, находящиеся на сервере. Области группируются в единый объект OPCEventAreas (OPC-области событий), а источники формируют объект OPCEventSources (OPC-источники событий).

Области и источники можно использовать для фильтрации событий.

Пространство имен сервера

Серверы могут иметь простое или иерархическое пространство имен. Если пространство имен является простым, в этом случае объекты OPCEventAreas и OPCEventSources обладают одними и теми же областями и источниками на сервере.

Если пространство имен имеет иерархическую структуру, области можно принимать за ветви дерева, а источники – за его листья (объекты на ветвях дерева). Объект OPCAutoEventAreaBrowser выступает в роли указателя, который перемещается вдоль областей. Объекты OPCEventAreas и OPCEventSources содержат только области и источники, в которых существует объект.

Примечание

OPC-сервер событий не поддерживает объект OPCAutoEventAreaBrowser.

Объект-семейство OPCEvents

Собирательный объект OPCEvents (OPC-события) – это параметр, передаваемый при вызове методов для обработки событий. Он содержит семейство объектов OPCEvent и методов, предназначенных для создания, удаления и управления этими объектами.

Объект OPCEvents формируется из уведомлений о событиях, передаваемых сервером, то есть, он доставляет произошедшие события.

Свойства объекта OPCEvents

Ниже перечислены свойства объекта OPCEvents (OPC-события). Описаны только свойства, характерные для SIMATIC NET. Описание свойств можно найти в следующей спецификации OPC:

Alarm & Events Automation Interface (Интерфейс автоматизации для доступа к аварийным сообщениям и событиям)
Версия 1.01
Декабрь 15, 1999

Свойство	Назначение
Count	Возвращает количество элементов
LastRefresh	Указывает, что объект-семейство OPCEvents является последним в последовательности уведомлений об обновлениях
Refresh	Указывает, что объект-семейство OPCEvents принадлежит уведомлению об обновлении

Методы объекта OPCEvents

Ниже перечислены методы объекта OPCEvents (OPC-события). Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Alarm & Events Automation Interface (Интерфейс автоматизации для доступа к аварийным сообщениям и событиям)
Версия 1.01
Декабрь 15, 1999

Метод	Назначение
Add	Создает новый объект OPCEvent (OPC-событие) и добавляет его в семейство
Item	Указывает имя элемента

Объект OPCEvent

Объект OPCEvent (OPC-событие) содержит уведомление об определенном событии.

Свойства объекта OPCEvent

Ниже перечислены свойства объекта OPCEvent (OPC-событие). Описаны только свойства, характерные для SIMATIC NET. Описание свойств можно найти в следующей спецификации OPC:

Alarm & Events Automation Interface (Интерфейс автоматизации для доступа к аварийным сообщениям и событиям)

Версия 1.01

Декабрь 15, 1999

Свойство	Назначение
AckRequired	Указывает, что для условия требуется подтверждение.
ActiveTime	Возвращает время, когда было активизировано условие.
ActorID	Возвращает события, связанные со слежением и выполнением условий, для которых требуется подтверждение.
ChangeAckState	Возвращает уведомление о событии в случае изменения свойства Acknowledge условия.
ChangeActiveState	Возвращает уведомление о событии в случае изменения свойства ActiveState условия.
ChangeAttribute	Возвращает уведомление о событии в случае изменения свойства Attribute условия.
ChangeEnableState	OPC-сервер событий SIMATIC NET не поддерживает свойство Enable условия
ChangeMessage	Возвращает уведомление о событии в случае изменения свойства Message условия. OPC-сервер событий SIMATIC NET не поддерживает ChangeMessage .
ChangeQuality	Возвращает уведомление о событии в случае изменения свойства Quality условия OPC-сервер событий SIMATIC NET не поддерживает ChangeQuality .

Свойство	Назначение
ChangeSeverity	Возвращает уведомление о событии в случае изменения свойства Severity условия. OPC-сервер событий SIMATIC NET не поддерживает ChangeSeverity .
ChangeSubCondition	Возвращает уведомление о событии в случае изменения свойства SubCondition условия.
ConditionAcknowledged	Указывает, что условие получило новый статус Acknowledged .
ConditionActive	Указывает, что условие получило новый статус Active .
ConditionName	Возвращает имя условия, которое ссылается на уведомление о событии.
Cookie	Возвращает "Cookie", определенное сервером и связанное с уведомлением о событии. OPC-сервер событий SIMATIC NET не поддерживает Cookie .
EventCategory	Возвращает код категории события.
Message	Возвращает текст описания уведомления.
OPCEventAttributes	Возвращает специальные атрибуты события, установленные производителем, возвращаемые в уведомлении о событии.
Quality	Возвращает код качества, соответствующий состоянию условия. OPC-сервер событий SIMATIC NET не поддерживает Quality .
Severity	Возвращает степень важности события.
Source	Возвращает источник уведомления о событии.
SubConditionName	Возвращает имя текущего субусловия. OPC-сервер событий SIMATIC NET не поддерживает SubConditionName .
Time	Возвращает время (UTC), в которое произошло событие.

Объект OPCEventCondition

Объект OPCEventCondition (OPC-событие возникновения условия) описывает текущее состояние условия.

Объект-семейство OPCEventSubConditions

Собираемый объект OPCEventSubConditions (OPC-субусловия возникновения события) – это семейство объектов OPCEventSubCondition (OPC-субусловие возникновения события). Семейство состоит из различных состояний условия, которые могут наблюдаться на OPC-сервере событий.

Объект OPCEventSubCondition

Объект OPCEventSubCondition (OPC-субусловие события) представляет специальное субусловие OPC-сервера событий. Объект OPCEventSubCondition содержит атрибуты субусловия.

6.3 Программирование пользовательского интерфейса

Пользовательский интерфейс (Custom Interface) сконструирован таким образом, чтобы его функции идеально подходили для выполнения предъявленных требований. Он не подходит для реализации доступа к данным с применением языков описания сценариев. Для этих целей предназначен интерфейс автоматизации (Automation Interface).

Пользовательский интерфейс можно применять как для доступа к переменным процесса (data access), так и для обработки событий и аварийных сообщений (Alarms & Events).

Применение

Пользовательский интерфейс применяется в тех случаях, когда необходимо создать приложение, оперирующее большим количеством переменных и требующее большой пропускной способности, с применением языка C++.

CLSID

Каждому COM-классу может быть присвоен уникальный идентификационный код. Этот код, состоящий из 128-ми битов, называется идентификатором CLSID. С помощью CLSID операционная система может обнаружить библиотеку DLL или файл EXE, посредством которых реализован COM-класс. Если клиенту требуется использовать объект класса, он ссылается на него, используя CLSID.

ProgID

Чтобы упростить идентификацию OPC-серверов, идентификаторам CLSID назначаются читабельные идентификаторы ProgID. Идентификаторы CLSID и ProgID назначаются изготовителем OPC-сервера.

Ниже приведены сведения, которые относятся ко всем протоколам, применяемым для OPC-сервера SIMATIC NET:

Интерфейс	ProgID
Data Access (Доступ к данным)	OPC.SimaticNET (стандартный OPC-сервер для работы с несколькими протоколами) OPC.SimaticNET (Высокопроизводительный OPC-сервер для работы с одним протоколом S7, SR или DP. Также допускается применение символов и DX) OPC.SimaticNET.DP (OPC-сервер для высокопроизводительного доступа с помощью протокола DP) OPC.SimaticNet.PD (сервер шины PROFIDrive)
Alarms & Events для S7 (Аварийные сообщения и события)	OPC.SimaticNetAlarms (A&E-сервер для S7) OPC.SimaticNetAlarms (Высокопроизводительный A&E-сервер для работы только с протоколом S7. Также допускается использовать символы и DX) OPC.SimaticNetAlarmsSNMP (A&E-сервер для SNMP)

Действия

Пользователь создает COM-объект и вызывает его методы.

Примечание

Объект Windows – это экземпляр COM-класса. COM-класс отличается от классов, применяемых в C++. В C++ класс – это определение типа. COM-класс – это описание объекта, которое не содержит типов.

6.3.1 Создание COM-объектов и чтение статуса OPC-сервера

Подробное описание применения пользовательского интерфейса для OPC Data Access можно найти в пояснениях к примерам. Ниже приводится лишь описание основной последовательности действий, не предусматривающей обработку ошибок.

Пользователь создает COM-объект и опрашивает его статус за пять действий.

Примеры создания COM-объекта на языке Visual C++ смотрите в разделах 9.4 и 9.5.

6.3.2 Объекты пользовательского интерфейса для доступа к данным

В этом разделе перечислены интерфейсы объектов и их методы. Описаны только интерфейсы, характерные для SIMATIC NET. Подробное описание интерфейсов можно найти в соответствующих спецификациях OPC.

Возвращаемые значения методов интерфейсов

Все методы возвращают результат с типом **HResult**.

6.3.3 Объект OPCServer

Объект класса OPC server (OPC-сервер) обладает рядом атрибутов, которые содержат, к примеру, сведения о состоянии или номер версии объекта OPCServer. Класс также обладает методами, посредством которых клиент может управлять объектами класса OPC Group (OPC-группа).

Клиентское приложение использует объект OPC-сервер только напрямую, используя механизмы COM. Все остальные объекты создаются посредством соответствующих методов OPC.

Интерфейсы объекта OPCServer

На рисунке ниже показаны интерфейсы объекта OPCServer.

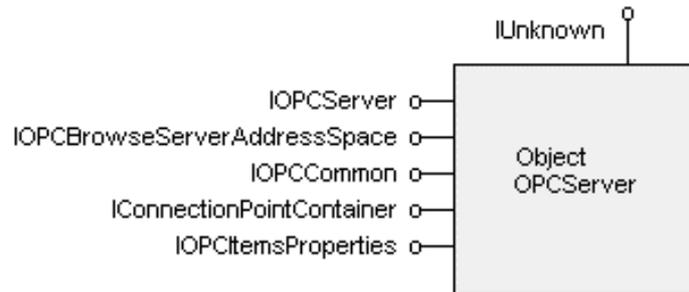


Рисунок 6–16 Объект OPCServer

Интерфейс IOPCServer

Интерфейс IOPCServer содержит методы, предназначенные для управления объектами класса OPC Group (OPC-группа). Также через него можно получить информацию о текущем статусе сервера.

Ниже приведен список методов, предусмотренных для интерфейса IOPCServer. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Custom Interface
 (Пользовательский интерфейс для доступа к данным)
 Версия 2.05
 Декабрь 17, 2001

Метод	Назначение
AddGroup	<p>Создает группу в объекте "Сервер".</p> <p>Параметр LCID не анализируется OPC-сервером SIMATIC NET.</p> <p>Параметр pTimeBias определяет разницу во времени (между UTC и местным часовым поясом).</p> <p>Скорости обновления, используемые OPC-сервером SIMATIC NET, являются кратными значениями длительности цикла, выбранной на этапе конфигурирования. Минимальная скорость обновления совпадает с длительностью цикла.</p> <p>Если параметр szName является пустым, генерируется имя, которое начинается с тильды (например, ~GROUP_1). Следовательно, имена, устанавливаемые пользователем, не должны начинаться с тильды (~).</p>

Метод	Назначение
CreateGroupEnumerator	Создает различные нумераторы для группы. OPC-сервер SIMATIC NET не поддерживает общие группы (public group). Таким образом, возвращаемые значения для методов должны полностью соответствовать вводимым значениям ...PRIVATE и ...PUBLIC (для общих групп) для параметра dwScope .
GetErrorString	Чтение сообщения об ошибке для соответствующего кода ошибки. OPC-сервер SIMATIC NET поддерживает отображение сообщений об ошибках на немецком и английском языках. Сообщения об ошибках операционной системы Windows отображаются на языке, который используется операционной системой.
GetGroupByName	Возвращает дополнительный указатель на имя закрытой группы (private group); счетчик обращений увеличивается на 1.
GetStatus	Возвращает информацию о статусе сервера. Возвращается имя и номер версии OPC-сервера.
RemoveGroup	Удаляет группу, имеющуюся на сервере. OPC-сервер SIMATIC NET не поддерживает bForce . Нельзя удалять группы, если по-прежнему существуют ссылки на них.

Интерфейс IOPCServerPublicGroups

OPC-сервер SIMATIC NET не поддерживает общие группы (public groups) и, следовательно, не поддерживает дополнительный интерфейс IOPCServerPublicGroups.

Интерфейс IOPCBrowseServerAddressSpace

С помощью дополнительного интерфейса IOPCBrowseServerAddressSpace можно просматривать пространство имен сервера. Пространство имен содержит все OPC-элементы, которые известны серверу.

Ниже приведен список методов, предусмотренных для интерфейса IOPCBrowseServerAddressSpace. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Custom Interface
(Пользовательский интерфейс для доступа к данным)
Версия 2.05
Декабрь 17, 2001

Метод	Назначение
BrowseAccessPaths	Выполняет чтение пути доступа для элемента с указанным ItemID. Метод BrowseAccessPaths не требуется для OPC-сервера SIMATIC NET.
BrowseOPCItemIDs	<p>Возвращает строку типа IEnumString, содержание которой определяется параметрами вызова.</p> <p>Если для параметра dwBrowseFilterType введено значение OPC-BRANCH, параметры vtDataTypeFilter и dwAccessRightsFilter не имеют значения.</p> <p>Применяются следующие правила для создания фильтра:</p> <ul style="list-style-type: none"> * любая строка символов, включая пустые строки + любая строка символов, состоящая хотя бы из одного символа ? ровно один символ [] Ровно один символ из указанного набора. Если используется символ фильтрации, перед ним необходимо размещать обратный слеш (\).
ChangeBrowsePosition	Переход на верхний уровень или верхнюю ветвь в пространстве имен
GetItemID	Выполняет чтение полностью всего идентификатора (ItemID) в иерархическом пространстве имен. OPC-сервер SIMATIC NET поддерживает метод GetItemID только для одиночных объектов ("листьев").
QueryOrganization	Возвращает структуру пространства имен. Пространство имен OPC-сервера SIMATIC NET имеет иерархическую структуру.

Интерфейс IOPCItemProperties

Интерфейс IOPCItemProperties содержит методы, предназначенные для запроса специальной информации сервера об элементе.

Ниже перечислены методы, предусмотренные для интерфейса IOPCItemProperties. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Custom Interface (Пользовательский интерфейс для доступа к данным)
Версия 2.05
Декабрь 17, 2001

Метод	Назначение
QueryAvailableProperties	Возвращает список имеющихся свойств элемента
GetItemProperties	Возвращает значения свойств элемента, для которых были переданы идентификаторы свойств (PropertyID)
LookupItemIDs	Возвращает список идентификаторов ItemID для списка идентификаторов PropertyID

Интерфейс IConnectionPointContainer

Интерфейс IConnectionPointContainer является стандартным COM-интерфейсом, который предназначен для передачи асинхронных событий через точки соединений. Более подробную информацию по использованию точек соединений можно найти в соответствующей литературе по COM.

Интерфейс IOPCCommon

Интерфейс IOPCCommon содержит методы, предназначенные для информирования сервера о параметрах языка и имени клиента.

Ниже перечислены методы, предусмотренные для интерфейса IOPCCommon. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Custom Interface
(Пользовательский интерфейс для доступа к данным)
Версия 2.05
Декабрь 17, 2001

Метод	Назначение
SetLocaleID	Устанавливает код языка для сервера. OPC-сервер SIMATIC NET поддерживает немецкий и английский языки.
GetLocaleID	Выполняет чтение кода языка сервера. OPC-сервер SIMATIC NET поддерживает немецкий и английский языки.
QueryAvailableLocaleIDs	Возвращает коды всех имеющихся языков сервера. OPC-сервер SIMATIC NET поддерживает немецкий и английский языки.
GetErrorString	Возвращает текст ошибки для кода ошибки
SetClientName	Передаёт серверу текст описания клиента

Интерфейс IPersistFile

Дополнительный интерфейс IPersistFile не поддерживается OPC-сервером SIMATIC NET.

6.3.4 Объект OPCGroup

Объект класса OPC Group (OPC-группа) управляет отдельными переменными процесса - OPC-элементами. С помощью объекта OPCGroup (OPC-группа) клиент может объединять отдельные OPC-элементы на основании определенных правил и признаков и выполнять над ними операции.

Интерфейсы объекты OPCGroup

На следующем рисунке показаны интерфейсы объектов OPCGroup.

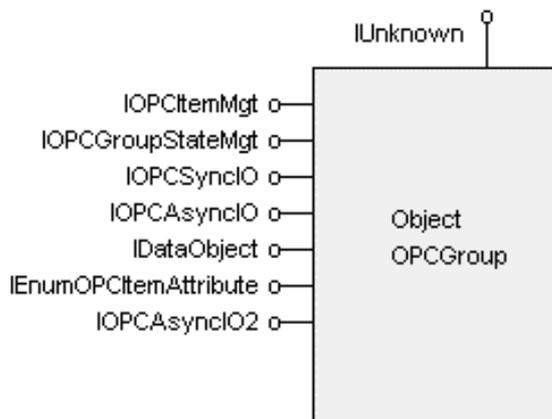


Рисунок 6–17 Объект OPCGroup

Интерфейс IOPCGroupStateMgt

Интерфейс IOPCGroupStateMgt предоставляет методы для управления группами. Он позволяет редактировать специальные параметры группы и копировать группы.

Ниже перечислены методы, предусмотренные для интерфейса IOPCGroupStateMgt. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Custom Interface
(Пользовательский интерфейс для доступа к данным)
Версия 2.05
Декабрь 17, 2001

Метод	Назначение
CloneGroup	Создает копию группы. При копировании атрибутов группы выполняются следующие правила: Устанавливается неактивное состояние. Назначается новый дескриптор сервера. szName может быть пустым. В этом случае генерируется уникальное имя.
GetState	Выполняет чтение статуса группы. Параметры pTimeBias и pLCID не имеют значения для OPC-сервера SIMATIC NET.
SetName	Изменяет имя группы
SetState	Изменяет свойства группы. Параметры pTimeBias и pLCID не имеют значения для OPC-сервера SIMATIC NET. Скорости обновления, используемые OPC-сервером SIMATIC NET, являются кратными значениями длительности цикла, указанной на этапе конфигурирования. Минимальная скорость обновления соответствует длительности цикла.

Интерфейс IOCPublicGroupStateMgt

OPC-сервер SIMATIC NET не поддерживает общие группы (public group). Поэтому дополнительный интерфейс IOCPublicGroupStateMgt не используется.

Интерфейс IOPCAsyncIO2

Интерфейс IOPCAsyncIO2 содержит методы для асинхронного чтения и записи элементов.

Асинхронность означает, что клиент инициирует операции чтения или записи, после чего продолжает выполнение. Для интерфейса используются точки соединений. Это упрощает обработку передаваемых данных.

Каждое прочитанное значение снабжается в OPC меткой времени. Поскольку системы SIMATIC не управляют метками времени, в качестве метки времени используется время, в которое значение было принято сервером.

Ниже перечислены методы, предусмотренные для интерфейса IOPCAsyncIO2. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Custom Interface

(Пользовательский интерфейс для доступа к данным)

Версия 2.05

Декабрь 17, 2001

Метод	Назначение
Read	Передает команду асинхронного чтения. На сервере контролируется превышение времени для вызова. Если установленное при конфигурировании время превышено, поступает уведомление со статусом E_ABORT .
Write	Передает команду асинхронной записи. На сервере контролируется превышение времени для вызова. Если установленное при конфигурировании время превышено, поступает уведомление со статусом E_ABORT .
Cancel2	Отменяет стоящее в очереди задание.
Refresh	Запрашивает текущее значение из кеша для каждого активного элемента
SetEnable	Разрешает отключение уведомления с использованием OnDataChange
GetEnable	Возвращает текущее значение для уведомления с использованием OnDataChange

Интерфейс IOPCAsyncIO

Этот интерфейс содержит методы для асинхронного чтения и записи элементов. Асинхронность означает, что клиент инициирует операцию чтения или записи, после чего продолжает работу.

Асинхронные операции возвращают идентификатор операции (transaction ID). Когда сервер завершает операцию чтения или записи, клиент уведомляется об этом через свой интерфейс IAdviseSink.

Примечание

Интерфейс IOPCAsyncIO в версии 2 был заменен интерфейсом IOPCAsyncIO2. Интерфейс IOPCAsyncIO2 использует точки соединений и работать с ним легче. В своих дальнейших проектах используйте IOPCAsyncIO2.

Каждое прочитанное значение снабжается в OPC меткой времени. Поскольку системы SIMATIC не управляют метками времени, в качестве метки времени используется время, в которое значение было принято сервером.

Ниже перечислены методы, предусмотренные для интерфейса IOPCAsyncIO. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Custom Interface
(Пользовательский интерфейс для доступа к данным)
Версия 2.05
Декабрь 17, 2001

Метод	Назначение
Cancel	Отменяет задание, стоящее в очереди.
Read	Передает команду асинхронного чтения. На сервере контролируется превышение времени для вызова. Время конфигурируется параметром Read/Write Timeout . Если это время превышено, в ответ возвращается вызов со статусом E_ABORT .
Refresh	Запрашивает текущее значение для каждого активного OPC-элемента.
Write	Передает команду асинхронной записи. На сервере контролируется превышение времени для вызова. Время конфигурируется параметром Read/Write Timeout . Если это время превышено, в ответ возвращается вызов с hrStatus=E_ABORT .

Интерфейс IOPCItemMgt

Интерфейс IOPCItemMgt содержит методы, предназначенные для управления несколькими элементами группы.

Ниже перечислены методы, предусмотренные для интерфейса IOPCItemMgt. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Custom Interface
(Пользовательский интерфейс для доступа к данным)
Версия 2.05
Декабрь 17, 2001

Метод	Назначение
AddItems	Добавляет один или несколько элементов в группу.
CreateEnumerator	Создает нумератор. Это объект EnumOPCItemAttributes.
RemoveItems	Удаляет один или несколько элементов из группы.
SetActiveState	Устанавливает активное состояние для одного или нескольких элементов группы.
SetClientHandles	Устанавливает дескриптор клиента для одного или нескольких элементов группы.
SetDataTypes	Устанавливает требуемый тип данных для одного или нескольких элементов группы.
ValidateItems	Проверяет действительность OPC-элемента.

Интерфейс IOPCSyncIO

Интерфейс IOPCSyncIO содержит методы, предназначенные для синхронного чтения или записи. Синхронность означает, что клиент дожидается завершения операции чтения или записи, прежде чем продолжить работу. Клиент может использовать для дальнейшей работы результат операции чтения или записи.

Поскольку OPC-сервер SIMATIC NET запускает для каждого клиента отдельную ветвь, работа других клиентов не останавливается, пока клиент ожидает результаты.

Ниже перечислены методы, предусмотренные для интерфейса IOPCSyncIO. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Custom Interface
(Пользовательский интерфейс для доступа к данным)
Версия 2.05
Декабрь 17, 2001

Метод	Назначение
Read	Читает значение и информацию о статусе, а также метку времени для одного или нескольких элементов. На сервере контролируется превышение времени вызова. Для каждого отдельного протокола на этапе конфигурирования задается соответствующее значение превышения времени.
Write	Записывает значение в один или несколько элементов группы. На сервере контролируется превышение времени для вызова. Для каждого отдельного протокола на этапе конфигурирования устанавливается соответствующее значение превышения времени.

Интерфейс IDataObject

Интерфейс IDataObject является стандартным COM-интерфейсом для передачи данных. Он содержит методы, предназначенные для установления соединения между клиентом и группой сервера, служащего для передачи уведомления.

Когда сервер передает клиенту уведомление, он обращается к клиенту через интерфейс клиента IAdviseSink, вызывая метод **OnDataChange** интерфейса IAdviseSink.

На следующем рисунке показано взаимодействие между интерфейсом IAdviseSink клиента и интерфейсом IDataObject сервера.

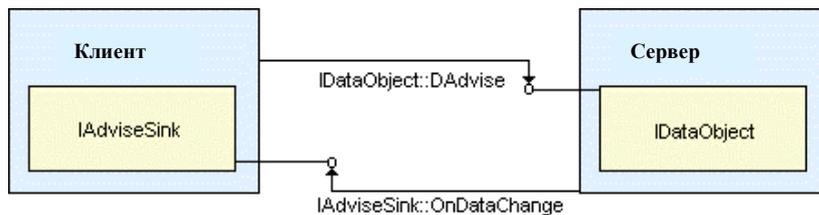


Рисунок 6–18 Взаимодействие между IAdviseSink и IDataObject

Примечание

Интерфейс IDataObject использовался в интерфейсе доступа к данным (Data Access) версии 1 для асинхронных коммуникаций. Начиная с версии 2, используются точки соединений, которые более просты и гибки в использовании. В своих последующих проектах используйте версию 2.

Ниже перечислены методы, предусмотренные для интерфейса IDataObject. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Custom Interface
 (Пользовательский интерфейс для доступа к данным)
 Версия 2.05
 Декабрь 17, 2001

Метод	Назначение
DAdvise	Устанавливает соединение между сервером и клиентом.
DUnadvise	Разрывает соединение между сервером и клиентом.

Интерфейс IEnumOPCItemAttributes

Интерфейс IEnumOPCItemAttributes базируется на стандартном интерфейсе IEnum. Он возвращает элементы группы. Интерфейс предоставляется методом **CreateEnumerator** интерфейса **IOPCItemMgt**. Его нельзя получить с помощью **QueryInterface**.

Ниже перечислены методы, предусмотренные для интерфейса IEnumOPCItemAttributes. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Data Access Custom Interface
(Пользовательский интерфейс для доступа к данным)
Версия 2.05
Декабрь 17, 2001

Метод	Назначение
Clone	Создает идентичную копию объекта IEnumOPCItemAttributes
Next	Считывает следующий OPC-элемент группы.
Reset	Переводит список к первому элементу группы
Skip	Пропускает указанное количество элементов в списке

6.3.5 Объекты пользовательского интерфейса для доступа к аварийным сообщениям и событиям

Интерфейсы и предусмотренные для них методы перечислены ниже. Описаны только интерфейсы, характерные для SIMATIC NET. Описание интерфейсов можно найти в соответствующих спецификациях OPC:

Возвращаемые значения методов интерфейсов

Все методы возвращают результат типа **HResult**.

6.3.6 Объект OPCEventServer

Объект класса OPC Event Server (OPC-сервер событий) создается клиентом. Клиент использует этот объект, когда ему требуются сервисы Alarms & Events (Аварийные сообщения и события).

С помощью объекта OPCEventServer можно выполнять следующие задачи:

- Создание объекта "Сообщение"
- Выполнение запросов
- Формирование событий
- Выбор языка для отображения текстов
- Подписка на получение специальных событий сервера

Интерфейсы объекта OPCEventServer

На следующем рисунке показаны интерфейсы объекта OPCEventServer.

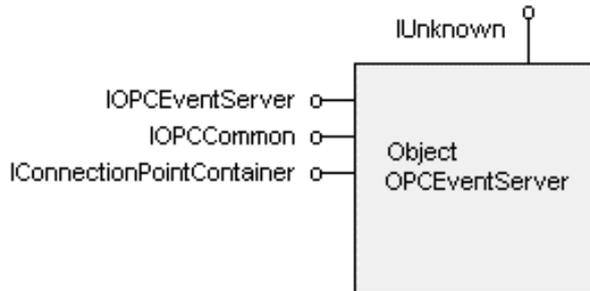


Рисунок 6–19 Объект OPCEventServer

Интерфейс IOPCCommon

Интерфейс IOPCCommon содержит методы, предназначенные для информирования сервера о параметрах языка и имени клиента.

Ниже перечислены методы, предусмотренные для интерфейса IOPCCommon. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Alarms & Events Custom Interface (Пользовательский интерфейс для доступа к аварийным сообщениям и событиям)
 Версия 1.10
 Октябрь 2, 2002

Метод	Назначение
SetLocaleID	Устанавливает код языка для сервера. OPC-сервер SIMATIC NET поддерживает немецкий и английский языки.
GetLocaleID	Выполняет чтение кодов языка для сервера. OPC-сервер SIMATIC NET поддерживает немецкий и английский языки.
QueryAvailableLocaleIDs	Возвращает коды всех имеющихся языков сервера. OPC-сервер SIMATIC NET поддерживает немецкий и английский языки.
GetErrorString	Выполняет чтение сообщения об ошибке для кода ошибки.
SetClientName	Передает серверу текст описания клиента.

Интерфейс IOPCEventServer

Интерфейс IOPCEventServer является основным интерфейсом для доступа к аварийным сообщениям и событиям (Alarms & Events). Он позволяет выполнять следующие задачи:

- Создание объектов "Подписка"
- Создание браузеров области
- Просмотр категорий событий
- Управление условиями

Примечание

SIMATIC NET не поддерживает дополнительные области (зоны), поэтому браузер областей не создается.

Ниже перечислены методы, предусмотренные для интерфейса IOPCEventServer. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Alarms & Events Custom Interface (Пользовательский интерфейс для доступа к аварийным сообщениям и событиям)
Версия 1.10
Октябрь 2, 2002

Метод	Назначение
GetStatus	Выполняет чтение информации о текущем статусе OPC-сервера.
CreateEventSubscription	Создает объект "Сообщение" для уведомления клиента. Объект "Сообщение" является "подпиской" на набор определенных событий. В ответ возвращается интерфейс, необходимый для обращения к объекту "Сообщение".
QueryAvailableFilters	Возвращает информацию об опциях фильтра, поддерживаемых сервером событий. OPC-сервер событий SIMATIC NET не поддерживает следующие фильтры: OPC_FILTER_BY_EVENTS: 0x01 OPC_FILTER_BY_CATEGORY: 0x02 OPC_FILTER_BY_SEVERITY: 0x04 OPC_FILTER_BY_SOURCE: 0x16

Метод	Назначение
QueryEventCategories	Возвращает категории событий, поддерживаемые сервером событий. OPC-сервер событий SIMATIC NET возвращает категории события только тогда, когда для параметра dwEventType выбрано значение OPC_SIMPLE_EVENT (Простое событие OPC) или OPC_CONDITION_EVENT (Условное событие OPC) .
QueryConditionNames	Возвращает условия, поддерживаемые сервером событий для определенной категории события.
QuerySubConditionNames	Возвращает субусловия, поддерживаемые сервером событий для определенной категории события.
QuerySourceConditions	Возвращает условия, поддерживаемые OPC-сервером событий для определенного источника.
QueryEventAttributes	Возвращает атрибуты, поддерживаемые сервером событий для определенной категории событий. OPC-сервер событий SIMATIC NET поддерживает специальные атрибуты. Атрибуты нельзя использовать в качестве идентификаторов элементов (temID) для доступа к данным.
TranslateToItemIDs	Выполняет чтение OPC-элементов, соответствующих атрибуту события, которое предназначено для использования с соответствующим OPC-сервером для доступа к данным (OPC Data Access Server). OPC-сервер событий (OPC Event Server) SIMATIC NET не поддерживает метод TranslateToItemIDs .
GetConditionState	Возвращает сведения о статусе условия источника.
EnableConditionByArea	Активирует все условия для всех источников внутри указанной области.
EnableConditionBySource	Активирует все условия для всех указанных источников.
DisableConditionByArea	Деактивирует все условия для всех источников внутри указанной области. OPC-сервер событий SIMATIC NET не поддерживает условия. Возвращается E_NOTIMPL .
DisableConditionBySource	Деактивирует все условия для всех указанных источников.

Метод	Назначение
AckCondition	Передает подтверждение события клиенту. Подтверждаются только события, формируемые по выполнению условий (условные события).
CreateAreaBrowser	Создает объект OPCEventAreaBrowser (OPC-браузер области событий) для просмотра пространства процесса OPC-сервер событий SIMATIC NET не поддерживает области (зоны). Возвращается E_NOTIMPL .

Интерфейс IConnectionPointContainer

Интерфейс IConnectionPointContainer является стандартным COM-интерфейсом, предназначенным для сообщения об асинхронных событиях через точки соединений. Более подробные сведения об использовании точек соединений можно найти в справочной литературе по COM.

6.3.7 Объект OPCEventSubscription

Объект класса OPC Event Subscription (OPC-подписка на события) передает сообщения о событиях клиенту, который использует интерфейс IConnectionPointContainer этого объекта.

Клиент может использовать несколько объектов OPCEventSubscription. Он может устанавливать различные критерии фильтрации для различных объектов.

Объект OPCEventSubscription является "подпиской" на определенные события.

Интерфейсы объекта OPCEventSubscription

На следующем рисунке показаны интерфейсы объекта OPCEventSubscription.

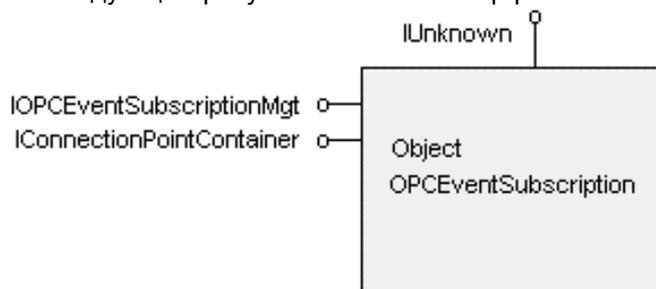


Рисунок 6–20 Объект OPCEventSubscription

Интерфейс IOPCEventSubscriptionMgt

Интерфейс IOPCEventSubscriptionMgt является главным интерфейсом, предназначенным для управления информацией о подписке на определенные события. Через этот интерфейс, например, можно выбирать события, относящиеся к клиенту.

Ниже перечислены методы, предусмотренные для интерфейса IOPCEventSubscriptionMgt. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Alarms & Events Custom Interface (Пользовательский интерфейс для доступа к аварийным сообщениям и событиям)
Версия 1.10
Октябрь 2, 2002

Метод	Назначение
SetFilter	<p>Устанавливает критерии фильтрации для выбора определенных событий для данной подписки на события. Параметры фильтра имеют следующие значения для OPC-сервера событий SIMATIC NET:</p> <p><i>Event Type (Тип события)</i> OPC-сервер событий поддерживает следующие типы событий OPC_SIMPLE_EVENT (Простое событие OPC) и OPC_CONDITION_EVENT (Условное событие OPC).</p> <p><i>Event Categories (Категории событий)</i> Категории событий описаны в разделе <i>Свойства OPC-сервера событий SIMATIC NET</i>.</p> <p><i>Severity (Степень важности)</i> Степень важности можно сконфигурировать в STEP 7 или в SIMATIC NCM PC/S7. Например, для S7_PROCESS_ALARM по умолчанию используется значение 500.</p> <p><i>Areas (Области (Зоны))</i> OPC-сервер событий SIMATIC NET не поддерживает области (зоны).</p> <p><i>Source (Источник)</i> Можно ввести имя соединения.</p>

Метод	Назначение
GetFilter	Возвращает используемые в настоящее время фильтры подписки на события. См. SetFilter .
SelectReturnedAttributes	Указывает атрибуты, возвращаемые вместе с сообщением о событии, для категории события.
GetReturnedAttributes	Возвращает перечень атрибутов, возвращаемых вместе с сообщением о событии для категории события.
Refresh	Передает все активные и все неактивные, неподтвержденные сообщения, формируемые по выполнению условий, которые соответствуют текущему фильтру клиента.
CancelRefresh	Отменяет выполнение текущего обновления. Поскольку никаких сообщений о событиях не передается для Refresh , метод CancelRefresh не действует.
GetState	Возвращает текущий статус подписки на события.
SetState	Устанавливает различные свойства подписки на события.

Интерфейс IConnectionPointContainer

Интерфейс IConnectionPointContainer является стандартным COM-интерфейсом, предназначенным для передачи сообщений об асинхронных событиях через точки соединений. Более подробные сведения об использовании точек соединений можно найти в соответствующей литературе по COM.

6.3.8 Объект OPCEventAreaBrowser

OPC-сервер событий (OPC Event Server) не поддерживает области (зоны), поэтому объект OPCEventAreaBrowser (OPC-браузер областей событий) использовать нельзя.

6.3.9 Интерфейсы клиента для доступа к аварийным сообщениям и событиям

Уведомление клиента о событиях

Уведомление клиента о событиях осуществляется через точки соединений. Для этого клиент должен предоставить COM-объект с интерфейсом IUnknown (Неизвестный интерфейс) и специальным интерфейсом IOPCEventSink, предназначенным для приема вызова. Когда клиент регистрируется в точке соединения, он передает серверу указатель на интерфейс IUnknown.

Интерфейсы объекта клиента

Объект, предоставляемый клиентом для приема сообщений, должен иметь показанную ниже структуру:

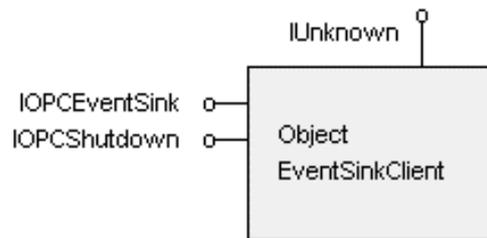


Рисунок 6–21 Интерфейсы клиента

6.3.10 Интерфейс IOPCEventSink

Интерфейс IOPCEventSink является главным интерфейсом клиента для приема сообщений. Он содержит один метод, который вызывается сервером для передачи сообщений.

Ниже приведен список методов, предусмотренных для интерфейса IOPCEventSink. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Alarms & Events Custom Interface (Пользовательский интерфейс для доступа к аварийным сообщениям и событиям)
 Версия 1.10
 Октябрь 2, 2002

Метод	Назначение
OnEvent	<p>Передает одно или несколько сообщений о событиях клиенту.</p> <p>Структура *pEvents содержит одно или несколько событий. OPC-сервер событий (OPC Event Server) вводит определенные значения в следующие элементы структуры:</p> <p><i>SzSource</i> Информация о соединении. Информацию о соединении нельзя преобразовать в ItemID с помощью метода TranslateToItemID и использовать в дальнейшем посредством интерфейса доступа к данным (Data Access).</p> <p><i>ftTime</i> Время возникновения на партнерском устройстве события. В атрибут EVENT_ATTR_S7_PCTIME вводится время, в которое сообщение было принято OPC-сервером событий.</p> <p><i>szMessage</i> Номер аварийного сообщения.</p> <p><i>dwEventType</i> OPC_SIMPLE_EVENT (Простое событие OPC)</p> <p><i>dwEventCategory</i> S7_PROCESS_ALARM (Аварийное сообщение процесса S7)</p> <p><i>pEventAttributes</i> Эта структура содержит атрибуты, передаваемые вместе с сообщением о событии. В состав атрибутов входят также значения, назначенные сообщению, источником которых является партнерское устройство.</p> <p>Все остальные элементы структуры не имеют значения.</p>

6.3.11 Интерфейс IOPCShutdown

Через этот интерфейс для приема вызова, базирующийся на точке соединения, сервер может информировать клиентов о своем предстоящем отключении (самостоятельном или со стороны). Благодаря этому клиенты могут подготовиться к выключению сервера.

Ниже перечислены методы интерфейса IOPCShutdown. Описаны только методы, характерные для SIMATIC NET. Описание методов можно найти в следующей спецификации OPC:

Alarms & Events Custom Interface (Пользовательский интерфейс для доступа к аварийным сообщениям и событиям)
Версия 1.10
Октябрь 2, 2002

Метод	Назначение
ShutdownRequest	Сервер уведомляет своих клиентов о том, что он выключается. Можно сконфигурировать текст с описанием причины отключения (szReason) в программе конфигурирования OPC-сервера.

6.3.12 Сообщения об ошибках, предусмотренные для переменных процесса интерфейса OPC DA

Коды ошибок, не зависящие от протокола

Код ошибки	Значение
OPC_E_INVALIDHANDLE (0xC0040001L)	Недопустимое значение дескриптора.
IDS_DUPLICATE (0xC0040002L)	Одно и то же значение было передано несколько раз.
IDS_UNKNOWNLCID (0xC0040003L)	Указанный идентификатор ("locale ID") не поддерживается сервером.
OPC_E_BADTYPE (0xC0040004L)	Преобразование канонического типа данных в запрошенный тип данных не поддерживается сервером.
OPC_E_PUBLIC (0xC0040005L)	Требуемая операция не может быть выполнена для "общей группы" (public group).

Код ошибки	Значение
OPC_E_BADRIGHTS (0xC0040006L)	Требуемая операция (чтение или запись) не допускается правами доступа, установленными для элемента.
OPC_E_UNKNOWNITEMID (0xC0040007L)	Имя (идентификатор элемента) отсутствует в пространстве имен сервера.
OPC_E_INVALIDITEMID (0xC0040008L)	Ошибка синтаксиса имени (идентификатора элемента).
OPC_E_INVALIDFILTER (0xC0040009L)	Недопустимое сочетание символов фильтра.
OPC_E_UNKNOWNPATH (0xC004000AL)	Недопустимое имя соединения, указанное в качестве AccessPath (Путь доступа).
OPC_E_RANGE (0xC004000BL)	Значение выходит за диапазон допустимых значений.
OPC_S_UNSUPPORTEDRATE (0x0004000DL)	Требуемая скорость обновления не поддерживается сервером. Используется ближайшее допустимое значение.
OPC_S_CLAMP (0x0004000EL)	Значение, переданное для записи, было принято, но было обрезано при выводе.
OPC_S_INUSE (0x0004000FL)	Операция не может быть полностью выполнена, поскольку по-прежнему существуют ссылки на объект.
OPC_E_NOTFOUND (0xC0040011L)	Метод GetErrorString возвращает строку в форме <i>dwError=%lx</i> , где <i>%lx</i> – это шестнадцатеричный идентификатор ошибки, неизвестной серверу SIMATIC OPC.
0xC0048003L	Произошло превышение времени, например, из-за разрыва соединения.
0xC0048004L	Используемый внутренний сервис был закрыт.
0xC0048005L	Требуемая операция (чтение или запись) запрещена правами доступа, выбранными для элемента.
0xC0048006L	Произошла непредусмотренная ошибка связи.
0xC0048007L	Значение превысило допустимый диапазон значений.
0xC0048008L	Значение находится ниже допустимого диапазона значений.
0xC0048009L	Произошла ошибка при преобразовании.

Коды ошибки для протокола DP

Код ошибки	Значение
0x85270101L	Ведущее устройство DP не работает.
0x85270102L	Ведомое устройство DP не работает.

Коды ошибок для протокола S7

Код ошибки	Значение
0x85270201L	Сервисы для передачи/приема буферов: недопустимое значение r_id.
0x85270202L	Сервисы для обслуживания доменов: недопустимые права доступа.
0x85270203L	Сервисы для обслуживания доменов: произошла ошибка блока.
0x85270204L	Сервисы для обслуживания доменов: произошла ошибка файла.

Коды ошибок для протокола PROFINET

Код ошибки	Значение
0x85270604L	Значение было только размещено в буфер.
0x85270607L	Неизвестный объект.
0x85270608L	Неизвестное свойство.
0x85270609L	Возвращенный тип не соответствует ожидаемому типу.
0x85270612L	Значение QoS (Код качества сервиса) не поддерживается.
0x85270613L	В настоящий момент система выполняет сохранение, изменение конфигурации пока что невозможно.
0x85270615L	Действие в настоящий момент не может быть применено.
0x85270616L	Доступ запрещен.
0x85270617L	Была обнаружена аппаратная неисправность.

Коды ошибок Windows

Код ошибки	Значение
0x80070005	Доступ запрещен.
0x80070057	Ошибка параметра.

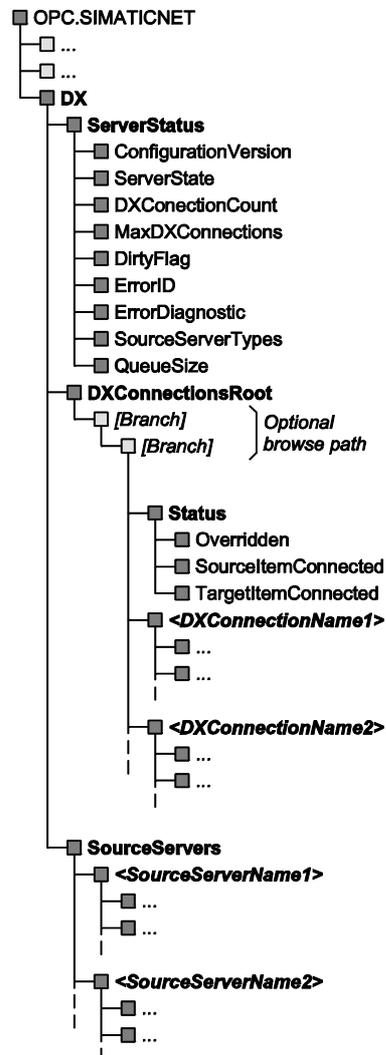
6.4 Программирование DX-интерфейса

6.4.1 Структура базы данных DX

Элементы базы данных DX в пространстве имен

В этом разделе описана структура базы данных DX (DX database) и представлен обзор задач, для которых могут использоваться ее элементы.

На следующем рисунке показано расположение отдельных элементов базы данных DX в пространстве имен. Элементы, предназначенные для DX-соединения и для сервера-источника, показаны на отдельных рисунках.



OPC DX-сервер SIMATIC NET базируется на DCOM-интерфейсе и на спецификации OPC DA 2.05.

Подробную информацию по теме OPC-DX можно найти в документе организации OPC Foundation: *OPC Data eXchange Specification, Version 1.0, Release March 5, 2003* (Спецификация обмена данными через OPC-интерфейс, версия 1.0, выпуск: 5 марта 2003 г.).

DX

DX-элемент представляет базу данных DX и располагается непосредственно под корневым узлом пространства адресов DA.

ServerStatus

Элементы ветви ServerStatus (Статус сервера) предоставляют базовую информацию о DX-сервере, например, работает ли сервер. Эти элементы обновляются DX-сервером, клиент их может только прочитать.

DXConnectionsRoot

Этот элемент содержит определения DX-соединений. DX-клиенты могут структурировать DX-соединения, создавая ветви под узлом *DXConnectionsRoot*.

<DXConnectionName>

Для каждого DX-соединения под элементом DXConnectionsRoot имеется отдельная ветвь. Имя DX-соединения является уникальным в пределах содержащей его ветви.

У DX-соединения имеются атрибуты, с помощью которых клиенты могут указывать способ передачи данных. Имеется два способа, посредством которых клиент может обращаться к этим атрибутам:

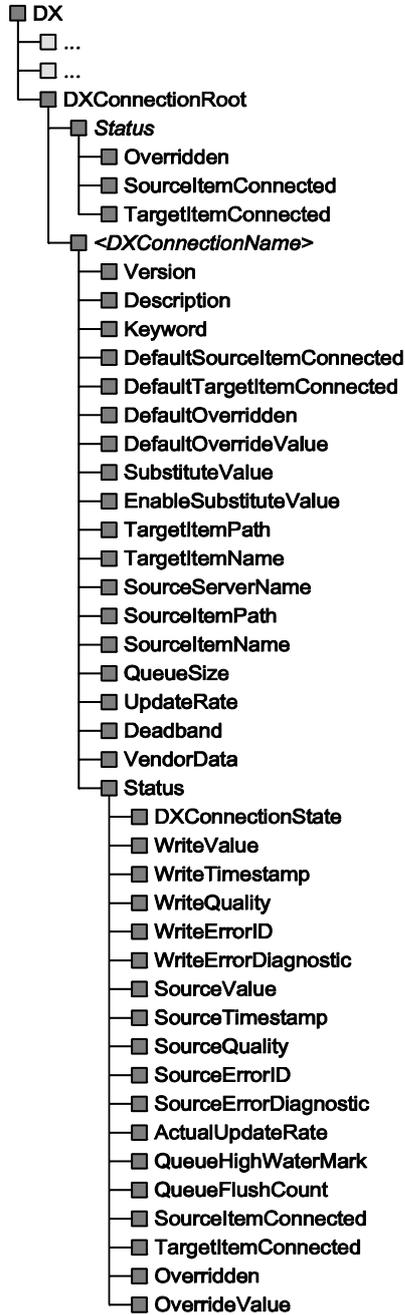
- Непосредственный доступ через интерфейс DA. В этом случае может быть прочитано большинство атрибутов.
- Использование способов конфигурирования DX. Эти способы, предназначенные для добавления, изменения и удаления DX-соединений, также предоставляют доступ для записи к атрибутам DX-соединения.

Единственным исключением является элемент *Status (Статус)* DX-соединения. Этот элемент можно читать или записывать с помощью методов DA-интерфейса. Клиент может читать или записывать элемент и его соответствующие атрибуты за один сеанс доступа. Если через DA-интерфейс осуществляется доступ для записи, те атрибуты, которые предназначены только для чтения, игнорируются. Точно так же, когда осуществляется доступ для чтения, недоступными являются элементы, предназначенные только для записи.

Другим вариантом является обращение к отдельным атрибутам с использованием их имен. Следует помнить, что атрибуты элемента "Status" (Статус) предназначены только для чтения, за исключением атрибутов *SourceItemConnected*, *TargetItemConnected*, *Overridden* и *OverrideValue*.

Элемент *Status* не следует путать с элементом с тем же именем, который располагается в каждой ветви непосредственно под *DXConnectionRoot* и влияет на каждое DX-соединение внутри ветви.

На следующем рисунке показаны элементы DX-соединения:



DA-клиенты могут регистрироваться с этими элементами, чтобы получать уведомления в случае изменения значений атрибутов.

Status (Статус)

DX-сервер добавляет этот элемент в каждую создаваемую им ветвь. Компоненты этого элемента влияют на все DX-соединения, расположенные в одной ветви, как и на соответствующий элемент Status (Статус). Если, к примеру, клиент записывает значение в атрибут *SourceItemConnected*, сервер записывает это значение во все элементы *SourceItemConnection*, принадлежащие DX-соединениям, располагающимся в ветвях нижнего уровня.

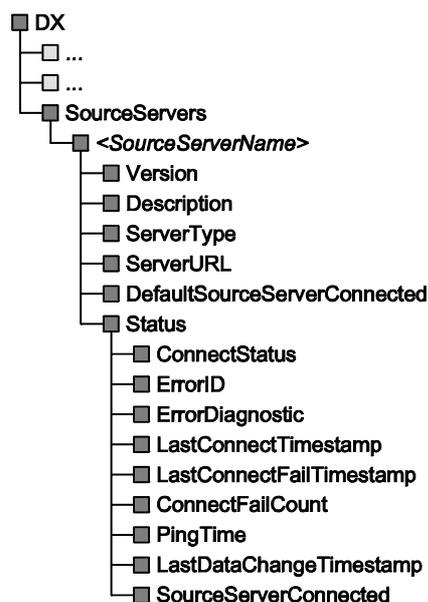
SourceServers (Серверы-источники)

Эта ветвь содержит все серверы-источники, определенные клиентом.

<SourceServerName> (Имя сервера-источника)

Сервер-источник добавляется, изменяется и удаляется клиентом. Для выполнения этих операций предусмотрены методы, принадлежащие сервисам конфигурирования DX. Имя сервера-источника является уникальным в пределах ветви *SourceServers* (Серверы-источники).

Атрибуты элемента содержат сведения о рабочем состоянии сервера-источника, например, время последнего установления соединения с сервером-источником. На следующем рисунке показаны атрибуты элемента для сервера-источника:



Все атрибуты доступны как DA-элементы. Кроме элемента *SourceServerConnected*, эти элементы могут быть только прочитаны. DA-клиенты могут регистрироваться с этими элементами, чтобы получать уведомления об изменениях значений атрибутов.

Чтобы получить доступ к элементу, DX-клиент должен использовать методы конфигурирования DX. DA-клиент также может обращаться к элементу через пространство имен.

6.4.2 Сервисы конфигурирования DX

Введение

Для конфигурирования базы данных DX (DX database), которая была описана в предыдущем разделе, предусмотрены специальные методы, известные также как сервисы конфигурирования DX. Эти методы определяются независимо от какой-либо конкретной реализации, хотя в спецификации DX описаны реализации для веб-сервисов и DCOM. OPC DX-сервер SIMATIC NET базируется на спецификации OPC Data Access (Доступ к данным через OPC) версии 2.05 и предоставляет DCOM-интерфейс для OPC-DX.

Как DX Server обрабатывает запросы

DX-сервер обрабатывает запросы один за другим, по мере их поступления. В случае необходимости, DX-сервер обновляет свою базу данных DX и атрибуты режима выполнения (runtime-атрибуты), возвращая ответ. Изменения становятся "видимыми" клиентом сразу же. Другими словами, DX-сервер не дожидается, пока изменения вступят в силу.

Благодаря такому поведению в режиме выполнения, клиенты могут вносить изменения в базу данных DX независимо от того, имеются ли соответствующие серверы-источники.

Необходимость наличия сервисов конфигурирования DX

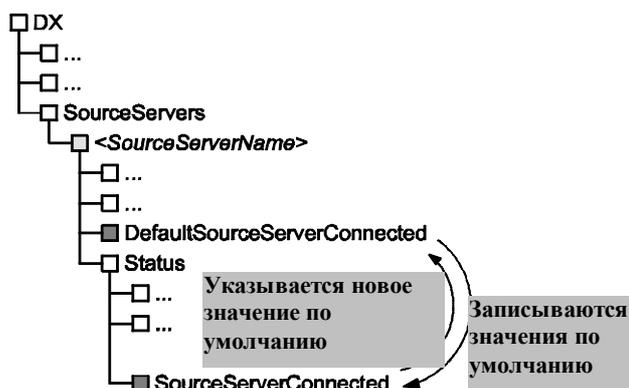
Конфигурирование может быть выполнено полностью только при наличии методов DX-интерфейса. Только использование сервисов конфигурирования DX позволяет обрабатывать запросы в соответствии с приведенным здесь описанием. Обращение к элементам базы данных DX через DA-интерфейс с целью конфигурирования возможно, хотя и с ограничениями.

Сервисы сервера-источника

Существует пять методов, относящихся к серверам-источникам:

- *GetServers (Прочитать серверы)*
Получение информации обо всех серверах-источниках из базы данных DX.
- *AddServers (Добавить серверы)*
Клиент вызывает этот метод, чтобы добавить один или несколько серверов-источников в базу данных DX.
- *ModifyServers (Изменить серверы)*
Метод служит для изменения одного или нескольких серверов-источников. Это может оказаться полезным в случае изменения местоположения сервера-источника в сети.
- *DeleteServers (Удалить серверы)*
Метод удаляет один или несколько серверов-источников из базы данных DX. В качестве параметра должен быть указан идентификатор удаляемого сервера. Если идентификатор не указан, из базы данных DX удаляются все серверы.

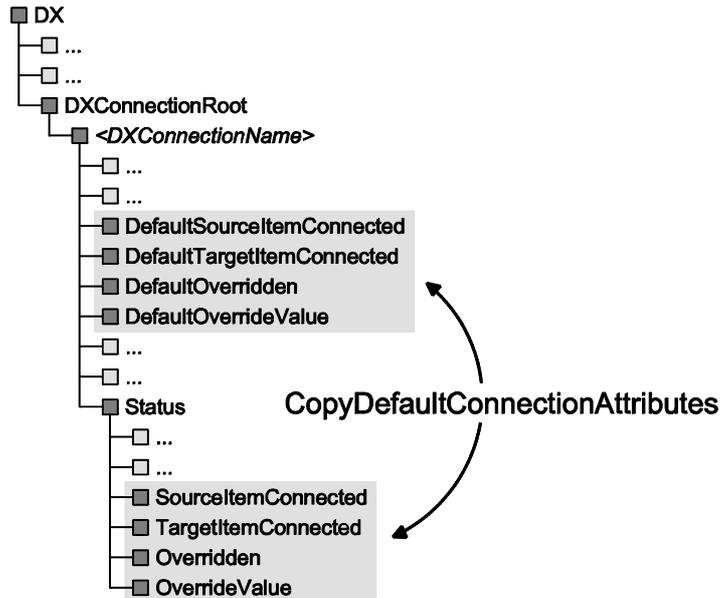
- *CopyDefaultServerAttributes* (Копировать принимаемые по умолчанию атрибуты сервера)
 Данный метод копирует значение элемента *DefaultSourceServerConnected* в атрибут *SourceServerConnected* элемента *Status* (Статус) и наоборот. В первом случае в атрибут элемента *Status* записывается принимаемое по умолчанию значение, а во втором случае – указывается новое значение, принимаемое по умолчанию. Описанное взаимодействие представлено на рисунке ниже:



Сервисы DX-соединения

- *QueryDXConnections* (Опросить DX-соединения)
 Этот метод возвращает все DX-соединения, удовлетворяющие определенным условиям. Критерий выбора передается при вызове метода в качестве входного параметра. DX-соединение должно обладать всеми свойствами, перечисленными во входном параметре, только в этом случае оно может быть выбрано с помощью *QueryDXConnections*.
- *AddDXConnections* (Добавить DX-соединения)
 Этот метод добавляет одно или несколько DX-соединений в базу данных DX.
- *UpdateDXConnections* (Обновить DX-соединения)
 Этот метод обновляет атрибуты конфигураций одного или нескольких DX-соединений. В параметрах, вводимых для данного метода, указываются DX-соединения, для которых выполняется обновление.
- *ModifyDXConnections* (Изменить DX-соединения)
 Метод изменяет одно или несколько DX-соединений. В качестве входного параметра клиенту требуется только указать атрибуты, которые ему требуется изменить.
- *DeleteDXConnections* (Удалить DX-соединения)
 С помощью данного метода клиент удаляет одно или несколько DX-соединений в пределах определенной ветви DX-базы. Клиент указывает, какие именно DX-соединения должны быть удалены, указывая соответствующие свойства. Будут удалены только те соединения, которые обладают этими свойствами.
 Если клиент указывает только ветвь базы данных DX и не указывает другие свойства, удаляются все DX-соединения этой ветви.

- *CopyDefaultConnectionAttributes*
(Копировать принимаемые по умолчанию атрибуты соединений)
Аналогично методу *CopyDefaultServerAttributes* этот метод копирует значения атрибутов конфигурации в атрибуты элемента *Status*. Метод воздействует на следующие атрибуты:
 - *DefaultSourceItem Connected* и *SourceItem Connected*
 - *DefaultTargetItem Connected* и *TargetItem Connected*
 - *DefaultOverridden* и *Overridden*
 - *DefaultOverrideValue* и *OverrideValue*



ResetConfiguration (Сбросить конфигурацию)

Этот метод удаляет все DX-соединения и серверы-источники из базы данных DX. Этот метод переводит DX-сервер в его исходное состояние, за исключением атрибута *ConfigurationVersion* (*Версия конфигурации*), который обновляется.

В качестве дополнительного параметра для данного метода можно указать атрибут *ConfigurationVersion* (*Версия конфигурации*). Если указан этот параметр, элементы базы данных DX удаляются только в том случае, если версия базы данных совпадает со значением этого параметра.

6.5 Программирование XML-интерфейса

Примечание

Интерфейс OPC XML поддерживается программным обеспечением SIMATIC NET, начиная с версии 6.1.

Введение

В этом разделе описаны методы, предусмотренные для интерфейса XML DA. В нем в графическом виде представлена структура тегов, а также описаны отдельные элементы и атрибуты.

В начале раздела рассказывается о синтаксисе описания методов. Затем следует описание основных схем. Описываются элементы, которые встречаются при описании схем OPC XML–DA больше одного раза. Далее рассматриваются отдельные методы.

Определение типов данных в языке XML Schema (XML-схемы)

В соответствии с определением XMLSchema2001, раздел schema (схема) в WSDL-документе интерфейса OPC XML–DA описывает комплексные типы данных, которые используются в качестве параметров для отдельных методов. Подробное описание можно найти в спецификации OPC XML–DA.

В коде веб-сервиса для каждого комплексного типа данных из XML-схемы создается класс C#. Методы веб-сервиса описываются в интерфейсе.

6.5.1 Описание элементов

Правила, применяемые для описания элементов

Описаниям элементов OPC–XML предшествует представление структуры тегов. Элементы выделяются жирным шрифтом, а атрибуты – курсивом. Степень вложенности тегов иллюстрируется величиной отступа от левой границы документа. Атрибуты тега также отображаются с отступом справа от соответствующего тега.

Пример

Структура тегов может выглядеть следующим образом:

```
<tag1>
  <tag2 att1="yes" att2="no">content of tag2
  </tag2>
  <tag3>content of tag3
  </tag3>
</tag1>
```

В описании <tag1> будет, следовательно, размещена структура, показанная ниже:

```
tag1
  tag2
    att1
    att2
  tag3
```

Элементы tag2 и tag3 смещены относительно элемента tag1, поскольку они являются подэлементами элемента tag1. Сразу после элемента tag2 темным синим шрифтом отображены атрибуты att1 и att2 этого элемента.

6.5.2 Основные схемы

Определение схемы для OPC XML-DA

Определение схемы предусматривается для всех элементов OPC-XML-DA. (Далее для элементов OPC-XML-DA будем применять термин "XML-элемент", чтобы отличать элементы OPC-XML-DA от OPC-элементов (item), которые будем называть просто "элементами" – прим. переводчика). Далее мы будем часто ссылаться на определения базовых, наиболее часто используемых XML-элементов. Некоторые из этих XML-элементов описаны ниже:

ItemProperty (Свойство элемента)

Этот XML-элемент предназначен для описания свойств элемента.

ItemValue (Значение элемента)

С помощью этого XML-элемента указывается значение элемента и любая другая дополнительная информация (например, метка времени).

OPCError (Ошибка OPC)

Содержит код ошибки и описание ошибки OPC.

ReplyBase (Основные сведения об ответе)

XML-элемент, содержащий основную информацию об ответе сервера (например, время, когда был передан ответ).

RequestOptions (Опции запроса)

Содержит более подробные сведения о запросе клиента.

6.5.3 ItemProperty (Свойство элемента)

Структура тега

```
ItemProperty  
  Name  
  Description  
  ItemPath  
  ItemName  
  ResultID  
  Value
```

XML-элементы и атрибуты

Name (Имя)

Содержит имя свойства

Description (Описание)

Описание свойства.

ItemPath (Путь к элементу)

Путь к элементу. Этот атрибут не поддерживается в SIMATIC NET.

ItemName (Имя элемента)

С помощью атрибута ItemName (Имя элемента) элемент можно однозначно идентифицировать в пространстве имен сервера.

ResultID (Идентификатор результата)

Этот атрибут содержит имя ошибки OPC, если происходит ошибка или некритичное исключение.

Value (Значение)

Текущее значение свойства.

6.5.4 ItemValue (Значение элемента)

Структура тега

```

ItemValue
  ValueTypeQualifier
  ItemPath
  ItemName
  ClientItemHandle
  Timestamp
  ResultID
  DiagnosticInfo
  Value
  Quality
    QualityField
    LimitField
    VendorField

```

Элементы и атрибуты

ValueTypeQualifier (Квалификатор типа значения)

Этот атрибут используется только для выбора типа данных для значений типа "время", "дата" и "продолжительность". Для всех остальных типов данных этот атрибут либо не существует, либо игнорируется, если он существует.

ItemPath (Путь к элементу)

Путь доступа к элементу. Этот атрибут в SIMATIC NET не поддерживается.

ItemName (Имя элемента)

С помощью атрибута ItemName (Имя элемента) элемент можно однозначно идентифицировать в пространстве имен сервера.

ClientItemHandle (Дескриптор элемента клиента)

Символьная строка, указанная клиентом в запросе, которую сервер возвращает в своем ответе. Она используется клиентом в более сложных системах, когда требуется идентификация и сопоставление ответов соответствующим запросам.

Timestamp (Метка времени)

Время последнего получения значений от сервера.

ResultID (Идентификатор результата)

Этот атрибут содержит имя ошибки OPC, если происходит ошибка или некритичное исключение.

Diagnostic Info (Диагностическая информация)

Подробная диагностическая информация, зависящая от сервера.

Value (Значение)

Значение элемента. Поскольку с этим атрибутом связана многообразная информация, необходим дополнительный атрибут xsi:type (например: xsi:type="xsd:float").

Quality (Код качества)

Сведения о "качестве" данных. Если сервер не возвращает этот атрибут, считается, что код качества = "Good" (Ошибок нет). Атрибут возвращается, когда код качества = "Bad" (Ошибка) или "Uncertain" (Неопределенное состояние).

QualityField (Поле кода качества)

Если код качества имеет значение "Good", этот атрибут не возвращается. Для кода качества "Bad", когда известно предшествующее значение элемента, это атрибут имеет значение "badLastKnownValue" (Последнее известное значение).

LimitField (Поле бита проверки границ)

Содержит имя, посредством которого можно обращаться к полю бита проверки границ OPC. Этот атрибут передается всегда, за исключением случая, когда статус проверки границ имеет значение "none" (нет).

VendorField (Поле изготовителя)

Числовое значение, которое соответствует полю бита изготовителя OPC. Передается этот атрибут или нет, зависит от самого производителя.

6.5.5 OPCError (Ошибка OPC)

Структура тега

OPCError
ID
Text

XML-элементы и атрибуты

ID (Идентификатор)

Содержит имя ошибки OPC.

Text (Текст)

Описание ошибки в текстовом виде. Содержание строки зависит от атрибута "LocaleID" (Идентификатор языка).

6.5.6 ReplyBase (Основная информация об ответе)**Структура тега****ReplyBase***RcvTime**ReplyTime**ClientRequestHandle**RevisedLocaleID**ServerState***XML-элементы и атрибуты*****RcvTime (Время приема)***

Указывает время, в которое сервер получил запрос. RcvTime является обязательным атрибутом.

ReplyTime (Время ответа)

Указывает время, в которое сервер передал ответ. Является обязательным атрибутом.

ClientRequestHandle (Дескриптор запроса клиента)

Если этот атрибут был включен в запрос клиента, он возвращается сервером в ответе.

RevisedLocaleID (Измененный идентификатор языка)

Если клиент указывает для атрибута "LocaleID" значение, которое поддерживается сервером, сервер возвращает для атрибута "LocaleID" принимаемое по умолчанию значение с атрибутом "RevisedLocaleID".

ServerState (Состояние сервера)

Указывает статус сервера и возвращается в любом случае. Является обязательным атрибутом.

6.5.7 RequestOptions (Опции запроса)

Структура тега

RequestOptions
ReturnErrorText
ReturnDiagnosticInfo
ReturnItemTime
ReturnItemName
ReturnItemPath
RequestDeadline
ClientRequestHandle
LocaleID

XML-элементы и атрибуты

ReturnErrorText (Возвращать сообщения об ошибках)

По умолчанию этот атрибут имеет значение "TRUE" (Верно). В этом случае сервер возвращает подробные описания ошибок.

ReturnDiagnosticInfo (Возвращать диагностическую информацию)

Если этот атрибут имеет значение "TRUE" (Верно), сервер возвращает подробную диагностическую информацию.

ReturnItemTime (Возвращать время элемента)

Указывает, должна ли для каждого элемента возвращаться метка времени. По умолчанию имеет значение "FALSE" (Ложь), метка времени не возвращается.

ReturnItemName (Возвращать имя элемента)

Указывает, должен ли для каждого элемента возвращаться атрибут "ItemName" (Имя элемента). По умолчанию имеет значение "FALSE", имя элемента не возвращается.

ReturnItemPath (Возвращать путь к элементу)

Указывает, должен ли для каждого элемента возвращаться атрибут "ItemPath" (Путь к элементу). По умолчанию имеет значение "FALSE" (Ложь), путь к элементу не возвращается.

RequestDeadline (Конечное время ответа)

Указывает время, в течение которого клиент будет ожидать ответ от сервера. Если некоторые элементы к этому времени не доступны, вместо их данных возвращается сообщение об ошибке. Если указано значение времени, которое меньше, чем текущее время сервера, весь запрос полностью завершается ошибкой. Клиенты должны указывать время UTC.

ClientRequestHandle (Дескриптор запроса клиента)

Если этот атрибут был включен в запрос клиента, он возвращается сервером в ответе. В больших и сложных системах эта информация помогает клиенту сопоставлять запросы ответам. Этот атрибут указывать необязательно.

LocaleID (Идентификатор языка)

Необязательный атрибут, с помощью которого клиент выбирает язык для некоторых возвращаемых данных.

6.5.8 Read (Чтение)**Структура тега****Read****Options**

ReturnErrorText
ReturnDiagnosticInfo
ReturnItemTime
ReturnItemName
ReturnItemPath
RequestDeadline
ClientRequestHandle
LocaleID

ItemList

ItemPath
ReqType
MaxAge

Items

ItemPath
ReqType
ItemName
ClientItemHandle
MaxAge

XML-элементы и атрибуты***Read (Чтение)***

XML-элемент <Read> содержит всю информацию о задании на чтение.

Options (Опции)

XML-элемент <Options> содержит опции, предусмотренные для запросов XML-DA. Этот XML-элемент имеет тип RequestOptions. Сведения об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

ItemList (Список элементов)

XML-элемент-контейнер для отдельных элементов. Запрос может содержать несколько элементов.

Для XML-элементов ItemList и Items можно использовать атрибуты *ItemPath*, *ReqType* и *MaxAge*. Значение атрибута, указанное для элемента, записывается вместо значения атрибута с тем же именем, принадлежащего XML-элементу ItemList.

ItemPath (Путь к элементу)

Путь доступа к элементу. Этот атрибут не поддерживается в SIMATIC NET.

MaxAge (Максимальный срок "годности")

Интервал времени типа xsd:int, который определяет максимальный срок "годности" данных. Данные не могут быть старше, чем указанный здесь срок годности.

ReqType (Тип запрошенных данных)

Указывает тип данных для значения элемента, запрошенного клиентом. Поддерживаются все типы данных, перечисленные в спецификации.

Items (Элементы)

XML-элемент, с помощью которого указывается отдельный элемент.

ItemName (Имя элемента)

Имя элемента.

ClientItemHandle (Дескриптор элемента клиента)

Строка, назначенная клиентом. Если этот атрибут указан клиентом, сервер должен вернуть его.

Пример

```
<soap:Body>
  <Read
    xmlns="http://opcfoundation.org/webservices/XMLDA/1.0/">
    <Options
      ReturnErrorText="false"
      ReturnItemTime="true"
      ReturnItemName="true"
      LocaleID="en" />
    <ItemList>
      <Items ItemName="Simple Types/UInt" />
      <Items ItemName="Simple Types/Int" />
      <Items ItemName="Simple Types/Float" />
    </ItemList>
  </Read>
</soap:Body>
```

С#-интерфейс для метода Synchronous Read Web (Синхронное чтение через web)

В случае использования пользовательских программ, написанных на С#, для чтения значений предусмотрен следующий метод. Должны быть указаны только некоторые параметры, перечисленные выше:

```
ReplyBase Read(RequestOptions Options,
               ReadRequestItemList ItemList,
               out ReplyItemList ItemValues,
               out OPCError[] Error);
```

6.5.9 ReadResponse (Ответ на задание на чтение)

Структура тега

```

ReadResponse
  ReadResult
    RcvTime
    ReplyTime
    ClientRequestHandle
    RevisedLocaleID
    ServerState
  RItemList
    Reserved
  Items
    ValueTypeQualifier
    ItemPath
    ItemName
    ClientItemHandle
    Timestamp
    ResultID
  DiagnosticInfo
  Value
  Quality
    QualityField
    LimitField
    VendorField
  Errors
    ID
    Text

```

XML-элементы и атрибуты

ReadResponse (Ответ на чтение)

С помощью XML-элемента <ReadResponse> сервер возвращает результаты на полученное им задание на чтение.

ReadResults (Результаты чтения)

XML-элемент <ReadResults> содержит основные сведения об ответе сервера.

Этот элемент имеет тип ReplyBase. Информацию об атрибутах этого элемента можно найти в соответствующем разделе настоящей документации.

RItemList (Список элементов ответа)

RItemList – это XML-элемент, содержащий отдельные элементы ответа. Ответ может содержать несколько элементов.

Reserved

Этот атрибут запрещает программам, написанным на языке WSDL, представлять ответный список в виде массива элементов для генерации кода.

Items (Элементы)

XML-элемент, с помощью которого указывается отдельный элемент.

Этот XML-элемент имеет тип ItemValue. Сведения об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

Errors (Ошибки)

Массив ошибок, возвращаемых с данным ответом.

Этот XML-элемент имеет тип OPCError. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

Пример

```
<soap:Body>
  <ReadResponse
    xmlns="http://opcfoundation.org/webservices/XMLDA/1.0/">
    <ReadResult
      RcvTime="2003-05-27T00:15:36.6400000-07:00"
      ReplyTime="2003-05-27T00:15:36.7500000-07:00"
      ServerState="running"
    />
    <RItemList>
      <Items
        ItemName="Simple Types/UInt"
        Timestamp="2003-05-27T00:15:36.7343750-07:00">
        <Value (значение)
xsi:type="xsd:unsignedInt">4294967295</Value>
        </Items>
      <Items
        ItemName="Simple Types/Int"
        Timestamp="2003-05-27T00:15:36.7343750-07:00">
        <Value xsi:type="xsd:int">2147483647</Value>
        </Items>
      <Items
        ItemName="Simple Types/Float"
        Timestamp="2003-05-27T00:15:36.7343750-07:00">
        <Value xsi:type="xsd:float">3.402823E+38</Value>
        </Items>
    </RItemList>
  </ReadResponse>
</soap:Body>
```

6.5.10 Write (Запись)

Структура тега

```

Write
  ReturnValuesOnReply
  Options
    ReturnErrorText
    ReturnDiagnosticInfo
    ReturnItemTime
    ReturnItemName
    ReturnItemPath
    RequestDeadline
    ClientRequestHandle
    LocaleID
  ItemList
    ItemPath
    Items
      ValueTypeQualifier
      ItemPath
      ItemName
      ClientItemHandle
      Timestamp
      ResultID
    DiagnosticInfo
  Value
  Quality
    QualityField
    LimitField
    VendorField

```

XML-элементы и атрибуты

Write (Запись)

XML-элемент <Write> содержит всю информацию о задании на запись. Задание на запись может быть выполнено для одного или нескольких элементов. Как вариант, также могут быть записаны значения в атрибуты "LimitBits" (Биты проверки границ) и "VendorBits" (Биты производителя).

ReturnValuesOnReply (Возвращать значения в ответе)

С помощью этого атрибута клиент может указать, должно ли для каждого элемента возвращаться значение. Возвращается значение, с которым сервер оперировал в связи с данным заданием на чтение. Это то же самое значение, которое сервер вернул бы, если бы сразу же вслед за заданием на запись поступило задание на чтение.

Если задание на запись завершается ошибкой, значения не возвращаются. В случае значений, предназначенных только для записи, сервер возвращает статус E_WRITEONLY.

Options (Опции)

XML-элемент <Options> содержит опции, которые предусмотрены для запросов XML-DA. Этот XML-элемент имеет тип RequestOptions. Информацию об атрибутах данного XML-элемента можно найти в соответствующем разделе настоящей документации.

ItemList (Список элементов)

Элемент-контейнер для отдельных элементов. Запрос может содержать несколько элементов.

ItemName (Имя элемента)

С помощью атрибута ItemName (Имя элемента) элемент можно однозначно идентифицировать в пространстве имен сервера.

Items (Элементы)

XML-элемент, с помощью которого указывается отдельный элемент.

Этот XML-элемент имеет тип ItemValue. Информацию об атрибутах данного XML-элемента можно найти в соответствующем разделе настоящей документации.

Пример

```
<soap:Body>
  <Write
    xmlns="http://opcfoundation.org/webservices/XMLDA/1.0/">
    <Options
      ReturnErrorText="false"
      ReturnItemName="true"
      LocaleID="en"
    />
    <ItemList>
      <Items ItemName="Simple Types/UInt">
        <Value xsi:type="xsd:unsignedInt">4294967295</Value>
      </Items>
      <Items ItemName="Simple Types/Int">
        <Value xsi:type="xsd:int">2147483647</Value>
      </Items>
      <Items ItemName="Simple Types/Float">
        <Value xsi:type="xsd:float">3.402823E+38</Value>
      </Items>
    </ItemList>
  </Write>
</soap:Body>
```

С#-интерфейс для метода Synchronous Write Web (Синхронная запись через web)

В случае применения программ пользователя, написанных на С#, для записи значений предусмотрен следующий метод. Должны быть указаны только некоторые параметры, перечисленные выше:

```
ReplyBase Write(RequestOptions Options,
                WriteRequestItemList ItemList,
                bool ReturnItemVal,
                out ReplyItemList ItemValues,
                out OPCError[] Error);
```

6.5.11 WriteResponse (Ответ на задание на запись)

Структура тега

```
WriteResponse
  WriteResult
    RcvTime
    ReplyTime
    ClientRequestHandle
    RevisedLocaleID
    ServerState
  RItemList
    Reserved
  Items
    ValueTypeQualifier
    ItemPath
    ItemName
    ClientItemHandle
    Timestamp
    ResultID
  DiagnosticInfo
  Value
  Quality
    QualityField
    LimitField
    VendorField
Errors
  ID
  Text
```

XML-элементы и атрибуты

WriteResponse (Ответ на задание на запись)

С помощью XML-элемента <WriteResponse> сервер возвращает результаты в ответ на задание на запись.

WriteResult (Результат записи)

XML-элемент <WriteResult> содержит основную информацию об ответе сервера.

Этот элемент имеет тип ReplyBase. Информацию об атрибутах данного XML-элемента можно найти в соответствующем разделе настоящей документации.

RitemList (Список элементов)

Элемент-контейнер для отдельных элементов. Запрос может содержать несколько элементов.

Reserved

Этот атрибут запрещает программам, написанным на языке WSDL, представлять ответный список в виде массива элементов для генерации кода.

Items (Элементы)

XML-элемент, с помощью которого указывается отдельный элемент.

Этот XML-элемент имеет тип ItemValue. Сведения об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

Errors (Ошибки)

Массив ошибок, возвращаемых с данным ответом.

Этот XML-элемент имеет тип OPCError. информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

Пример

```
<soap:Body>
  <WriteResponse
    xmlns="http://opcfoundation.org/webservices/XMLDA/1.0/">
    <WriteResult
      RcvTime="2003-05-27T05:19:26.3687500-07:00"
      ReplyTime="2003-05-27T05:19:26.4687500-07:00"
      ServerState="running"
    />
    <RitemList>
      <Items ItemName="Simple Types/UInt" />
      <Items ItemName="Simple Types/Int" />
    </RitemList>
  </WriteResponse>
</soap:Body>
```

```
<Items ItemName="Simple Types/Float" />
</RItemList>
</WriteResponse>
</soap:Body>
```

6.5.12 Применение "подписок"

Поддержание связи при неоднократном вызове методов

В отличие от большинства других Web-сервисов, Web-сервисы OPC XML-DA поддерживают так называемую свободную (незакрепленная) связь с подписавшимся Web-клиентом посредством неоднократного вызова метода. Такая "незакрепленная" связь известна также как "подписка с опросом" (polled subscription). Подписка с опросом позволяет выполнять чтение значений элементов циклически.

Подписка и отписка клиента (Регистрация и отмена регистрации)

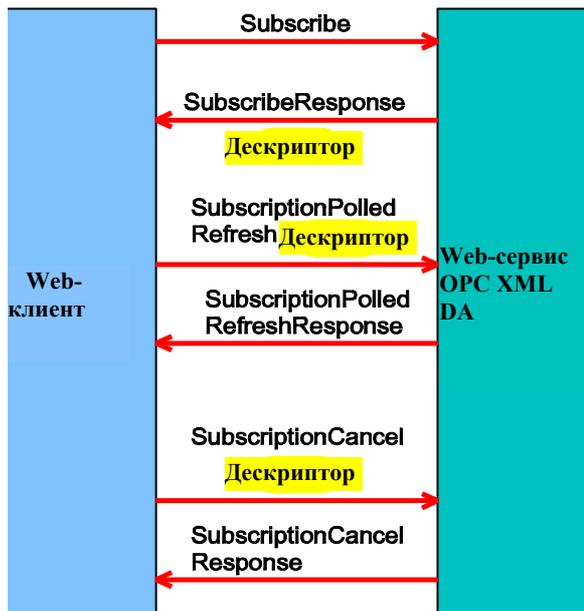
Клиент регистрирует подписку посредством Web-сервиса OPC XML-DA. Последний затем указывает, какие элементы "интересны" клиенту, и опрашивает их значения. Когда от клиента поступает очередной запрос в рамках опроса, в ответ возвращаются текущие значения. Подписка отменяется клиентом.

Идентификация с помощью дескрипторов

С целью идентификации подписки с каждым последующим вызовом в качестве параметра передается дескриптор, назначенный сервером. Также имеется возможность указать список дескрипторов, чтобы для одного запроса SubscriptionPolledRefresh было действительно несколько подписок. Это повышает эффективность опроса.

В качестве дескриптора подписки клиенту возвращается адрес объекта подписки, что позволяет однозначно идентифицировать подписку в следующем вызове SubscriptionPolledRefresh и SubscriptionCancel.

Использование подписки наглядно продемонстрировано на следующем рисунке:



Период действительности контролируемых элементов

Метод `SubscriptionPolledResponse` возвращает клиенту значения, которые были запрошены методом `SubscriptionPolledRefresh`. Метод `SubscriptionCancel` отменяет все подписки и их элементы. Если в дальнейшем от клиента не поступает новых запросов `SubscriptionPolledRefresh` в течение указанного промежутка времени (`SubscriptionPingRate`), Web-сервис OPC DA автоматически закрывает подписку точно так же, как и метод `SubscriptionCancel`.

6.5.13 Subscribe (Задание на подписку)

Структура тега

Subscribe

ReturnValuesOnReply
SubscriptionPingRate

Options

ReturnErrorText
ReturnDiagnosticInfo
ReturnItemTime
ReturnItemName
ReturnItemPath
RequestDeadline
ClientRequestHandle
LocaleID

ItemList

ItemPath
ReqType
Deadband
RequestedSamplingRate
EnableBuffering

Items

ItemPath
ReqType
ItemName
ClientItemHandle
Deadband
RequestedSamplingRate
EnableBuffering

XML-элементы и атрибуты

Subscribe (Задание на подписку)

XML-элемент <Subscribe> содержит всю информацию о задании на подписку.

ReturnValuesOnReply (Возвращать значения в ответе)

Если для данного атрибута выбрано значение TRUE (Верно), сервер возвращает значения, имеющиеся для SubscribeResponse. Если выбрано значение FALSE (Ложь), сервер не возвращает значения для SubscribeResponse.

SubscriptionPingRate (Период проверки подписчика)

Значение этого атрибута определяет интервал, с которым сервер проверяет наличие клиента. Если клиент не вступал в связь с сервером в течение указанного периода времени, сервер может освободить все ресурсы, которые были необходимы для обслуживания подписки клиента.

Options (Опции)

XML-элемент <Options> содержит опции, предусмотренные для запросов XML-DA. Этот элемент имеет тип RequestOptions. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

Item List (Список элементов)

XML-элемент-контейнер для отдельных элементов. Запрос может содержать несколько элементов.

Для XML-элементов ItemList и Items можно использовать атрибуты *ItemPath*, *ReqType*, *Deadband*, *RequestedSamplingRate* и *EnableBuffering*. Значение атрибута, установленное для элемента, также записывается вместо значения атрибута с тем же именем, принадлежащего XML-элементу ItemList.

ItemPath (Путь к элементу)

Путь к элементу. Этот атрибут не поддерживается в SIMATIC NET.

ReqType (Тип запроса)

Указывает тип данных для значения элемента, запрошенного клиентом.

Deadband (Зона нечувствительности)

Этот атрибут указывает пороговое значение, на которое должно измениться значение элемента, чтобы был запущен метод SubscriptionPolledRefresh. Пороговое значение указывается в процентах от максимального диапазона значений элемента. Следовательно, значение этого параметра должно находиться в пределах 0 ... 100 процентов и может использоваться только для элементов типа integer (целое число) и float (число с плавающей запятой).

RequestedSamplingRate (Запрошенная скорость опроса)

Время, указанное клиентом в миллисекундах, по истечении которого сервер проверяет значение на наличие изменений.

EnableBuffering (Разрешение размещения в буфер)

Если для данного атрибута установлено значение TRUE (Верно), сервер сохраняет все изменившиеся значения в буфер в соответствии со значением параметра RequestedSamplingRate. Сервер возвращает сохраненные данные клиенту при следующем запросе PolledRequest.

Item Name (Имя элемента)

Имя элемента.

ClientItemHandle (Дескриптор элемента клиента)

Строка, назначенная клиентом. Если этот атрибут указывается клиентом, сервер должен его вернуть.

6.5.14 SubscribeResponse (Ответ на запрос о подписке)

Структура тега

```
SubscribeResponse
  ServerSubHandle
  SubscribeResult
    RcvTime
    ReplyTime
    ClientRequestHandle
    RevisedLocaleID
    ServerState
  RItemList
    RevisedSamplingRate
    Items
      RevisedSamplingRate
      ItemValue
        ValueTypeQualifier
        ItemPath
        ItemName
        ClientItemHandle
        Timestamp
        ResultID
      DiagnosticInfo
      Value
      Quality
        QualityField
        LimitField
        VendorField
  Errors
    ID
    Text
```

XML-элементы и атрибуты

SubscribeResponse (Ответ на запрос о подписке)

XML-элемент <SubscribeResponse> содержит всю информацию о методе SubscribeResponse.

ServerSubHandle (Субдескриптор сервера)

Значение, указанное сервером для данного элемента, должно использоваться с запросами SubscriptionPolledRefresh и SubscriptionCancel. Этот атрибут идентифицирует клиента, который передает запрос.

SubscribeResult (Результат подписки)

XML-элемент <SubscribeResult> содержит основную информацию об ответе сервера.

Этот XML-элемент имеет тип ReplyBase. Подробные сведения об этом элементе можно найти в соответствующем разделе настоящей документации.

RitemList (Список элементов)

Содержит элементы. Имеющиеся значения элементов передаются клиенту только в том случае, если клиент запросил их, установив соответствующее значение для атрибута ReturnValuesOnReply (см. Subscribe).

RevisedSamplingRate (Поддерживаемая скорость опроса)

Наиболее короткий цикл обновления, поддерживаемый сервером. Это значение возвращается клиенту сервером.

Items (Элементы)

XML-элемент, с помощью которого указывается отдельный элемент.

Этот элемент имеет тип ItemValue. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

Errors (Ошибки)

Массив ошибок, произошедших в связи с этим ответом.

Этот XML-элемент имеет тип OPCError. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

6.5.15 SubscriptionPolledRefresh (Обновление элементов подписки)

Структура тега

SubscriptionPolledRefresh

HoldTime

WaitTime

ReturnAllItems

Options

ReturnErrorText

ReturnDiagnosticInfo

ReturnItemTime

ReturnItemName

ReturnItemPath

RequestDeadline

ClientItemHandle

LocaleID

ServerSubHandles

XML-элементы и атрибуты

***SubscriptionPolledRefresh* (Обновление элементов подписки)**

С помощью этого XML-элемента клиент запрашивает сервер выполнить обновление элементов, которые были определены в предшествующем запросе Subscription (Подписка).

***HoldTime* (Время удержания)**

Этот атрибут указывает серверу не передавать какие-либо обновленные значения клиенту, пока внутреннее время сервера не будет равно или не превысит значение этого атрибута.

***WaitTime* (Время ожидания)**

По достижении времени, указанного атрибутом HoldTime, сервер ожидает в течение времени, указанного атрибутом WaitTime, прежде чем вернуть ответ. Если значение элемента изменяется в течение времени ожидания, ответ передается сразу же.

***ReturnAllItems* (Вернуть все элементы)**

Если для данного атрибута выбрано значение FALSE (Ложь), сервер возвращает только те элементы, значения которых изменились с момента последнего запроса SubscriptionPolledRefresh, до поступления текущего запроса SubscriptionPolledRefresh.

Если для атрибута выбрано значение TRUE (Верно), сервер возвращает все элементы, которые были определены в соответствующем запросе Subscription (Подписка).

Options (Опции)

XML-элемент <Options> содержит опции, которые предусмотрены для запросов XML-DA. Этот элемент имеет тип RequestOptions. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

ServerSubHandles (Субдескрипторы сервера)

Этот атрибут используется сервером для идентификации подписки, для которой должен выполняться опрос.

6.5.16 SubscriptionPolledRefreshResponse (Ответ на запрос SubscriptionPolledRefresh)

Структура тега

```

SubscriptionPolledRefreshResponse
  DataBufferOverflow
  SubscriptionPolledRefreshResult
    RcvTime
    ReplyTime
    ClientRequestHandle
    RevisedLocaleID
    ServerState
  InvalidServerSubHandles
  RItemList
    SubscriptionHandle
    Items
      ValueTypeQualifier
      ItemPath
      ItemName
      ClientItemHandle
      Timestamp
      ResultID
    DiagnosticInfo
    Value
    Quality
      QualityField
      LimitField
      VendorField
  Errors
    ID
    Text

```

XML-элементы и атрибуты

SubscriptionPolledRefreshResponse ***(Ответ на запрос SubscriptionPolledRefresh)***

Этот XML-элемент содержит всю информацию, которую сервер передает клиенту в качестве ответа на запрос SubscriptionPolledRefresh.

DataBufferOverflow ***(Переполнение буфера данных)***

Если данный атрибут имеет значение TRUE (Верно), это означает, что произошедшие изменения элементов не могут быть сохранены из-за отсутствия ресурсов. Каждый элемент содержит признак, повлияло ли на него отсутствие ресурсов.

SubscriptionPolledRefreshResult ***(Результат запроса SubscriptionPolledRefresh)***

Этот XML-элемент содержит основную информацию об ответе сервера. Он имеет тип ReplyBase. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

InvalidServerSubHandles ***(Недопустимые субдескрипторы сервера)***

Сервер признал субдескрипторы сервера недействительными.

RitemList ***(Список элементов)***

XML-элемент <RitemList> содержит значения всех элементов, если клиент в своем запросе SubscriptionPolledRefresh передал атрибут ReturnAllItems со значением TRUE (Верно). В противном случае сервер передает только изменившиеся значения.

SubscriptionHandle ***(Дескриптор подписки)***

Дескриптор XML-элемента RitemList, указанный клиентом в вызове на подписку.

Items ***(Элементы)***

XML-элемент, с помощью которого указывается отдельный элемент.

Этот элемент имеет тип ItemValue. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

Errors ***(Ошибки)***

Массив ошибок, произошедших в связи с этим ответом.

Этот элемент имеет тип OPCError. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

6.5.17 SubscriptionCancel (Отмена подписки)

Структура тега

SubscriptionCancel
ServerSubHandle
ClientRequestHandle

XML-элементы и атрибуты

SubscriptionCancel (Отмена подписки)

С помощью XML-элемента <SubscriptionCancel> клиент отменяет подписку, и сервер может освободить соответствующие ресурсы.

ServerSubHandle (Субдескриптор сервера)

Значение, присвоенное этому XML-элементу сервером, идентифицирует подписку, которая отправляет запрос.

ClientRequestHandle (Дескриптор запроса клиента)

Если этот атрибут был включен в запрос клиентом, он возвращается сервером вместе с ответом SubscriptionCancelResponse. В больших и сложных системах эта информация позволяет клиенту сопоставлять ответы запросам. Этот атрибут указывать необязательно.

6.5.18 SubscriptionCancelResponse (Ответ на запрос SubscriptionCancel)

Структура тега

SubscriptionCancelResponse
ClientRequestHandle

XML-элементы и атрибуты

SubscriptionCancelResponse (Ответ на запрос SubscriptionCancel)

С помощью этого XML-элемента сервер подтверждает обработку запроса SubscriptionCancel клиента.

ClientRequestHandle (Дескриптор запроса клиента)

Если клиент использует этот атрибут в своем запросе, сервер возвращает его в своем ответе.

6.5.19 Browse (Обзор)

Структура тега

Browse

LocaleID
ClientRequestHandle
ItemPath
ItemName
ContinuationPoint
MaxElementsReturned
BrowseFilter
ElementNameFilter
VendorFilter
ReturnAllProperties
ReturnPropertyValues
ReturnErrorText
PropertyNames

XML-элементы и атрибуты

Browse (Обзор)

XML-элемент <Browse> содержит всю информацию, необходимую для обзора иерархического пространства адресов.

LocaleID (Идентификатор языка)

Необязательный атрибут, с помощью которого клиент указывает язык для некоторых возвращаемых данных.

ClientRequestHandle (Дескриптор запроса клиента)

Если этот атрибут был включен клиентом в запрос, он возвращается сервером в ответе. В больших и сложных системах эта информация позволяет клиенту сопоставлять ответы запросам. Этот атрибут указывать необязательно.

Item Path (Путь к элементу)

Путь к элементу, с которого начинается навигация по пространству адресов. Если поступает второй запрос обзора, этот атрибут должен иметь такое же значение, что и в предшествующем запросе.

ItemName (Имя элемента)

Имя элемента, с которого начинается навигация в пространстве адресов. Если поступает второй запрос на обзор, этот атрибут должен иметь то же значение, что и в предшествующем запросе.

ContinuationPoint (Точка продолжения)

Если этот запрос на обзор не является первым, можно указать точку, начиная с которой будет возобновлен обзор.

MaxElementsReturned**(Максимальное количество возвращаемых элементов)**

Максимальное количество возвращаемых элементов.

BrowseFilter (Фильтр обзора)

Этот атрибут с типом данных browseFilter (все, ветвь, элемент) указывает, какой поднабор элементов должен быть возвращен.

ElementNameFilter (Фильтр имен элементов)

Регулярное выражение, служащее для выбора элементов.

VendorFilter (Фильтр по производителям)

Выражение, зависящее от производителя, которое служит для выбора специальной информации производителя. Его влияние на XML-элемент ElementNameFilter не определено.

ReturnAllProperties (Возвращать все свойства)

Если для данного атрибута выбрано значение TRUE (Верно), сервер будет возвращать все имеющиеся свойства для каждого возвращаемого элемента. В этом случае атрибут PropertyName (Имя свойства) игнорируется.

ReturnPropertyValues (Возвращать значения свойств)

Если для данного атрибута выбрано значение TRUE (Верно), сервер возвращает не только имена свойств, но также и их значения.

ReturnErrorText (Возвращать текст ошибки)

Если для данного атрибута установлено значение TRUE (Верно), сервер возвращает подробное описание ошибок.

PropertyNames (Имена свойств)

Имена свойств, возвращаемых для каждого элемента. Если для атрибута ReturnAllProperties выбрано значение TRUE (Верно), сервер будет возвращать все свойства, независимо от атрибута PropertyName.

6.5.20 BrowseResponse (Ответ на запрос Browse)

Структура тега

BrowseResponse
ContinuationPoint
MoreElements
BrowseResult
RcvTime
ReplyTime
ClientRequestHandle
RevisedLocaleID
ServerState
Elements
Name
Item Path
ItemName
IsItem
HasChildren
Properties
Name
Description
ItemPath
ItemName
ResultID
Value
Errors
ID
Text

XML-элементы и атрибуты

BrowseResponse (Ответ на запрос Browse)

Содержит всю информацию об ответе на запрос Browse (Обзор).

ContinuationPoint (Точка продолжения)

В этом атрибуте сервер может указать точку, начиная с которой должен быть начат обзор, инициируемый следующим запросом.

MoreElements (Превышение количества элементов)

Если сервер не поддерживает атрибут ContinuationPoint, и количество возвращаемых значений превышает значение атрибута MaxElementsReturned, сервер устанавливает этот атрибут в состояние TRUE (Верно).

BrowseResult (Результат обзора)

XML-элемент <BrowseResponse> содержит основную информацию об ответе сервера.

Этот элемент имеет тип ReplyBase. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

Elements (Элементы)

Содержит информацию об обработанных элементах древовидной структуры.

Name (Имя)

Имя элемента в пространстве имен, которое видит пользователь (например, при отображении в виде дерева).

ItemPath (Путь к элементу)

Путь к элементу. Этот атрибут не поддерживается в SIMATIC NET.

ItemName (Имя элемента)

С помощью атрибута ItemName элемент может однозначно идентифицироваться в пространстве имен сервера.

IsItem (Является элементом)

Если для данного атрибута установлено значение TRUE (Верно), этот элемент является элементом, который можно использовать в запросах на чтение, запись и подписку.

HasChildren (Имеет дочерние элементы)

Если для данного атрибута установлено значение TRUE (Верно), соответствующий элемент имеет дочерние элементы.

Properties (Свойства)

XML-элемент <Properties> содержит информацию о свойствах, которые могут быть доступны с помощью вызовов Browse (Обзор) или GetProperties (Прочитать свойства).

Этот элемент имеет тип ItemProperty. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

Errors (Ошибки)

Массив ошибок, произошедших в связи с данным запросом.

Этот элемент имеет тип OPCError. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

6.5.21 GetProperties (Прочитать свойства)

Структура тега

GetProperties

LocaleID
ClientRequestHandle
ItemPath
ReturnAllProperties
ReturnPropertyValues
ReturnErrorText
ItemIDs
ItemPath
ItemName
PropertyNames

XML-элементы и атрибуты

GetProperties (Прочитать свойства)

С помощью XML-элемента <GetProperties> можно выполнить опрос свойств.

LocaleID (Идентификатор языка)

Необязательный атрибут, с помощью которого клиент может выбрать язык для определенных возвращаемых данных.

ClientRequestHandle (Дескриптор запроса клиента)

Если этот атрибут был включен клиентом в запрос, он возвращается сервером в ответе. В больших и сложных системах эта информация позволяет клиенту сопоставлять ответы запросам. Этот атрибут указывать необязательно.

ItemPath (Путь к элементу)

Путь к элементу. Этот атрибут не поддерживается в SIMATIC NET.

ReturnAllProperties (Возвращать все свойства)

Если для данного атрибута выбрано значение TRUE (Верно), сервер будет возвращать все свойства для каждого возвращаемого элемента. В этом случае значение XML-элемента PropertyNames (Имена свойств) игнорируется.

ReturnPropertyValues (Возвращать значения свойств)

Если для данного атрибута выбрано значение TRUE (Верно), сервер возвращает не только имена свойств, но также и их значения.

ReturnErrorText (Возвращать текст ошибки)

Если для данного атрибута выбрано значение TRUE (Верно), сервер возвращает подробное описание ошибки.

ItemIDs (Идентификаторы элементов)

Содержит список элементов, параметры которых будут опрашиваться.

ItemPath (Путь к элементу)

Путь к элементу. Этот параметр не поддерживается в SIMATIC NET.

ItemName (Имя элемента)

С помощью атрибута ItemName элемент можно однозначно идентифицировать в пространстве имен сервера.

PropertyNames (Имена свойств)

Имена свойств возвращаются для каждого элемента. Если для атрибута ReturnAllProperties выбрано значение TRUE (Верно), сервер возвращает все свойства независимо от атрибута PropertyNames.

6.5.22 GetPropertiesResponse (Ответ на запрос GetProperties)

Структура тега

```
GetPropertiesResponse  
  GetPropertiesResult  
    RcvTime  
    ReplyTime  
    ClientRequestHandle  
    RevisedLocaleID  
    ServerState  
  PropertyLists  
    ItemPath  
    ItemName  
    ResultID  
  Properties  
    Name  
    Description  
    ItemPath  
    ItemName  
    ResultID  
  Value  
Errors  
  ID  
  Text
```

XML-элементы и атрибуты

GetPropertiesResponse (Ответ на запрос *GetProperties*)

Этот XML-элемент является ответом сервера на запрос *GetProperties*.

GetPropertiesResult (Результат запроса *GetProperties*)

XML-элемент `<GetPropertiesResult>` содержит основную информацию об ответе сервера.

Этот элемент имеет тип `ReplyBase`. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

PropertyLists (Списки свойств)

Для каждого элемента возвращается XML-элемент этого типа, который содержит запрошенные свойства элемента.

Для XML-элементов PropertyLists и Properties можно использовать атрибуты *Item Path*, *ItemName* и *ResultID*. Если эти атрибуты указаны в XML-элементе Properties, значения одноименных атрибутов XML-элемента PropertyLists будут перезаписаны этим XML-элементом.

ItemPath (Путь к элементу)

Путь к элементу. Этот параметр не поддерживается в SIMATIC NET.

ItemName (Имя элемента)

С помощью атрибута ItemName элемент можно однозначно идентифицировать в пространстве имен сервера.

ResultID (Идентификатор результата)

Этот атрибут содержит имя ошибки OPC, если произошла ошибка или некритичное исключение.

Properties (Свойства)

Этот XML-элемент имеет тип ItemProperty. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

Errors (Ошибки)

Массив ошибок, произошедших в связи с данным запросом.

Этот элемент имеет тип OPCError. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

6.5.23 GetStatus (Прочитать статус)

Структура тега**GetStatus***LocaleID**ClientRequestHandle*

XML-элементы и атрибуты

GetStatus (Прочитать статус)

Запрос GetStatus выполняет проверку Web-сервиса и получает в ответ специальную информацию производителя, которая недоступна для других методов OPC.

LocaleID (Идентификатор языка)

Необязательный атрибут, с помощью которого клиент может выбрать язык для определенных возвращаемых данных.

ClientRequestHandle (Дескриптор запроса клиента)

Если этот атрибут был включен клиентом в запрос, он возвращается сервером в ответе. В больших и сложных системах эта информация позволяет клиенту сопоставлять ответы запросам. Этот атрибут указывать необязательно.

Пример

```
<soap:Body>
  <GetStatus
    LocaleID="de-AT"
    xmlns="http://opcfoundation.org/webservices/XMLDA/1.0/"
  />
</soap:Body>
```

С#-интерфейс для метода GetStatus Web (Прочитать статус через Web)

В случае применения пользовательских программ на языке С#, для проверки Web-сервиса имеется следующий метод:

```
ServerStatus GetStatus(string LocaleID,
                      string ClientRequestHandle,
                      out ServerStatus);
```

6.5.24 GetStatusResponse (Ответ на запрос GetStatus)

Структура тега

```

GetStatusResponse
  GetStatusResult
    RcvTime
    ReplyTime
    ClientRequestHandle
    RevisedLocaleID
    ServerState
  Status
    StartTime
    ProductVersion
  StatusInfo
  VendorInfo
  SupportedLocaleIDs
  SupportedInterfaceVersions

```

XML-элементы и атрибуты

GetStatusResponse (Ответ на запрос GetStatus)

Содержит всю информацию об ответе на запрос GetStatus

GetStatusResult (Результат чтения статуса)

XML-элемент <GetStatusResult> содержит всю информацию об ответе сервера.

Этот элемент имеет тип ReplyBase. Информацию об атрибутах этого XML-элемента можно найти в соответствующем разделе настоящей документации.

Status (Статус)

XML-элемент <Status> содержит информацию о состоянии сервера.

StartTime (Время запуска)

Время (UTC), когда был запущен сервер.

ProductVersion (Версия продукта)

Номер версии сервера, состоящий из основного, второстепенного и встроенного номера.

StatusInfo (Информация о состоянии)

Содержит дополнительную информацию о статусе сервера и может зависеть от языка.

VendorInfo (Информация о производителе)

Строка, зависящая от производителя, содержащая дополнительную информацию о сервере. Значение этого атрибута должно включать название компании и сведения о поддерживаемых устройствах.

SupportedLocaleIDs (Идентификаторы поддерживаемых языков)

Языки, поддерживаемые сервером. Серверу не требуется поддерживать все языки всех элементов.

SupportedInterfaceVersions (Поддерживаемые версии интерфейса)

Массив строк, содержащий версии спецификации XML-DA, поддерживаемые сервером.

Словарь

AccessPath (Путь доступа)

AccessPath (Путь доступа) – это дополнительная (необязательная) информация об объектах класса OPC Item (OPC-элемент). Необходимость указания и синтаксис пути доступа зависят от конкретного сервера. Параметр AccessPath указывает (в случае необходимости) путь, по которому сервер должен читать или записывать данные переменной процесса. Параметр AccessPath может, к примеру, содержать путь или имя соединения.

ActiveState (Активное состояние)

для интерфейса Data Access (Доступ к данным): OPC-сервер может читать, записывать и контролировать переменную процесса. В процессе контроля (мониторинга) сервер проверяет, изменилось ли значение переменной процесса по истечении определенного интервала времени. Если значение изменилось, клиенту автоматически передается новое значение. Клиент может управлять мониторингом переменной процесса, используя параметр ActiveState. Атрибут ActiveState (Активное состояние) предусмотрен для каждой группы и каждого объекта OPC item (OPC-элемент). Если клиенту требуется контролировать переменную процесса, он переводит OPC-элемент, представляющий соединение с этой переменной процесса, в активное состояние (параметр ActiveState = TRUE). ActiveState группы также должен быть установлен равным TRUE. Если группа неактивна, мониторинг переменных процесса этой группы не выполняется.

CLSID

Идентификационный код для OLE-классов

COM

Чтобы объекты, реализованные на базе различных платформ или компьютеров с различными архитектурами, были совместимы друг с другом, необходимо определить правило интерпретации объекта различными платформами. Для этого служит компонентная объектная модель (COM = Component Object Model). Моделью COM определен стандарт взаимодействия между компонентами. Модель COM предоставляет возможность направления вызовов из одного процесса в другой процесс и даже на другой компьютер.

COM-библиотека

COM-библиотека – это часть операционной системы, которая управляет заносимой в реестр информацией о COM-объектах, "известных" системе, например, об OPC-серверах. При обращении клиента к COM-библиотеке выполняется поиск адресуемого объекта и соединение устанавливается в зависимости от его расположения: если адресуемый объект и клиент находятся на одном компьютере, соединение устанавливается через локальный механизм COM; если адресуемый объект и клиент находятся на разных компьютерах, соединение устанавливается через DCOM.

CSMA/CD

(Коллективный доступ с контролем несущей и обнаружением коллизий, стандарт IEEE 802.3)

Технология управления произвольными сеансами доступа к среде связи. Любой узел в любой момент времени может передавать данные, если шина в этот момент времени не используется другим узлом. Если два узла начинают передачу данных одновременно или с очень небольшой разницей во времени, когда шина свободна, возникает конфликт (так называемая коллизия) в связи с тем, что распространение сигнала занимает определенное время. Узлы "слушают" шину и обнаруживают коллизию. Каждый узел после этого приступает к передаче данных только спустя некоторое, случайным образом выбранное время ожидания. В шинах, в которых применяется технология CSMA/CD, например, в Industrial Ethernet, данные, как правило, передается со скоростью 10 Мбит/с.

DCOM

Объектная модель, служащая для передачи вызовов между различными компьютерами (машинами), получившая название DCOM (Распределенная Компонентная Объектная Модель) и встроенная в операционную систему, например, в Windows NT 4.0.

Ethernet

Технология, применяемая для создания локальных сетей (LAN) и территориальных сетей (WAN). Ethernet определяет стандартные процедуры для объединения нескольких компьютеров в сеть. Он определяет способ управления потоком данных в сетевом кабеле. В Ethernet используется технология управления доступом к среде передачи CSMA/CD (Коллективный доступ с контролем несущей и обнаружением коллизий), определенный в международном стандарте IEEE 802.3. Такой тип управления доступом к среде передачи предполагает, что рабочая станция проверяет, свободен ли канал связи, прежде чем приступить к передаче данных. Сети Ethernet базируются на шинной топологии. В качестве среды передачи данных применяются экранированные коаксиальные кабели, витые пары и волоконно-оптические кабели. В SIMATIC NET Industrial Ethernet используются стандарты Ethernet и Fast Ethernet (быстрая сеть Ethernet).

FETCH (Прочитать)

Адрес запрошенных данных передается коммуникационному партнеру, после чего коммуникационный партнер возвращает запрошенные данные. Коммуникации инициируются OPC-сервером.

FMS-коммуникации

FMS - Спецификация сообщений шины полевого уровня. В SIMATIC NET FMS-коммуникации используются в сетях PROFIBUS. FMS предоставляет сервисы для передачи структурированных данных. FMS-сервер принадлежит уровню 7 эталонной модели ISO/OSI.

GAN (Глобальная сеть)

Всемирная сеть связи, объединяющая компьютеры, расположенные на различных континентах. Для связи в таких сетях, как правило, используются спутники. Поддерживаются различные скорости передачи и классы сервисов. Передача сообщения от источника к адресу занимает значительное время. Пример: Internet.

IT-коммуникации

IT - информационные технологии. Основными средствами коммуникаций в офисах являются, как правило, электронная почта (E-mail) и Web-браузеры. Для связи, как правило, используются сети Ethernet, телефонные кабели и Internet. Программируемый контроллер SIMATIC внедряется в мир IT-коммуникаций через Industrial Ethernet. Для SIMATIC доступны средства и каналы связи, применяемые в IT-коммуникациях, поскольку в SIMATIC поддерживается протокол TCP/IP, в частности, SMTP (Простой протокол передачи почтовый сообщений) для E-mail и HTTP (Протокол передачи гипертекста) для работы с Web-браузерами.

LAN (Локальная сеть)

Локальные сети – это системы, обеспечивающие обмен данными между отдельными узлами (партнерами) по каналу связи с высокой пропускной способностью (например, по волоконно-оптическому или коаксиальному кабелю) в пределах ограниченной площади. Локальная сеть находится полностью в ведении администратора сети. Такие системы, вследствие своего "частного" характера, могут оказаться несовместимыми с сетями других типов. В общем случае, локальные сети характеризуются однородной структурой передачи данных, в которой поддерживаются скорости до 10 - 300 Мбит/с. Их можно объединять как непосредственно, так и через территориальные или всемирные сети, образуя структуры, охватывающие большую площадь. Общественные организации не предоставляют услуги, связанные с локальными сетями. Пример: Industrial Ethernet, PROFIBUS.

MAP

MAP - Протокол промышленной автоматизации. Стандарт на локальные сети, применяемый, главным образом, для промышленной автоматизации.

OCX

OCX – это управляющий элемент ActiveX. OCX – это COM-объект, который можно использовать многократно, вставляя его в приложения в качестве управляющего элемента (control).

OLE

OLE - Связь и внедрение объектов. OLE – это технология, применяемая для обмена данными между различными программами Windows. OLE устанавливает в операционной системе Windows соединение между программой-сервером и программой-контейнером. После этого объекты программы-сервера могут быть включены в документ программы-контейнера.

OPC Foundation

Организация OPC Foundation (Фонд поддержки OPC) была создана с целью определения и документирования стандарта на OPC-интерфейс. Организация OPC Foundation поддерживается фирмой SIEMENS и другими ведущими компаниями, занятыми в области автоматизации. Совместимость с Windows обеспечивается фирмой Microsoft.

OPC-клиент

Определение для OPC-совместимого приложения Windows, например, системы операторского контроля и визуализации.

OPC-элемент (OPC Item)

Класс OPC item (OPC-элемент) – это самый младший класс модели классов OPC. Объект, принадлежащий классу OPC item, представляет соединение с переменной процесса.

PG/OP-коммуникации

PG – это обозначение программатора, OP - операторской панели. PG/OP-коммуникации обеспечивают доступ к программируемым контроллерам SIMATIC из программных пакетов конфигурирования STEP 5 и STEP 7. Понятия PG/OP-коммуникации и PG/OP-протокол – это синонимы.

PROFIBUS

PROFIBUS - Шина полевого уровня и уровня процессов. Стандарт, определяющий шинную систему полевого уровня (стандарт: EN 50 170). PROFIBUS – это стандартная коммуникационная система, базирующаяся на эталонной модели ISO/OSI, которая предназначена для организации связи между открытыми системами и реализует следующие уровни эталонной модели: уровень 1 (физический уровень), уровень 2 (канальный уровень) и уровень 7 (прикладной уровень). Таким образом, стандарт PROFIBUS определяет среду передачи, метод доступа к среде передачи, а также пользовательские протоколы и интерфейс пользователя. Для передачи данных используются цифровые сигналы. Уровни 3...6 в PROFIBUS намеренно не используются, что позволяет реализовать быстрый обмен данными в режиме реального времени. Функциональность этих уровней обеспечивается интерфейсом LLI (Интерфейс нижнего уровня). LLI также используется для взаимодействия уровня 7 с уровнем 2. В качестве среды передачи (физический уровень) в PROFIBUS используется экранированная витая пара (электрические сети) или волоконно-оптический кабель (оптические сети).

PROFIBUS-PA

PROFIBUS-PA – расширение PROFIBUS-DP. В PROFIBUS-PA используются оптимизированные методы обмена данными с полевыми устройствами (например, подача питания на полевые устройства по тем же проводникам, которые используются для передачи данных, и применение в опасных зонах) при сохранении коммуникационных функций PROFIBUS-DP. PROFIBUS-PA является разновидностью PROFIBUS-DP для применения стандарта PROFIBUS во взрывоопасных зонах (там, где требуется искробезопасность) при сохранении целостности системы PROFIBUS-DP.

ProgID

Уникальное имя OPC-сервера, присвоенное ему производителем.

S7-коммуникации, протокол S7

Протокол S7 (S7-коммуникации) служит для организации связи между программируемыми контроллерами SIMATIC S7 и SIMATIC M7, а также для связи между этими системами и персональными компьютерами. Понятия "S7-коммуникации" и "протокол S7" являются синонимами.

Протокол S7 относится к канальному уровню. Он управляет потоком данных и отвечает за обнаружение, уведомление и исправление ошибок передачи. На уровне 2 определяются следующие аспекты: управление доступом к каналу связи, объединение данных в кадры и надежность передачи данных. В частности, на уровне 2 определяется возможность выбора следующих методов доступа к шине: CSMA/CD и метод передачи маркера.

SEND/RECEIVE-коммуникации

Протокол SEND/RECEIVE (SEND/RECEIVE-коммуникации) служит для надежного обмена данными между программируемыми контроллерами SIMATIC S7 и SIMATIC S5. Протокол SEND/RECEIVE для PROFIBUS принадлежит уровня 2 эталонной модели ISO/OSI, а для Industrial Ethernet – уровню 4. Понятия "SEND/RECEIVE-коммуникации" и "протокол SEND/RECEIVE" являются синонимами.

SNMP

Простой протокол управления сетью, базирующийся на UDP. Открытый протокол для управления сетями.

TCP/IP

TCP/IP - Протокол управления передачей/Межсетевой протокол. TCP – это протокол, который служит для организации, установления и разрыва соединения между двумя сетями, управления потоком данных и проверки завершения передачи данных. IP – это протокол, предназначенный для организации и адресации к данным. TCP/IP используется в сетях всемирного и локального уровня.

UTC

UTC (Универсальное координированное время) – стандартное всемирное время, ранее называвшееся Greenwich Mean Time (Среднее время по Гринвичу).

VFD (Виртуальное полевое устройство)

VFD – это модель (имитация) полевого устройства. На этапе конфигурирования OPC-сервер регистрируется в VFD.

WAN (Территориальная сеть)

Служит для объединения компьютеров, находящихся в пределах одной страны или в разных странах (или даже континентах). Для создания этих сетей используются металлические или волоконно-оптические кабели. В пределах сети используются, как правило, различные скорости передачи. Передача данных от источника адресату занимает значительное время. Сети WAN также могут подсоединяться к сетям GAN, либо сами могут быть расширены до уровня GAN. Услуги сетей WAN часто предоставляются общественными организациями. Например: сеть DATEX-P Почтового ведомства Германии.

Аварийное сообщение (тревога)

для интерфейса OPC Alarms & Events (Доступ к аварийным сообщениям и событиям через OPC): аварийное сообщение (тревога) – это событие, формируемое при возникновении внештатной ситуации.

Активный узел

Ведущее устройство, участвующее в обмене данными по сети PROFIBUS, называют активным узлом или станцией.

Асинхронность, асинхронный доступ

В случае асинхронного доступа к данным программа продолжает выполняться в процессе чтения или записи данных.

Беспроводная передача данных

Способ передачи данных (физическая среда) для PROFIBUS. Для беспроводной передачи данных в PROFIBUS используется модуль инфракрасной связи (ILM), обеспечивающий передачу данных на расстоянии 15 метров в прямой видимости.

Ведомое устройство

Ведомые устройства не могут быть инициаторами связи. Они не получают маркер и отвечают только на запросы, поступающие от ведущего устройства. В этой связи ведомые устройства являются пассивными узлами.

Ведущее устройство (master)

Ведущие устройства управляют обменом данными (коммуникациями) по шине. В каждом цикле каждое ведущее устройство однократно получает маркер, дающий право на осуществление обмена данными с ведомыми устройствами или другими ведущими устройствами. По истечении времени удержания маркера узел передает маркер следующему ведущему устройству. Ведущее устройство называют активным узлом.

Виртуальное устройство

Также обозначается аббревиатурой VFD (Виртуальное полевое устройство).

VFD – это модель (имитация) полевого устройства. На этапе конфигурирования OPC-сервер регистрируется в VFD.

Внутрипроцессный сервер (In-process server)

Внутрипроцессный сервер обращается к локальному серверу через пользовательский OPC-интерфейс. Внутрипроцессный сервер располагается в том же пространстве процесса, что и клиент, и доступен только для этого процесса. В общем случае, внутрипроцессные серверы реализуются в виде библиотек DLL. OPC-интерфейс автоматизации реализуется как внутрипроцессный сервер.

Значение

При запросе значения переменной выполняется обращение к данным процесса, в результате чего возвращается самое последнее значение. Это могут быть показания датчика, параметр управления, информация о состоянии или статус сетевого соединения.

Интерфейсный модуль

Средство подключения устройств или компьютеров в коммуникационную сеть. В этом случае устройство подключается к сети не через свои встроенные порты, а с применением интерфейсных модулей (IM) или коммуникационных процессоров (CP).

Канонический тип данных

Канонический тип данных – это исходный тип данных переменной, в котором она представляется на стороне источника и в котором она поставляется OPC-сервером.

Категории

для Alarms & Events (Аварийные сообщения и события): введение категорий позволяет объединять события, поставляемые OPC-сервером событий, в отдельные группы и выбирать события по категориям. Определение событий осуществляется производителем OPC-сервера событий. В спецификациях OPC определен список рекомендуемых категорий событий.

Кеш

для Data Access (Доступ к данным): под кешом понимается промежуточный буфер, предназначенный для размещения всех переменных, связанных с OPC-элементами. Для всех пользователей используется один общий кеш. Это означает, что, если к одной переменной обращается одновременно несколько пользователей, значение в кеш заносится только один раз. Чтобы операции чтения (чтения и обновления) читали данные из кеша сервера, следует указывать опцию OPC_DS_CACHE.

Коаксиальный кабель

Среда передачи данных, применяемая на уровне отдельных ячеек автоматизации и уровне общезаводского управления (сети PROFIBUS и Industrial Ethernet). Коаксиальный кабель состоит из нескольких слоев: расположенный по центру проводник окружен изолирующей оболочкой, изолирующий материал заключен в оплетку, оплетка укрыта изолирующей оболочкой. Коаксиальные кабели служат для передачи данных с использованием электрических сигналов, когда основным требованием к системе связи является простота прокладки кабеля и дальнейшего расширения системы.

Код качества (Quality)

Содержит информацию о надежности (достоверности) прочитанной переменной.

Коммуникации (связь)

Под коммуникациями (связью) понимается обмен данными между двумя коммуникационными партнерами (устройствами или модулями).

Коммуникации на полевого уровне и уровне процесса.

Коммуникации полевого уровня и уровня процесса служат для подключения исполнительных устройств и датчиков к программируемым логическим контроллерам. Обмен сигналами процесса с исполнительными устройствами и датчиками может производиться как циклически, так и индивидуально, с применением управляющих команд. Для коммуникаций полевого уровня и уровня процесса в SIMATIC NET предусмотрены протоколы PROFIBUS–DP и AS–Interface. Для автоматизации зданий используется сеть EIB.

Коммуникационный процессор

Процессор для организации коммуникаций в пределах систем автоматизации. Коммуникационный процессор служит для управления процессом связи.

Конфигурирование

На этапе конфигурирования для виртуального устройства настраиваются системные параметры, зависящие от протокола.

Логическое состояние (статус)

Логическое состояние – это часть специальной информационной переменной протокола S7, которая предоставляет информацию о поддерживаемых сервисах. Для передачи данных используется тип VT_BSTR.

Локальный сервер

Локальный сервер работает на том же компьютере, что и клиент. Как любое другое независимое приложение, он обладает собственным пространством процесса и пространством имен. В частности, OPC-сервер SIMATIC NET является локальным сервером. Он реализуется в виде EXE-файла.

Метка времени

Метка времени позволяет, к примеру, узнать время, в которое было получено значение переменной.

Модель классов

В спецификации OPC отдельные существующие интерфейсы и их методы отнесены к различным классам. Такая структура получила название модели классов. Каждый класс содержит несколько объектов и определяет, какими методами и свойствами должны обладать эти объекты как представители определенного класса.

Модель классов OPC

В спецификации OPC интерфейсы и их методы сгруппированы в три иерархических класса: OPC Server (OPC-сервер), OPC Group (OPC-группа) и OPC Item (OPC-элемент). Наивысшее положение в модели занимает класс OPC Server, который обладает различными атрибутами, содержащими сведения о состоянии и версии объекта OPC Server. Класс OPC Group управляет OPC-элементами. Создавая OPC-группы, клиент может объединять отдельные OPC-элементы по ряду признаков и выполнять над созданными группами определенные операции. Класс OPC items (OPC-элемент) представляет соединение с переменной процесса. Объект класса OPC Server является разновидностью контейнера для объекта класса OPC group. Последний, в свою очередь, является контейнером для объектов класса OPC item.

Обмен данными

Обмен данными между программируемыми контроллерами или между программируемыми контроллерами и компьютерами.

Опрос (Polling)

Под опросом понимается циклический обмен данными между DP-станциями. Ведущее устройство DP, получившее маркер ("производительный" режим ведущего устройства DP), передает принадлежащим ему ведомым устройствам DP кадры циклического опроса. Каждому ведомому устройству DP отправляется отдельный кадр опроса.

Оптическая передача данных

Способ передачи данных в сетях PROFIBUS и Industrial Ethernet. В этом случае для передачи данных используются оптические сигналы, передаваемые по волоконно-оптическому кабелю с волокнами из стекла или пластика. Имеются кабели как для применения внутри зданий, так и для прокладки снаружи зданий.

Пассивный узел

Ведомое устройство, участвующее в обмене данными по сети PROFIBUS, называется пассивным узлом или пассивной станцией.

Передача данных с помощью электрических сигналов (передача данных по медному кабелю)

Способ передачи данных, применяемый в сетях PROFIBUS и Industrial Ethernet. Стандартной средой для передачи данных в этом случае является экранированная витая пара. Могут также использоваться следующие типы кабелей: кабель с PE-оболочкой, кабель, не содержащий галоген, подземный кабель, кабель в исполнении для применения в опасных зонах, кабели для применения в шлейфах и для волочения.

Переменная

Это метка, вместо которой в режиме выполнения вставляется текущее значение.

ПЛК (PLC)

Программируемый логический контроллер. Связующее звено между уровнем датчиков и уровнем производственного оборудования, используется также для организации связи и обмена данными между пользователем и компонентами производственного процесса.

Подробное локальное состояние

Подробное локальное состояние – часть специальной информационной переменной протокола S7, предоставляющая подробные сведения о состоянии локального VFD-устройства. Для интерпретации возвращаемого значения требуется описание локального VFD-устройства. Данные передаются с типом VT_UI2.

Полевой уровень

Полевой уровень является одним из уровней автоматизации. На этом уровне автоматизируемые производственные процессы (установки) связываются с программируемыми контроллерами. Полевые устройства выполняют измерения, формируют сигналы и передают команды уровня ячейки автоматизации производственному оборудованию или процессу. Объемы данных, передаваемые на этом уровне, относительно малы. Типичной на этом уровне является иерархическая структура коммуникаций, в которой несколько полевых устройств обмениваются данными с одним ведущим устройством.

Принцип "ведущий – ведомый" (master–slave)

Метод доступа в шине, заключающийся в том, что активный узел шины PROFIBUS (ведущее устройство) централизованно управляет всем трафиком данных в шине. Ведущее устройство передает данные периферийным устройствам (ведомым устройствам) и посылает им запросы на передачу данных ведомому устройству. Ведомое устройство получает информацию от ведущего устройства и выставляет его на свои выходы (управляет периферийными устройствами). Принцип "ведущий – ведомый", главным образом, применяется на полевом уровне (например, в сети PROFIBUS–DP).

Принцип "клиент-сервер"

Такая организация обмена данными, при которой клиент передает коммуникационные задания, а сервер их выполняет.

Программируемый контроллер SIMATIC

Программируемый контроллер SIMATIC – это автономная компонуемая система, предназначенная для решения любых задач промышленной автоматизации. ПЛК SIMATIC выполняет все функции, необходимые для реализации задач автоматизации, ранее требовавших применения различных специальных систем. Устройства других производителей могут быть включены в систему через стандартные интерфейсы. ПЛК SIMATIC применяется в самых различных отраслях промышленности, например, в автомобилестроении, в нефтехимии, в строительстве, в электронной промышленности, в энергетике, в системах диспетчеризации транспорта, в системах водоснабжения и в системах утилизации отходов. Существуют различные серии программируемых контроллеров SIMATIC: SIMATIC S7/M7/C7, SIMATIC PG/PC, SIMATIC HMI, SIMATIC S5 и SIMATIC 505.

Промышленная витая пара (ITP)

Среда передачи данных, применяемая на уровнях отдельных ячеек автоматизации и уровне общезаводского управления (сеть Industrial Ethernet). Витая пара состоит из двух пар проводов, перевитых между собой, что обеспечивает высокую устойчивость к воздействию помех. Витую пару, применяемую в сетях SIMATIC NET, называют промышленной витой парой (Industrial Twisted-Pair = ITP). Этот кабель применяется для "электрической" передачи данных, в частности, для создания структурированных кабельных систем на промышленном предприятии.

Протокол

Протокол – набор правил, согласованных между двумя коммуникационными партнерами, позволяющий им обмениваться данными. В соответствии с эталонной моделью ISO/OSI для каждого уровня устанавливаются соответствующие протоколы. Помимо прочего, протоколом определяются следующие аспекты обмена данными: режим связи, установление соединения, защита данных и скорость передачи.

Протокол DP

DP - децентрализованная периферия (система распределенного ввода/вывода). В SIMATIC NET протокол DP применяется в сетях PROFIBUS. Протокол DP служит для организации обмена данными с распределенными устройствами ввода/ вывода.

Протокол FMS

FMS - Спецификация сообщений шины полевого уровня. В SIMATIC NET FMS-коммуникации используются в сетях PROFIBUS. FMS предоставляет сервисы для передачи структурированных данных. FMS-сервер принадлежит уровню 7 эталонной модели ISO/OSI.

Протокол PG/OP

PG – это обозначение программатора, OP - операторской панели. PG/OP-коммуникации обеспечивают доступ к программируемым контроллерам SIMATIC из программных пакетов конфигурирования STEP 5 и STEP 7. Понятия PG/OP-коммуникации и PG/OP-протокол – это синонимы.

Протокол SEND/RECEIVE

Протокол SEND/RECEIVE (SEND/RECEIVE-коммуникации) служит для надежного обмена данными между программируемыми контроллерами SIMATIC S7 и SIMATIC S5. Протокол SEND/RECEIVE для PROFIBUS принадлежит уровню 2 эталонной модели ISO/OSI, а для Industrial Ethernet – уровню 4. Понятия "SEND/RECEIVE-коммуникации" и "протокол SEND/RECEIVE" являются синонимами.

Протокол SR

Протокол SEND/RECEIVE (SEND/RECEIVE-коммуникации) служит для надежного обмена данными между программируемыми контроллерами SIMATIC S7 и SIMATIC S5. Протокол SEND/RECEIVE для PROFIBUS принадлежит уровню 2 эталонной модели ISO/OSI, а для Industrial Ethernet – уровню 4. Понятия "SEND/RECEIVE-коммуникации" и "протокол SEND/RECEIVE" являются синонимами.

Синхронность, синхронный доступ

В режиме синхронного доступа выполнение программы прекращается до тех пор, пока переменная не будет прочитана или записана.

Событие

для Alarms & Events (Аварийные сообщения и события): под событием понимается любое происшествие, которое может представлять интерес для клиента. Хотя события также могут формироваться по выполнению определенных условий, их необязательно привязывать к условиям. Примером событий, не связанных с выполнением условий, являются сообщения об ошибках, поступающие из системы связи.

Соединение "точка-точка"

Соединение "точка-точка" (point-to-point) позволяет организовать связь по последовательному интерфейсу. Это соединение можно устанавливать между двумя программируемыми контроллерами или между программируемым контроллером и компьютером.

Среда передачи

Для обмена данными используются различные среды передачи (физические каналы связи). Различные каналы связи отличаются между собой протяженностью, помехоустойчивостью и скоростью передачи данных. Далее перечислены возможные каналы связи в порядке возрастания их сложности и функциональности: незранированный неперевитый двухпроводный кабель (например, для AS-интерфейса), незранированная промышленная витая пара (например, для EIB), экранированная промышленная витая пара (например, для PROFIBUS), коаксиальный кабель (например, для Industrial Ethernet), триаксильный кабель (например, для PROFIBUS), волоконно-оптический кабель (для Industrial Ethernet и PROFIBUS).

Стекланный волоконно-оптический кабель

Среда передачи данных, применяемая на уровнях отдельных ячеек автоматизации и уровне общезаводского управления в сетях PROFIBUS и Industrial Ethernet. Стекловолоконные кабели и волоконно-оптические кабели применяются для оптической передачи данных, в частности, когда важно реализовать гальваническую развязку и обеспечить высокую устойчивость к воздействию электромагнитных помех.

Топология

Топология описывает структуру сети. Она определяет способ подключения устройств или компьютеров в сеть (к каналу связи). Возможны следующие топологии: линейная топология (шина), звезда, кольцо, резервированное кольцо и дерево.

Топология "дерево"

Сеть, имеющая древовидную структуру. Все узлы подключаются к сети по кабелю. Древовидная сеть образуется из отдельных шин различной длины.

Топология "звезда"

Сеть, имеющая форму звезды. Все узлы соединены по сети с сервером, который организует обмен данными. Выход из строя сервера обозначает выход из строя всей сети. Выход из строя одного узла не приводит к прекращению работы сети.

Топология "кольцо"

Сеть, имеющая форму кольца. В этой топологии каждый узел имеет равно по одному узлу слева и справа от себя. Данные переходят по кольцу от передающего узла к принимающему узлу. Данные во всей сети передаются только в одном направлении. Выход из строя одного узла приводит к выходу из строя всей сети.

Топология "резервированное кольцо"

Разновидность сети с шинной топологией. В топологии резервированного кольца шина замыкается сама на себя, образуя кольцо. В качестве среды передачи используется волоконно-оптический кабель. В топологии резервированного кольца связь не прекращается даже в случае возникновения неисправности в сети. В случае выхода из строя коммуникационного узла или обрыва волоконно-оптического кабеля сеть в течение нескольких миллисекунд реконфигурируется, превращаясь в полнофункциональную шину.

Триаксиальный кабель

Также известен как экранированный двухпроводный кабель. Служит как средство передачи данных, применяемое на уровне отдельных ячеек автоматизации и общезаводском уровне управления (сети PROFIBUS и Industrial Ethernet). Триаксиальный кабель – это экранированный коаксиальный кабель. В отличие от обычного кабеля, в нем предусмотрена дополнительная наружная герметизированная алюминиевая оболочка. Триаксиальные кабели служат для передачи данных с использованием электрических сигналов. Они особенно хорошо подходят для применения в промышленных условиях, поскольку они защищены от воздействия внешних помех благодаря наличию заземляемого экрана.

Удаленный сервер

Удаленный сервер располагается на другом компьютере. Клиент может получить к нему доступ через сетевое соединение.

Уровень 1 эталонной модели ISO/OSI

Этот уровень называют физическим уровнем. Он отвечает за "прозрачную" передачу неинтерпретируемого потока битов в том порядке, в котором они поступают из уровня 2. На уровне 1 определяются следующие параметры: тип передачи данных, физическая среда передачи данных, скорость передачи, а также электрические, механические, функциональные характеристики кабеля. В частности, на уровне 1 определяется один из возможных кабелей: коаксиальный кабель, триаксиальный кабель, волоконно-оптический кабель и витая пара.

Уровень 2 эталонной модели ISO/OSI

Уровень 3 эталонной модели ISO/OSI

Этот уровень называют сетевым уровнем. Он отвечает за доставку данных от одного узла к другому (от передатчика к приемнику). На уровне 3 определяются следующие параметры: адреса для межсетевого обмена, маршрутизация, если канал связи состоит из нескольких систем, данные через которые проходят транзитом, а также метод управления потоком. В частности, на уровне 3 определяется, каким образом реализуется связь между двумя подсетями.

Уровень 4 эталонной модели ISO/OSI

Этот уровень называют транспортным уровнем. Он предоставляет пользователю надежное соединение. На уровне 4 определяются следующие параметры: установление соединения, передача данных, разрыв соединения, повторная передача пакетов, организация последовательностей пакетов и сборка пакетов. В частности, на уровне 4 определяется надежность передачи пакетов данных.

Уровень 5 эталонной модели ISO/OSI

Этот уровень называют сеансовым уровнем и предназначен он для синхронизации данных перед их передачей. На уровне 5 определяются следующие параметры: как устанавливается соединение, как оно поддерживается и разрывается. Повтор передачи в определенной точке синхронизации. В частности, на уровне 5 определяется способ координирования сеансов передачи данных.

Уровень 6 эталонной модели ISO/OSI

Этот уровень называют уровнем представления. Очень часто участники коммуникаций "разговаривают" на разных языках. Прежде чем происходит обмен данными, уровень 6 переводит языки различных систем на единый язык, использующий абстрактный синтаксис. На уровне 6 определяются следующие параметры: представление данных и трансляция стандартных форм представления систем связи в специальные формы представления отдельных устройств. В частности, на уровне 6 определяется реализация общего языка для обмена данными.

Уровень 7 эталонной модели ISO/OSI

Этот уровень называют прикладным уровнем. Он предоставляет специальные сервисы (службы) для различных прикладных задач связи. На уровне 7 описываются коммуникационные сервисы для различных прикладных задач. В частности, на уровне 7 определяется предоставление коммуникационных сервисов, например: чтение/запись и пуск/стоп.

Уровень автоматизации

Вся система автоматизации логически разделяется на отдельные уровни, на каждом из которых выполняются определенные задачи системы. Различают следующие уровни системы автоматизации: уровень управления производством (общезаводской уровень), уровень отдельной ячейки автоматизации (автоматизация отдельного цеха), полевой уровень и уровень исполнительных устройств и датчиков.

Уровень исполнительных механизмов и датчиков

Уровень исполнительных механизмов и датчиков (Actuator–sensor level) является одним из уровней системы автоматизации. На уровне исполнительных механизмов и датчиков ведущее устройство обменивается данными с исполнительными механизмами и датчиками, включенными в сеть этого ведущего устройства. Главным свойством уровня исполнительных механизмов и датчиков является высокая скорость обмена данными небольшого объема (несколько битов).

Уровень отдельной ячейки автоматизации (cell level)

Один из уровней автоматизации. На этом уровне выполняются отдельно задачи оптимизации и автоматизации. На уровне отдельной ячейки автоматизации (отдельного производственного цеха) организуется связь между программируемыми контроллерами, персональными компьютерами и устройствами операторского управления и визуализации.

Уровень управления производством (общезаводской уровень управления)

Уровень управления производством – это один из уровней автоматизации. На этом уровне выполняется широкий круг задач управления, охватывающий весь процесс производства в целом (функции менеджмента). К таким задачам относится хранение значений переменных процесса, обработка и оптимизация, а также анализ и вывод результатов в виде отчетов. Требуемые данные собираются, например, от нескольких производственных цехов или фабрик и обрабатываются централизованно. На уровне управления производством также возможно обращаться к территориально разнесенным участкам (находящимся в разных точках планеты). Количество узлов может достигать до нескольких тысяч.

Ускоренная отправка данных (Expedited data)

Ускоренная передача небольших объемов данных. Ускоренная передача данных поддерживается только ISO-транспортом и не поддерживается TCP/IP.

Физическое состояние

Часть информационной переменной протокола S7, предоставляющая сведения о работоспособности физического устройства. Имеет тип данных VT_BSTR.

Шина (шинная топология)

Под шиной понимается сеть с линейной топологией (магистраль). Такая топология подразумевает объединение всех узлов в единую сеть, шину. Каждый узел может обмениваться данными с любым другим узлом, но не одновременно. В любой момент времени только один узел может передавать данные, а остальные узлы в это время могут только принимать данные. Механизм управления обменом данными называют методом доступа в сеть.

Эталонная модель ISO/OSI

Эталонная модель ISO/OSI определяет семь логических уровней, описывающих разные аспекты обмена данными между коммуникационными партнерами в открытой системе. Уровни располагаются друг над другом (самым высоким является уровень 7), каждый уровень базируется на нижестоящем уровне. Эталонная модель ISO/OSI используется, к примеру, для описания системы передачи данных, метода доступа к среде передачи или правил установления соединений. Взаимодействие происходит только между соседними уровнями. Конкретный способ реализации отдельных уровней эталонной моделью не устанавливается. Предусмотрены следующие уровни: уровень 1 (физический уровень), уровень 2 (канальный уровень), уровень 3 (сетевой уровень), уровень 4 (транспортный уровень), уровень 5 (сеансовый уровень), уровень 6 (уровень представления), уровень 7 (прикладной уровень).