ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р МЭК 61131-9— 2017

КОНТРОЛЛЕРЫ ПРОГРАММИРУЕМЫЕ

Часть 9

Одноточечный интерфейс цифровой связи для небольших датчиков и исполнительных устройств

(IEC 61131-9:2013, IDT)

Издание официальное

Москва
Стандартинформ
2017

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Негосударственным образовательным частным учреждением дополнительного профессионального образования «Новая инженерная школа» (НОЧУ «НИШ») на основе собственного перевода на русский язык англоязычной версии указанного в пункте 4 стандарта, который выполнен Российской комиссией экспертов МЭК/ТК 65 и Федеральным государственным унитарным предприятием «Всероссийский научно-исследовательский институт стандартизации» (ВНИИНМАШ)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 306 «Измерения и управление в промышленных процессах»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 14 сентября 2017 г. № 1124-ст

4 Настоящий стандарт идентичен международному стандарту МЭК 61131-9:2013 «Контроллеры программируемые. Часть 9. Одноточечный интерфейс цифровой связи для небольших датчиков и исполнительных устройств» [IEC 61131-9:2013 «Programmable controllers — Part 9: Single-drop digital communication interface for small sensors and actuators (SDCI)», IDT].

Международный стандарт разработан Техническим комитетом МЭК ТК 65 «Измерение, управление и автоматизация в промышленных процессах».

При применении настоящего стандарта рекомендуется применять вместо ссылочных международных стандартов соответствующие им национальные стандарты, сведения о которых приведены в дополнительном приложении ДА

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.gost.ru)

© Стандартинформ, 2017

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины, определения, обозначения, сокращения и условные обозначения . . . 3
3.1 Термины и определения . . . 3
3.2 Обозначения и сокращения . . . 5
3.3 Условные обозначения . . . 7
4 Обзор одноточечного интерфейса цифровой связи SDCI (IO-Link™) . . . 9
4.1 Назначение технологии . . . 9
4.2 Позиционирование в иерархии средств автоматизации . . . 9
4.3 Проводка, соединители и электропитание . . . 10
4.4 Возможности связи SDCI . . . 10
4.5 Роль Ведущего узла . . . 12
4.6 Конфигурирование SDCI . . . 13
4.7 Отображение на полевые шины . . . 13
4.8 Структура стандарта . . . 14
5 Физический уровень (PL) . . . 14
5.1 Общие положения . . . 14
5.2 Услуги физического уровня . . . 15
5.3 Передатчик/приемник . . . 18
5.4 Источник питания . . . 25
5.5 Среда передачи данных . . . 26
6 Стандартный ввод и вывод (SIO) . . . 29
7 Канальный уровень (DL) . . . 29
7.1 Общие положения . . . 29
7.2 Услуги канального уровня . . . 31
7.3 Протокол канального уровня . . . 46
8 Прикладной уровень (AL) . . . 80
8.1 Общие положения . . . 80
8.2 Услуги прикладного уровня . . . 81
8.3 Протокол прикладного уровня . . . 89
9 Модуль управления системой (SM) . . . 98
9.1 Общие положения . . . 98
9.2 Модуль управления системой на Ведущем узле . . . 98
9.3 Модуль управления системой на Устройстве . . . 113
10 Устройство . . . 126
10.1 Обзор . . . 126
10.2 Обмен Данными процесса (PDE) . . . 127
10.3 Менеджер параметров (PM) . . . 128
10.4 Хранилище данных (DS) . . . 135
10.5 Диспетчер Событий (ED) . . . 138
10.6 Характеристики Устройства . . . 138
10.7 Проектные нормы и ограничения Устройства . . . 140
10.8 Описание устройства ввода/вывода (IODD) . . . 142

10.9 Диагностика Устройства . . . 142
10.10 Возможности подключения Устройства . . . 145
11 Ведущий узел . . . 145
11.1 Обзор . . . 145
11.2 Менеджер конфигурации (CM) . . . 148
11.3 Хранилище данных (DS) . . . 153
11.4 Обмен Данными запроса (ODE) . . . 160
11.5 Блок диагностики (DU) . . . 161
11.6 Обмен Данными процесса (PDE) . . . 162
11.7 Инструмент конфигурирования порта и Устройства (PDCT) . . . 164
11.8 Приложение шлюза . . . 165
Приложение A (обязательное) Кодирование, временные ограничения и ошибки . . . 169
Приложение B (обязательное) Параметры и команды . . . 188
Приложение C (обязательное) Типы ошибок (ошибки ISDU) . . . 204
Приложение D (обязательное) Коды ошибок (диагностическая информация) . . . 207
Приложение E (обязательное) Типы данных . . . 211
Приложение F (обязательное) Структура объекта данных Хранилища данных . . . 221
Приложение G (обязательное) Соответствие Ведущего узла и Устройства . . . 222
Приложение H (справочное) Вероятности остаточных ошибок . . . 227
Приложение I (справочное) Примерная последовательность передачи ISDU . . . 229
Приложение J (справочное) Рекомендуемые методы определения изменения параметров . . . 233
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации . . . 234
Библиография . . . 235

Введение

0.1 Общие положения

Настоящий стандарт является частью серии стандартов на программируемые контроллеры и связанные с ними периферийные устройства и должен применяться совместно с другими частями серии.

В том случае, если имеются противоречия между настоящим стандартом и другими стандартами МЭК (за исключением стандартов, устанавливающих требования по безопасности), следует руководствоваться положениями настоящего стандарта в области программируемых контроллеров и связанных с ними периферийных устройств.

Более широкое применение микроконтроллеров, встроенных в доступные по цене датчики и исполнительные устройства, предоставило возможности для добавления диагностических и конфигурационных данных с целью поддержки возрастающих требований к приложениям.

Управление приводом в технологии SDCI (IO-Link™)[1)] нуждается в таких доступных по цене датчиках и исполнительных устройствах для обмена диагностическими и конфигурационными данными с контроллером [программируемым контроллером (PC) или программируемым логическим контроллером (PLC)], применяя недорогую технологию цифровой связи при одновременном обеспечении обратной совместимости с токовыми сигналами ввода/вывода.

В концепциях полевой шины технология SDCI определяет многофункциональный интерфейс для подключения датчиков и исполнительных устройств к Ведущему узлу, который может комбинироваться со шлюзами для образования узла удаленных входов и выходов полевой шины.

Любое Устройство, совместимое с SDCI, может быть подключено к любому доступному интерфейсному порту Ведущего узла. Устройство, совместимое с SDCI, выполняет преобразование физического сигнала в цифровую форму в Устройстве. Затем Устройство направляет результат прямо в стандартный формат, применяя «кодовую коммутацию» в линии передачи сигналов ввода/вывода 24 В и удаляя, таким образом, необходимость в различных модулях цифрового ввода, цифрового вывода, аналогового ввода, аналогового вывода и множестве различных кабелей.

Физическое соединение производится из точки в точку от каждого Устройства к Ведущему узлу, применяя три провода длиной не более 20 м. Физический интерфейс SDCI имеет обратную совместимость с обычной сигнальной линией 24 В, установленной в МЭК 61131-2. Поддерживаются скорости передачи 4,8; 38,4 и 230,4 кбит/с.

Ведущий узел интерфейса SDCI обнаруживает, идентифицирует и управляет Устройствами, подключенными к его портам.

Инструментальные средства позволяют ассоциировать Устройства с их соответствующими описаниями устройства ввода/вывода (IODD) и их конфигурациями для удовлетворения требований приложения.

Технология SDCI определяет три различных уровня диагностических возможностей: для немедленного реагирования автоматизированными средствами на этапе изготовления; для среднесрочного реагирования путем вмешательства оператора; для долговременного ввода в эксплуатацию и технического обслуживания с использованием расширенной диагностической информации.

Структура настоящего стандарта описана в подразделе 4.8.

Соответствие настоящему стандарту может подтверждаться только в том случае, если выполняются требования приложения G.

Общепринятые термины установлены в МЭК 61131-1 или в серии МЭК 60050. Конкретные термины устанавливаются в каждой части.

0.2 Патентная декларация

Международная электротехническая комиссия (МЭК) обращает внимание на то заявление, что соответствие настоящему стандарту может включать использование патентов, касающихся интерфейса

[1)] IO-Link™ является торговой маркой «IO-Link Consortium». Данная информация приводится для удобства пользователей настоящим стандартом и не требует от МЭК подтверждения владельца торговой марки держателя или любого из его продуктов. Соответствие настоящему стандарту не требует использования зарегистрированных торговых знаков IO-Link™. Использование зарегистрированных торговых знаков IO-Link™ требует разрешения от «IO-Link Consortium».

двухточечной последовательной связи для небольших датчиков и исполнительных устройств в следующей редакции, где обозначение [xx] указывает на владельца патентного права:

DE 10030845B4 EP 1168271B1 US 6889282B2	[AB]	Система коммутации полевых шин для исполнительных устройств и датчиков
EP 1203933 B1	[FE]	Сенсорное устройство для измерения по меньшей мере одной переменной
DE 10 2004 035 831.1	[SI]	Оперативное состояние вычислительной системы проверяется сравнением фактических параметров с эталонными значениями и модификациями операционной системы при необходимости
DE 102 119 39 A1 US 2003/0200323 A1	[SK]	Соединительное устройство для присоединения устройств к системе шин

МЭК не дает никаких заключений относительно документального подтверждения, юридической силы и объема применения настоящих патентных прав.

Владельцы настоящих патентных прав заверили МЭК, что они готовы в добровольном порядке переуступить лицензии либо бесплатно, либо на приемлемых и справедливых условиях заявителям по всему миру. В этом отношении заявления владельцев данных патентных прав зарегистрированы в МЭК.

Дополнительная информация доступна в следующих источниках:

[AB]	ABB AG Heidelberg Germany — Германия
[FE]	Festo AG Esslingen Germany — Германия
[SI]	Siemens AG Otto-Hahn-Ring 6 81739 Munich Germany — Германия
[SK]	Sick AG Waldkirch Germany — Германия

Обращается внимание на возможность того, что некоторые элементы настоящего стандарта могут являться объектом патентных прав, отличных от установленных выше. МЭК не несет ответственности за установление частично либо полностью таких патентных прав.

ИСО (gost.e-ecolog.ru и МЭК (gost.e-ecolog.ru поддерживают оперативные базы данных патентов, имеющих отношение к их стандартам. Пользователям рекомендуется обращаться к базам данных для получения самой последней информации, относящейся к патентам.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

КОНТРОЛЛЕРЫ ПРОГРАММИРУЕМЫЕ

Часть 9

Одноточечный интерфейс цифровой связи для небольших датчиков и исполнительных устройств

Programmable controllers. Part 9. Single-drop digital communication interface for small sensors and actuators (SDCI)

Дата введения — 2018—09—01

1 Область применения

Настоящий стандарт устанавливает технологию одноточечного интерфейса цифровой связи SDCI для небольших датчиков и исполнительных устройств (более известных под названием IO-Link™), которая расширяет традиционные интерфейсы цифрового ввода и вывода, определенные в МЭК 61131-2, к двухточечной линии связи. Данная технология делает возможными передачу параметров Устройствам и доставку диагностических данных от Устройств к системе автоматизации.

Данная технология в основном предназначена для применения в системах автоматизации производства с небольшими датчиками и исполнительными устройствами, которые включают в себя небольшие и экономически эффективные микроконтроллеры.

Настоящий стандарт устанавливает услуги связи SDCI и протокол (физический уровень, канальный уровень и прикладной уровень в соответствии с эталонной моделью ISO/OSI) как для Ведущих узлов SDCI, так и для Устройств.

Настоящий стандарт также устанавливает требования к испытаниям на электромагнитную совместимость. Настоящий стандарт не распространяется на интерфейсы связи или системы, включающие многоточечные соединения, а также интеграцию SDCI в системах более высокого уровня, таких как полевые шины.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие документы. Для датированных ссылок применяют только указанное издание ссылочного документа. Для недатированных ссылок применяют последнее издание ссылочного документа (включая все его изменения).

IEC 60947-5-2, Low-voltage switchgear and controlgear — Part 5-2: Control circuit devices and switching elements — Proximity switches (Низковольтная аппаратура распределения и управления. Часть 5-2. Аппаратура для цепей управления и коммутирующие элементы. Бесконтактные переключатели)

IEC 61000-4-2, Electromagnetic compatibility (EMC) — Part 4-2: Testing and measurement techniques — Electrostatic discharge immunity tes [Электромагнитная совместимость (ЭМС).Часть 4-2. Методы испытаний и измерений. Испытания устойчивости к электростатическим разрядам]

IEC 61000-4-3, Electromagnetic compatibility (EMC) — Part 4-3: Testing and measurement techniques — Radiated, radio-frequency, electromagnetic field immunity test [Электромагнитная совместимость

Издание официальное

(ЭМС). Часть 4-3. Методы испытаний и измерений. Испытания устойчивости к радиоактивному излучению, радиочастотам и электромагнитному полю]

IEC 61000-4-4, Electromagnetic compatibility (EMC) — Part 4-4: Testing and measurement techniques — Electrical fast transient/burst immunity test [Электромагнитная совместимость (ЭМС). Часть 4-4. Методы испытаний и измерений. Испытания устойчивости к кратковременным выбросам напряжения и импульсным помехам]

IEC 61000-4-5, Electromagnetic compatibility (EMC) — Part 4-5: Testing and measurement techniques — Surge immunity test [Электромагнитная совместимость (ЭМС). Часть 4-5. Методы испытаний и измерений. Испытания устойчивости к динамическим изменениям напряжения]

IEC 61000-4-6, Electromagnetic compatibility (EMC) — Part 4-6: Testing and measurement techniques — Immunity to conducted disturbances, induced by radio-frequency fields [Электромагнитная совместимость (ЭМС). Часть 4-6. Методы испытаний и измерений. Устойчивость к кондуктивным помехам, наведенным радиочастотными электромагнитными полями]

IEC 61000-4-11, Electromagnetic compatibility (EMC) — Part 4-11: Testing and measurement techniques — Voltage dips, short interruptions and voltage variations immunity tests [Электромагнитная совместимость (ЭМС). Часть 4-11. Методы испытаний и измерений. Испытания устойчивости к кратковременным падениям напряжения, кратковременным прерываниям электроснабжения и перепадам напряжения]

IEC 61000-6-2, Electromagnetic compatibility (EMC) — Part 6-2: Generic standards — Immunity for industrial environments [Электромагнитная совместимость (ЭМС). Часть 6-2. Общие стандарты. Устойчивость к промышленным окружающим средам]

IEC 61000-6-4, Electromagnetic compatibility (EMC) — Part 6-4: Generic standards — Emission standard for industrial environments [Электромагнитная совместимость (ЭМС). Часть 6-4. Общие стандарты. Нормы выбросов в промышленных окружающих средах]

IEC 61076-2-101, Connectors for electronic equipment — Product requirements — Part 2-101: Circular connectors — Detail specification for M12 connectors with screw-locking (Соединители для электронного оборудования. Требования к изделию. Часть 2-101. Круглые соединители. Детальные технические условия для соединителей М12 с контровыми устройствами)

IEC 61131-1, Programmable controllers — Part 1: General information (Контроллеры программируемые. Часть 1. Общие положения)

IEC 61131-2, Programmable controllers — Part 2: Equipment requirements and tests (Контроллеры программируемые. Часть 2. Требования к оборудованию и испытания)

IEC/TR 62390, Common automation device — Profile guideline (Общие средства автоматики. Руководящие положения к описанию)

ISO/IEC 646:1991, Information technology — ISO 7-bit coded character set for information interchange (Информационные технологии. 7-битный набор кодированных символов ИСО для обмена информацией)

ISO/IEC 646:1991, Information technology — ISO 7-bit coded character set for information interchange (Информационные технологии. Структура кода символов и методы расширения)

ISO/IEC 10646, Information technology — Universal Multiple-Octet Coded Character Set (UCS) [Информационные технологии. Универсальный многооктетный набор кодированных символов (UCS)]

ISO/IEC 10731, Information technology — Open Systems Interconnection — Basic Reference Model — Conventions for the definition of OSI services (Информационные технологии. Взаимодействие открытых систем. Базовая эталонная модель. Соглашения для определения служб OSI)

ISO/IEC 19505 (all parts), Information technology — Object Management Group Unified Modeling Language (OMG UML) [Информационные технологии. Унифицированный язык моделирования Рабочей группы по управлению объектами (OMG UML), (все части ISO/IEC 19505)]

ISO 1177, Information processing — Character structure for start/stop and synchronous character oriented transmission (Обработка информации. Структура символов для стартстопной и синхронной символьно-ориентированной передачи)

IEEE Std 754-2008, IEEE Standard for Floating-Point Arithmetic (Стандарт IEEE для арифметики с плавающей точкой)

Internet Engineering Task Force (IETF): RFC 5905 — Network Time Protocol Version 4: Protocol and Algorithms Specification; available at <gost.e-ecolog.ru ая группа инженеров Интернета (IETF). Запрос комментария RFC 5905. Протокол сетевой синхронизации, версия 4. Протокол и спецификация алгоритмов; доступный по <gost.e-ecolog.ru

3 Термины, определения, обозначения, сокращения и условные обозначения

3.1 Термины и определения

В настоящем стандарте применены термины по МЭК 61131-1 и МЭК 61131-2, а также следующие термины с соответствующими определениями:

3.1.1 **адрес** (address): Часть управления M-последовательностью в ссылочных данных для категорий данных канала связи.

3.1.2 **прикладной уровень** (application layer; AL): Дополнительная часть протокола <SDCI>, ответственная за передачу объектов Данных процесса и объектов Данных запроса.

3.1.3 **параметр блока** (block parameter): Согласованный доступ к параметрам через множественные индексы или субиндексы.

3.1.4 **контрольная сумма** (checksum): Дополнительная часть совокупных мер интерфейса <SDCI> по непротиворечивости данных на канальном уровне в дополнение к биту четности UART.

3.1.5 **CHKPDU** (CHKPDU): Данные защиты передаваемой информации в канале связи индексированного сервисного блока данных (ISDU), сгенерированные путем сложения по модулю двух октетов запроса или ответа.

3.1.6 **кодовая коммутация** (coded switching): Связь SDCI, основанная на стандартных уровнях двоичного сигнала МЭК 61131-2.

3.1.7 **COM1** (COM1): Режим связи SDCI со скоростью передачи данных 4,8 кбит/с.

3.1.8 **COM2** (COM2): Режим связи SDCI со скоростью передачи данных 38,4 кбит/с.

3.1.9 **COM3** (COM3): Режим связи SDCI со скоростью передачи данных 230,4 кбит/с.

3.1.10 **COMx** (COMx): Один из трех возможных режимов связи SDCI: COM1, COM2 или COM3.

3.1.11 **канал связи** (communication channel): Логическое соединение между Ведущим узлом и Устройством.

Примечание — Определены четыре канала связи: канал процесса, канал страницы, канал ISDU (для параметров) и канал диагностики.

3.1.12 **ошибка связи** (communication error): Непредвиденное нарушение в работе протокола передачи SDCI.

3.1.13 **время цикла** (cycle time): Время передачи M-последовательности между Ведущим узлом и его Устройствами, включая последующее время простоя.

3.1.14 **Устройство** (Device): Отдельный пассивный узел сети относительно Ведущего узла, такой как датчик или исполнительное устройство.

Примечание — Термин «Устройство», начинающийся с заглавной буквы, применятся для оборудования SDCI, в то время как в остальных случаях применяется термин «устройство», начинающийся со строчной буквы.

3.1.15 **непосредственные параметры** (Direct Parameters): Параметры с прямой (страничной) адресацией, ациклически передаваемые через канал связи страниц без подтверждения.

3.1.16 **динамический параметр** (dynamic parameter): Часть набора параметров Устройства, определенная встроенными интерфейсами пользователя, такими как обучаемые кнопки или панели управления в дополнение к статическим параметрам.

3.1.17 **Событие** (Event): Экземпляр изменения условий в Устройстве.

Примечание 1 — Термин «Событие», начинающийся с заглавной буквы, применятся для Событий SDCI, в то время как в остальных случаях применяется термин «событие», начинающийся со строчной буквы.

Примечание 2 — Событие указывается в виде флага События в циклической информации о состоянии Устройства, затем происходит ациклическая передача данных События (обычно диагностической информации) через канал связи диагностики.

3.1.18 **возврат в исходный режим** (allback): Переход порта из режима кодовой коммутации в режим коммутирующего сигнала.

3.1.19 **уровень контроля** (inspection level): Уровень проверки идентичности Устройства.

3.1.20 **расслоение** (interleave): Сегментированная циклическая передача Данных процесса более чем с двумя октетами через последовательные циклы.

3.1.21 **ISDU** (ISDU): Индексированный сервисный блок данных, используемый для ациклической передачи параметров с подтверждением, которые могут быть сегментированы в нескольких М-последовательностях.

3.1.22 у**старевшее Устройство или Ведущий блок** (legacy Device or Master): Устройство или Ведущий блок в соответствии с [8].

3.1.23 **М-последовательность** (M-sequence): Последовательность из двух сообщений, включающая сообщение Ведущего узла и соответствующее ему сообщение Устройства.

3.1.24 **управление М-последовательностью** (M-sequence control): Первый октет в сообщении Ведущего узла, указывающий операцию чтения/записи, тип канала связи и адрес: например, смещение или управление потоками.

3.1.25 **ошибка М-последовательности** (M-sequence error): Непредвиденное или неправильное содержание сообщения или отсутствие ответа.

3.1.26 **тип М-последовательности** (M-sequence type): Отдельный особенный формат М-последовательности или набор специфических форматов М-последовательности.

3.1.27 **Ведущий узел** (Master): Активный одноранговый узел сети, присоединенный через порты к одному или более Устройствам (вплоть до *n*) и обеспечивающий интерфейс к шлюзу для систем связи более высокого уровня или PLC.

Примечание — Термин «Ведущий узел», начинающийся с заглавной буквы, применяется для оборудования SDCI, в то время как в остальных случаях применяется термин «ведущий узел», начинающийся со строчной буквы.

3.1.28 **сообщение** (message): Последовательность <SDCI> фреймов UART, передаваемая либо от Ведущего узла к его Устройствам, либо в обратном направлении, в соответствии с правилами протокола SDCI.

3.1.29 **Данные запроса** (On-request Data; OD): Ациклически передаваемые данные по запросу от приложения Ведущего узла, состоящие из параметров или данных События.

3.1.30 **физический уровень** (physical layer; PL): Первый уровень эталонной модели взаимодействия открытых систем OSI, который предоставляет механические, электрические, функциональные или процедурные средства для активации, поддержки и деактивации физических соединений для передачи битов между объектами канального уровня.

Примечание — Физический уровень также предоставляет средства для процедур пробуждения и возврата в исходное состояние.

[ИСТОЧНИК: ИСО/МЭК 7498-1:1994, 7.7.2, изменено — текст удален из статьи, добавлено примечание]

3.1.31 **порт** (port): Интерфейс среды передачи данных Ведущего узла к одному Устройству.

3.1.32 **режим работы порта** (port operating mode): Состояние порта Ведущего узла, которое может быть: INACTIVE, DO, DI, FIXEDMODE или SCANMODE.

3.1.33 **Данные процесса** (Process Data): Входные или выходные значения от или к дискретному или непрерывному процессу автоматизации, циклически передаваемые с высоким приоритетом и в сконфигурированной последовательности после запуска Ведущего узла.

3.1.34 **цикл Данных процесса** (Process Data cycle): Законченная передача всех Данных процесса от или к отдельному Устройству, которая может включать несколько циклов в случае сегментации (расслоения).

3.1.35 **отдельный параметр** (single parameter): Доступ к независимому параметру через один отдельный Индекс или Субиндекс.

3.1.36 **стандартный ввод/вывод;** SIO (SIO). Режим работы порта в соответствии с цифровым вводом и выводом, определенным в МЭК 61131-2, который установлен после включения питания, возврата в исходное состояние или неудачных попыток передачи данных.

3.1.37 **статический параметр** (static parameter): Часть набора параметров Устройства, сохраняемая Ведущим узлом на случай замены без использования технических средств.

3.1.38 **коммутирующий сигнал** (switching signal): Двоичный сигнал от или к Устройству при нахождении в режиме SIO (в отличие от «кодовой коммутации» передачи данных SDCI).

3.1.39 **модуль управления системой** (system management; SM): Средства <SDCI> для управления и координации внутренних слоев связи и исключений внутри Ведущего узла и его портов и внутри каждого Устройства.

3.1.40 **фрейм UART** (UART frame): Последовательность битов <SDCI>, начинающаяся со стартового бита, с последующими восемью битами, несущими октет данных следующим битом четности, и оканчивающаяся одним стоповым битом.

3.1.41 **пробуждение** (wake-up): Процедура, заставляющая Устройство изменить его режим с SIO на SDCI.

3.1.42 **запрос пробуждения** (wake-up request; WURQ): Услуга физического слоя, используемая Ведущим узлом для инициирования пробуждения Устройства и помещения его в состояние готовности приема.

3.2 Обозначения и сокращения

В настоящем стандарте применены следующие обозначения и сокращения:

Δf_{DTRM} — допустимое отклонение в скорости передачи данных (измеренное в %);

ΔVS — пульсация питающего напряжения (измеренная в В);

AL — прикладной уровень;

BEP — вероятность битовой ошибки;

C/Q — соединение для передачи данных (C) или коммутирующего сигнала SIO (Q);

CL_{eff} — эффективная общая емкость кабеля (измеренная в нФ);

CQ — входная емкость соединения C/Q (измеренная в нФ);

DI — цифровой вход;

DL — канальный уровень;

DO — цифровой выход;

f_{DTR} — скорость передачи данных (измеренная в бит/с);

H/L — высокий/низкий сигнал на выходе приемника;

I/O — ввод/вывод;

ILL — ток входной нагрузки на входе C/Q в *V0* (измеренный в А);

IODD — описание устройства ввода/вывода (см. 10.8);

Iq — ток драйвера в насыщенном рабочем состоянии (измеренный в А);

IQH — ток драйвера верхнего плеча в насыщенном рабочем состоянии ON (измеренный в А);

IQL — ток драйвера нижнего плеча в насыщенном рабочем состоянии ON (измеренный в А);

IQPK — максимальный ток драйвера в ненасыщенном рабочем состоянии ON (измеренный в А);

IQPKH — максимальный ток драйвера верхнего плеча в ненасыщенном рабочем состоянии ON (измеренный в А);

IQPKL — максимальный ток драйвера нижнего плеча в ненасыщенном рабочем состоянии ON (измеренный в А);

IQQ — ток в рабочей точке на входе C/Q в *V0* с неактивными выходными драйверам (измеренный в А);

IQ_{WU} — амплитуда тока запроса пробуждения Ведущего узла (измеренный в А);

IS — питающий ток в *V+* (измеренный в А);

ISIR — допустимый импульс тока в *V+* (измеренный в А);

LED — светоизлучающий диод;

L– — источник питания (–);

L+ — источник питания (+);

N24 — дополнительный источник питания 24 В (–);

n_{WU} — счетчик повторных попыток пробуждения;

On/Off — сигнал переключения ON/OFF драйвера;

OD — данные запроса;

OVD — обнаружен избыточный сигнал;

P24 — дополнительный источник питания 24 В (+);

PD — данные процесса;

PDCT — инструмент конфигурирования порта и устройства;

PL — физический уровень;

PLC — программируемый логический контроллер;

PS — напряжение источника питания (измеренное в В);

r — время достижения устойчивого уровня относительно начала стартового бита (измеренный в T_{BIT});

RL_{eff} — сопротивление шлейфа кабеля (измеренное в Ом);

s — время выхода на устойчивый уровень относительно начала стартового бита (измеренный в T_{BIT});

SDCI — одноточечный интерфейс цифровой связи;

SIO — стандартный ввод/вывод (режим цифровой коммутации) [IEC 61131-2] управление системой;

SM — модуль управления системой;

t_1 — задержка передачи фрейма UART на Ведущем узле (измеренная в T_{BIT});

t_1 — задержка передачи фрейма UART на Устройстве (измеренная в T_{BIT});

t_A — задержка ответа на Устройстве (измеренная в T_{BIT});

T_{BIT} — время передачи бита (измеренное в с);

$t_C Y_C$ — время цикла на уровне М-последовательности (измеренное в с);

t_{DF} — время спада (измеренное в с);

T_{DMT} — время задержки при установлении порта связи Ведущего узла (измеренное в T_{BIT});

T_{DR} — время нарастания сигнала (измеренное в с);

T_{DSIO} — время задержки на устройстве для перехода в режим SIO после запроса пробуждения (измеренное в с);

T_{DWU} — задержка повторной попытки пробуждения (измеренная в с);

$t_{M\text{-}sequence}$ — продолжительность М-последовательности (измеренная в T_{BIT});

$t_{id}I_e$ — время простоя между двумя М-последовательности (измеренное в с);

t_H — время обнаружения для верхнего плеча (измеренное в с);

t_L — время обнаружения для нижнего плеча (измеренное в с);

t_{ND} — время подавления шума (измеренное в с);

T_{RDL} — время готовности к пробуждению после включения питания (измеренное в с);

T_{REN} — время подготовки к получению сигнала (измеренное в с);

T_{SD} — время обнаружения устройства (измеренное в с);

T_{WU} — длительность импульса запроса пробуждения (измеренная в с);

UART — универсальный асинхронный приемник-передатчик;

UML — универсальный язык моделирования UML [ИСО/МЭК 19505];

V+ — напряжение в L+;

V0 — напряжение в L–;

$VD+_L$ — перепад напряжения в линии между соединениями L+ на Ведущем узле и Устройстве (измеренный в В);

$VD0_L$ — перепад напряжения в линии между соединениями L– на Ведущем узле и Устройстве (измеренный в В);

VDQ_L — перепад напряжения в линии между соединениями C/Q на Ведущем узле и Устройстве (измеренный в В);

VHYS — гистерезис порогового напряжения приемника (измеренный в В);

VI — входное напряжение в соединении линии C/Q относительно *V0* (измеренное в В);

VIH — диапазон входного напряжения в соединении C/Q для верхнего плеча (измеренный в В);

VIL — диапазон входного напряжения в соединении C/Q для нижнего плеча (измеренный в В);

VRQ — остаточное напряжение на драйвере в насыщенном рабочем состоянии ON (измеренное в В);

VRQH — остаточное напряжение на драйвере верхнего плеча в насыщенном рабочем состоянии ON (измеренное в В);

VRQL — остаточное напряжение на драйвере нижнего плеча в насыщенном рабочем состоянии ON (измеренное в В);

VTH — пороговое напряжение приемника относительно *V0* (измеренное в В);

VTHH — пороговое напряжение приемника для безопасного обнаружения высокого сигнала (измеренное в В);

VTHL — пороговое напряжение приемника для безопасного обнаружения низкого сигнала (измеренное в В);

WURQ — запрос пробуждения.

3.3 Условные обозначения

3.3.1 Общие положения

Модель обслуживания, сервисные примитивы и схемы, приведенные в настоящем стандарте, являются абстрактными описаниями. Реализация сервисов может охватывать отдельные проблемы и различаться.

3.3.2 Параметры обслуживания

Сервисные примитивы применяются для представления взаимодействий провайдера услуг и их потребителя (ИСО/МЭК 10731). Сервисные примитивы передают параметры, которые указывают на информацию, имеющуюся при взаимодействии провайдера и потребителя. В конкретном интерфейсе нет необходимости явно формулировать все параметры.

Спецификация услуги в настоящем стандарте представлена в табличном формате для описания параметров компонентов в сервисных примитивах. Параметры, применяемые в каждой группе сервисных примитивов, приведены в таблицах. Каждая таблица содержит не более пяти колонок:

1) имя параметра;

2) примитив запроса (.req);

3) примитив индикации (.ind);

4) примитив ответа (.rsp);

5) примитив подтверждения (.cnf).

В каждой строке таблицы располагается один параметр (или его компонент). Под соответствующими колонками сервисных примитивов используется код для указания типа использования параметра в примитиве, указанном в колонке:

M — обязательный параметр для примитива;

U — параметр является возможностью пользователя: он может представляться или опускаться в зависимости от конкретного использования услуги пользователем. Если параметр не представлен, предполагается значение параметра по умолчанию;

C — параметр зависит от других параметров или от среды пользователя услуги;

— — параметр никогда не применяется;

S — параметр является выделенным объектом.

Некоторые строки более детально классифицируются элементами в скобках. Такими элементами могут являться:

a) специфическое ограничение параметра «(=)» указывает, что параметр семантически эквивалентен значению параметра, указанному в сервисном примитиве;

b) указание, что к строке таблицы применяется некоторое примечание «*n*», означает, что данное примечание «*n*» содержит дополнительную информацию, относящуюся к параметру и его использованию.

3.3.3 Сервисные процедуры

Процедуры определяются в терминах:

- взаимодействий между сущностями приложения через обмен блоками данных протокола;
- взаимодействий между провайдером и потребителем услуг слоя связи в системе через вызов сервисных примитивов.

Данные процедуры применяются к экземплярам связи между системами, поддерживающими ограниченные во времени услуги между слоями связи.

3.3.4 Атрибуты услуги

Природа различных услуг (Ведущего узла и Устройства) характеризуется атрибутами. Все услуги определяются с точки зрения задействованного слоя относительно слоя более высокого уровня.

I — Инициатор услуги (по отношению к слою более высокого уровня).

R — Получатель (ответчик) услуги (из слоя более высокого уровня).

3.3.5 Рисунки

Для рисунков, на которых показаны структура и услуги слоев протокола, применяются следующие условные обозначения:

- стрелка, имеющая только имя услуги, представляет как запрос, так и соответствующее подтверждение (запрос следует по направлению стрелки);

- запрос без подтверждения, а также все указания и ответы, помеченные как таковые (т. е. service.req, service.ind, service.rsp).

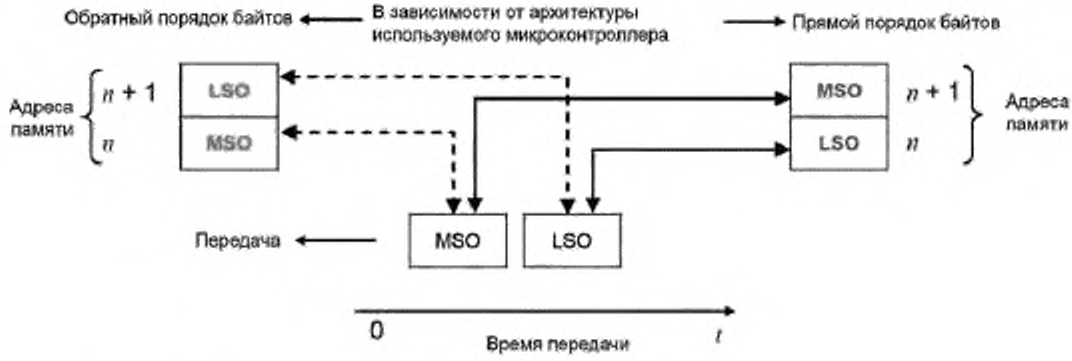
Пример услуги с подтверждением показан на рисунке 1.



Рисунок 1 — Пример подтвержденного сообщения

3.3.6 Порядок передачи октета

На рисунке 2 показано, как типы данных, основанные на слове (WORD), передаются из памяти в среду передачи данных и наоборот (т. е. старший октет передается первым, см. 7.3.3.2 и 7.3.6.1).



Обозначения:
MSO — самый старший октет;
LSO — самый младший октет.

Рисунок 2 — Запоминающее устройство и порядок передачи для типов данных, основанных на слове (WORD)

3.3.7 Поведенческое описание

Для поведенческого описания применяется нотация языка UML 2 (ИСО/МЭК 19505) (например, понятия: состояние, последовательность, действие, временные диаграммы, сторожевые условия). Расположение соответствующих таблиц перехода состояний соответствует IEC/TR 62390.

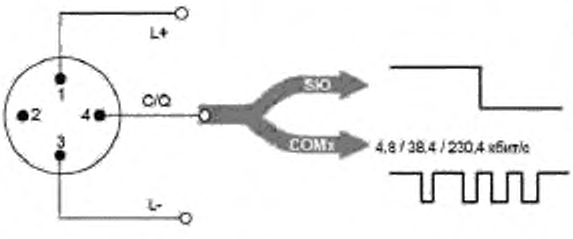
Вследствие ограничений инструментов проектирования применяются следующие исключения. Для диаграмм состояний параметры услуги (печатными буквами) добавляются к имени услуги через символ подчеркивания, как, например, в DL_SetMode_INACTIVE. Для диаграмм последовательности сервисный примитив добавляется через символ подчеркивания вместо точки, параметры услуги добавляются в скобках, как, например, в DL_Event_ind (OPERATE). Временные ограничения помечаются «tm (время в мс)».

Вызовы услуг, получаемых асинхронно, не моделируются в деталях в диаграммах состояний.

4 Обзор одноточечного интерфейса цифровой связи SDCI (IO-Link™)

4.1 Назначение технологии

На рисунке 3 показана основная концепция технологии SDCI.



Контакт разъема	Сигнал	Определение	Стандарт
1	L+	24 В	IEC 61131-2
2	I/Q	Не подключен, DI или DO	IEC 61131-2
3	L-	0 В	IEC 61131-2
4	Q	«Коммутирующий сигнал» DI, DO (SIO)	IEC 61131-2
	C	«Кодовая коммутация» (COM1, COM2, COM3)	IEC 61131-9

Рисунок 3 — Совместимость SDCI с МЭК 61131-2

Технология одноточечного интерфейса цифровой связи для небольших датчиков и исполнительных устройств SDCI (более известных под названием IO-Link™) определяет направление перехода от существующих интерфейсов цифрового ввода и цифрового вывода для коммутационных 24-вольтовых Устройств, как установлено в МЭК 61131-2, к каналам связи между двумя пунктами. Так, например, цифровые модули I/O в существующих периферийных устройствах полевой шины могут быть заменены модулями Ведущего узла SDCI, обеспечивая как классические интерфейсы цифрового I/O, так и SDCI. Технология аналоговой передачи может быть заменена SDCI, сочетая присущие SDCI устойчивость, параметризацию и диагностические возможности с сохранением возможностей цифро-аналогового и аналого-цифрового преобразования.

4.2 Позиционирование в иерархии средств автоматизации

На рисунке 4 показана область применения технологии SDCI в иерархии средств автоматизации.

Технология SDCI определяет обобщенный интерфейс для присоединения датчиков и исполнительных устройств к устройству Ведущего узла, который может комбинироваться с возможностями межсетевого интерфейса для создания удаленного узла I/O полевой шины.

Отправной точкой в проектировании SDCI является традиционный интерфейс 24 В DI и DO, определенный в МЭК 61131-2 и показанный в таблице 6. Таким образом, технология SDCI предлагает подключаемость классических датчиков 24 В («коммутирующих сигналов») в качестве операционного режима по умолчанию. Дополнительная подключаемость обеспечивается для исполнительных устройств, когда порт сконфигурирован в режиме одноточечного интерфейса цифровой связи.

В настоящее время многие датчики и исполнительные устройства уже оборудованы микроконтроллерами, предлагающими интерфейс UART, который может быть расширен добавлением нескольких аппаратных компонентов и программным обеспечением интерфейса для поддержки связи SDCI. Данный второй операционный режим применяет «кодовую коммутацию» линии сигналов I/O 24 В. После активации режим SDCI поддерживает параметризацию, циклический обмен данными, диагностическую отчетность, информацию по идентификации и технической поддержке и память внешних параметров для резервирования Устройств и быстрой перезагрузки замененных устройств. Датчики и исполнительные устройства с возможностями SDCI именуются в настоящем стандарте «Устройствами». Для улучшения запуска данные Устройства, как правило, предоставляют энергонезависимую память для параметров.

Примечание — Конфигурирование и параметризация Устройств поддерживается с помощью описания устройства на базе XML (см. [6]), которое не является частью настоящего стандарта.

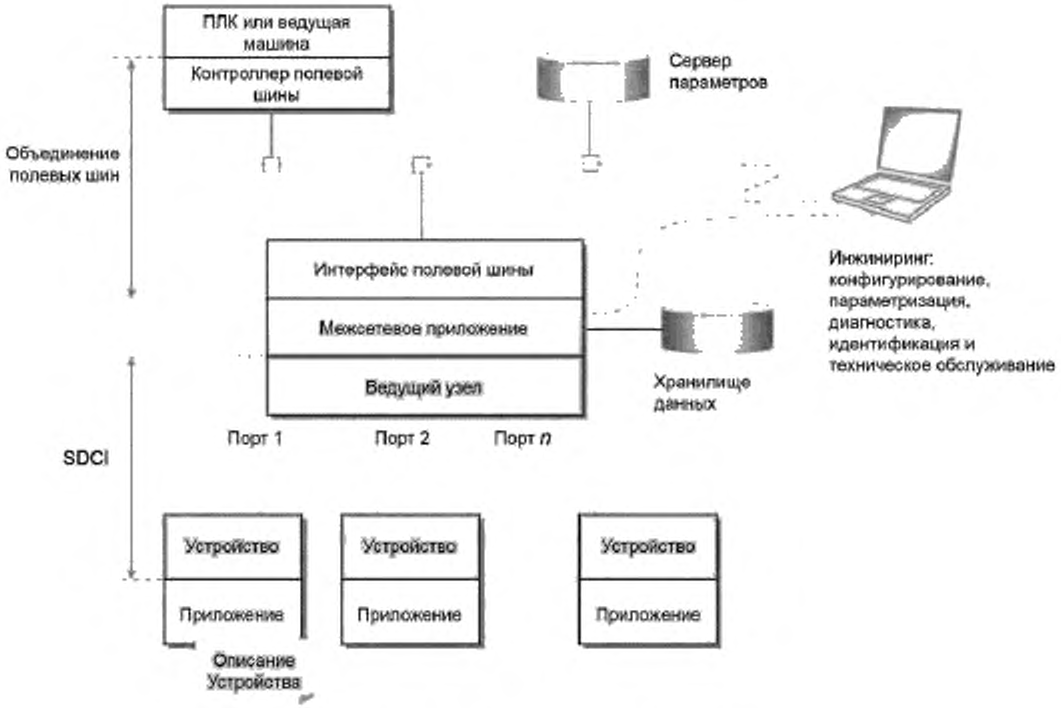


Рисунок 4 — Сфера применения технологии SDCI в иерархии средств автоматизации

4.3 Проводка, соединители и электропитание

Соединение по умолчанию (порт класса А) имеет четыре контакта (см. рисунок 3). Порт по умолчанию для порта класса А соответствует МЭК 60947-5-2 и применяет три провода для 24 В, 0 В и сигнальной линии. Четвертый провод может применяться для дополнительной сигнальной линии, соответствующей МЭК 61131-2.

Соединения с пятью контактами (порт класса В) устанавливаются для Устройств, требующих дополнительного питания от независимого источника питания 24 В.

Примечание — Устройство с портом класса А, применяющее четвертый провод, несовместимо с Ведущим узлом с портом класса В.

Максимальная длина кабелей — 20 м, экранирование не требуется.

4.4 Возможности связи SDCI

Обобщенная модель Устройства показана на рисунке 5 и объясняется в следующих разделах.

Устройство может получать PD (вывод) для управления дискретным или непрерывным процессом автоматизации или посылать PD (ввод), представляющие текущее состояние или значения измерений. Как правило, Устройство предоставляет параметры, позволяющие пользователю конфигурировать функции Устройства для удовлетворения конкретных потребностей. Для поддержки этого определено большое пространство параметров с доступом через Индекс (0 … 65535; с предварительно определенной организацией) и Субиндекс (0 … 255).

Первые два значения индекса 0 и 1 зарезервированы для страниц 1 и 2 Непосредственных параметров с максимальной длиной 16 октетов каждая. Страница 1 параметров в основном предназначена для команд Ведущего узла, таких как запуск Устройства и его возврат в исходное состояние, извлечение специфической операционной и идентификационной информации Устройства. Страница 2 параметров предусматривает максимум 16 октетов специфических параметров Устройства.

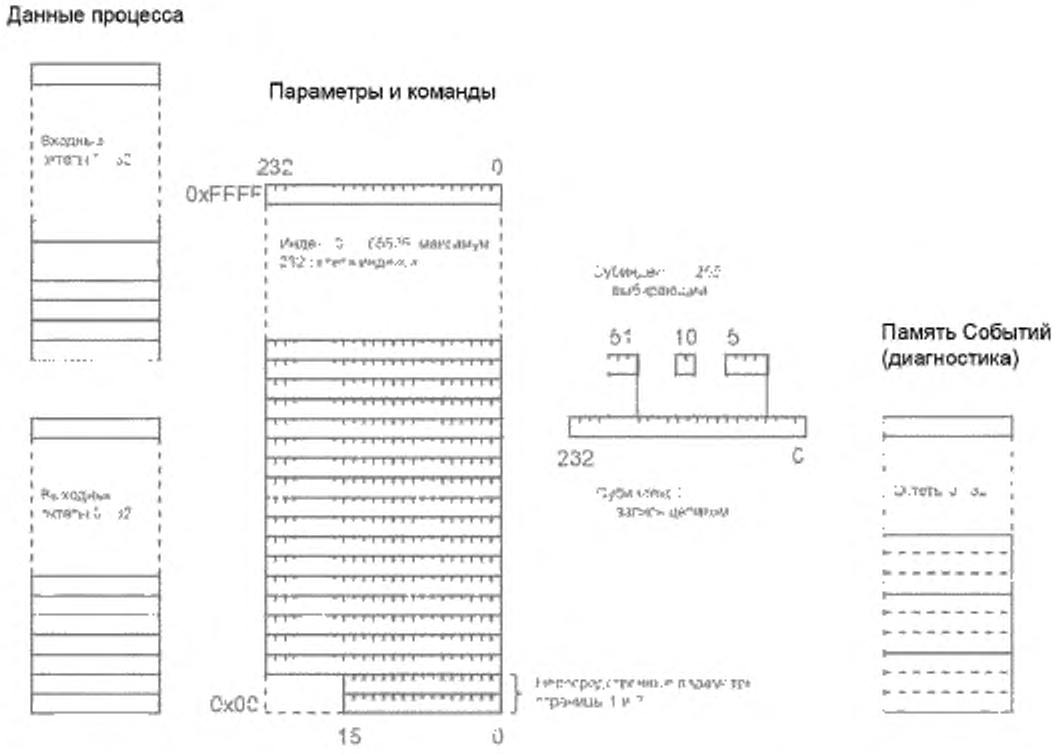


Рисунок 5 — Обобщенная модель Устройства для SDCI (с точки зрения Ведущего узла)

Другие индексы (2 … 65 535) обеспечивают доступ к одной записи с максимальным размером 232 октета. Субиндекс 0 определяет передачу полной записи, адресуемой Индексом, другие субиндексы определяют передачи отобранных элементов из записи.

В пределах записи отдельные элементы данных могут начинаться с любого битового смещения, и их длина может быть в пределах от 1 бита до 232 октетов, но общее число элементов данных в записи не может превышать 255. Организация элементов данных в записи определяется в IODD.

Все изменения состояния Устройства, требующие выдачи отчета или вмешательства, сохраняются в памяти Событий до передачи. Затем в циклическом обмене данными устанавливается Флаг События для указания наличия События.

Связь между Ведущим узлом и Устройством осуществляется из точки в точку и основывается на принципе, что сначала Ведущий узел посылает сообщение, а затем Устройство посылает ответное сообщение (см. рисунок 36). Оба сообщения вместе называются М-последовательностью. Определяется несколько типов М-последовательности для поддержания требований пользователя к передаче данных (см. рисунок 37).

Данные различных категорий передаются через различные каналы связи на DL, как показано на рисунке 6.

- Операционные данные, такие как вводы и выводы Устройства, передаются через канал связи процесса, применяя циклическую передачу. Операционные данные также могут быть связаны с описателями, такими как допустимый, недопустимый.

- Параметры конфигурирования и технического обслуживания передаются ациклически. Для прямого доступа к страницам 1 и 2 параметров предусмотрен канал связи страниц, а канал ISDU применяется для доступа к дополнительным параметрам и командам.

- События Устройства передаются, применяя асинхронные передачи по каналу диагностики. События устройства разделяются на три уровня серьезности: ошибки, предупреждения и уведомления.



Рисунок 6 — Отношения между характеристиками данных и типами передачи

Первый октет сообщения Ведущего узла определяет направление передачи данных (чтение или запись) и тип канала связи.

На рисунке 7 показано, что каждый порт Ведущего узла имеет свой собственный DL, обеспечивающий сопряжение с общим AL Ведущего блока. На AL услуги DL транслируются в действия с объектами PD (ввод/вывод), объектами OD (чтение/запись) и Событиями. Приложения Ведущего узла включают в себя: Управление конфигурацией (CM), механизм Хранилища данных (DS), Блок диагностики (DU), Обмен Данными запроса (ODE) и Обмен Данными процесса (PDE).

SM проверяет идентификацию подключенных Устройств и настраивает порты и Устройства, чтобы они соответствовали выбранной конфигурации и свойствам подключенных Устройств. Модуль контролирует состояние машин на AL и DL, например, при запуске.

4.5 Роль Ведущего узла

Ведущее устройство размещает от 1 до *n* портов и их соответствующих DL. Во время запуска оно изменяет состояние порта на режим, выбранный пользователем. Возможные состояния порта: INACTIVE (неактивное), DI, DO, FIXEDMODE (фиксированный режим) или SCANMODE (режим сканирования). Если затребована связь, Ведущий узел применяет специальный импульс тока пробуждения для инициации связи с Устройством. Затем Ведущий узел автоматически настраивает скорость передачи данных для портов COM1, COM2 или COM3 (см. таблицу 8) и проверяет «паспорт» каждого подключенного устройства, т. е. идентификатор изготовителя VendorID, идентификатор Устройства DeviceID и характеристики связи.

Если имеется несоответствие между параметрами Устройства и сохраненным набором параметров в Ведущем узле, параметры в Устройстве перезаписываются (см. 11.3) или параметры, сохраненные в Ведущем узле, корректируются в зависимости от конфигурации.

Допускается также запустить устройство в режиме DI, переключить на связь SDCI для конфигурирования и параметризации и затем применить команду «fallback» возврата в исходный режим (см. 11.8.5) для переключения назад в режим DI для нормального функционирования.

Координация портов также является задачей Ведущего блока, который пользователь может конфигурировать путем выбора режима цикла порта. В режиме «FreeRunning» («Автономный») каждый порт определяет собственный цикл, основываясь на свойствах подключенного Устройства. В режиме «MessageSync» («Синхронизация сообщений») сообщения, посланные на подключенные порты, активизируются одновременно или в определенном шахматном порядке. В режиме «FixedValue» («Фиксированное значение») каждый порт применяет определенное пользователем фиксированное время цикла (см. 11.2.2.2).

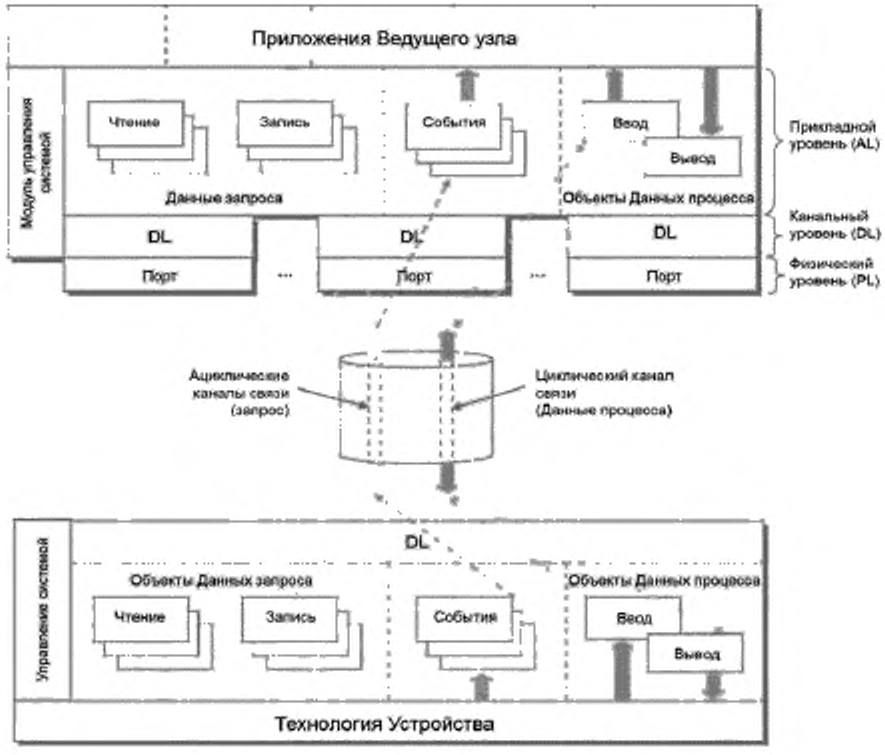


Рисунок 7 — Передача объекта данных на AL

Ведущий узел ответственен за сборку и разборку всех данных от Устройств к Устройствам (см. раздел 11).

Ведущий узел предоставляет область DS размером не менее 2048 октетов на Устройство для резервного копирования данных Устройства (см. 11.3). Ведущий узел может комбинировать эти данные Устройства с другими подходящими данными для собственных целей, делать эти данные доступными для приложения более высокого уровня с целью создания резервной копии или управления набором параметров (см. 11.8.3).

4.6 Конфигурирование SDCI

Инженерно-техническое обеспечение Ведущего узла, как правило, производится PDCT. PDCT регулирует как свойства порта, так и свойства Устройства (см. параметры, показанные на рисунке 5). PDCT сочетает как интерпретатор IODD, так и конфигуратор (см. 11.7). IODD обеспечивает необходимые свойства для организации связи и необходимые параметры с их границами для достижения требуемой функции датчика или исполнительного устройства. PDCT также поддерживает компиляцию PD для передачи на полевую шину и в обратном направлении.

4.7 Отображение на полевые шины

Интеграция Ведущего узла в системе полевой шины, т. е. определение функций шлюза для обмена данными с объектами более высокого уровня в шине, находится вне области применения настоящего стандарта.

Пример — Данные функции включают отображение обмена PD, реализацию программно-управляемой параметризации или удаленного сервера параметров и распространение диагностической информации.

Интеграция PDCT в инженерно-технические инструментальные средства конкретной полевой шины находится вне области применения настоящего стандарта.

4.8 Структура стандарта

На рисунке 8 показана логическая структура Ведущего узла и Устройства. PL SDCI устанавливается в разделе 5, подробное описание режима SIO устанавливается в разделе 6. Услуги DL, протокол, пробуждение, M-последовательности и обработчики DL устанавливаются в разделе 7. Услуги и протокол AL устанавливаются в разделе 8, функциональные обязанности SM устанавливаются в разделе 9.

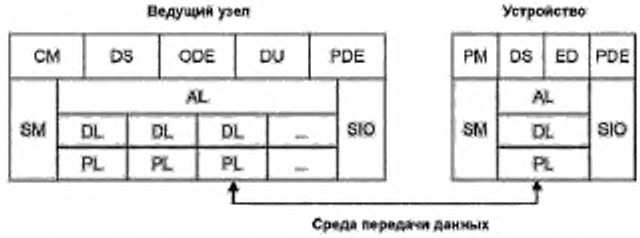


Рисунок 8 — Логическая структура Ведущего узла и Устройства

В разделе 10 устанавливаются приложения Устройства и его характерные особенности. Приложения включают в себя: PDE, PM, DS и Диспетчер событий (ED). Специфические технологические приложения не являются частью настоящего стандарта. Эти приложения могут описываться в профилях для конкретных семейств Устройств.

В разделе 11 устанавливаются приложения Ведущего узла и его характерные особенности. Приложения включают в себя: PDE, ODE, Управление конфигурацией CM, DS и DU.

В настоящий стандарт включено несколько обязательных и справочных приложений. В приложении A устанавливаются доступные типы M-последовательностей. В приложении B устанавливаются параметры страницы Непосредственных параметров и фиксированных параметров Устройства. В приложении C приведены типы ошибок при ациклических передачах, а в приложении D — коды Событий (диагностическая информация Устройств). В приложении E приведены доступные базовые и составные типы данных. В приложении F приведена структура объектов DS. В приложении G описаны вопросы соответствия и электромагнитной совместимости, а в приложении H приведены кривые вероятностей остаточных ошибок, подтверждающие уровень целостности данных SDCI. В приложении I дан пример последовательности ациклических передач данных. В приложении J объясняются два рекомендуемых метода для определения изменений параметров в контексте DS.

5 Физический уровень (PL)

5.1 Общие положения

5.1.1 Базовая комплектация

Система 3-проводных присоединений SDCI основана на спецификациях, приведенных в МЭК 60947-5-2. Три линии применяются следующим образом: (L+) — для источника питания 24 В; (L–) — для линии заземления и (C/Q) — для коммутирующего сигнала (Q) или связи SDCI (C), как показано на рисунке 9.

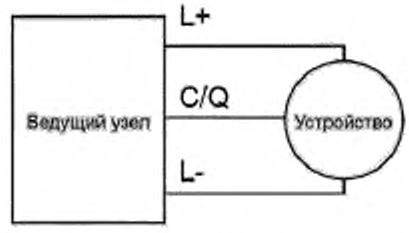


Рисунок 9 — Система 3-проводного соединения

Примечание 1 — Двоичные датчики, соответствующие МЭК 60947-5-2, совместимы с системой 3-проводного присоединения SDCI (в том числе с точки зрения энергопотребления).

Поддержка системы 3-проводного присоединения SDCI обязательна для Ведущего узла. Порты с такой характеристикой называются портами класса А.

Порт класса А применяет четырехконтактный соединитель. Четвертый провод может применяться для дополнительной сигнальной линии в соответствии с МЭК 61131-2. Поддержка четвертого провода необязательна как в Ведущих узлах, так и в Устройствах.

Присоединения с пятью проводами (порт класса В) устанавливаются для Устройств, требующих дополнительного питания от независимого источника питания 24 В.

Примечание 2 — Устройство с портом класса А, применяющее четвертый провод, несовместимо с Ведущим узлом с портом класса В.

5.1.2 Топология

Топология системы SDCI применяет одноточечные связи между Ведущим узлом и его Устройствами, как показано на рисунке 10. Ведущий узел может иметь много портов для подключения Устройств. К каждому порту будет подключаться только одно Устройство.

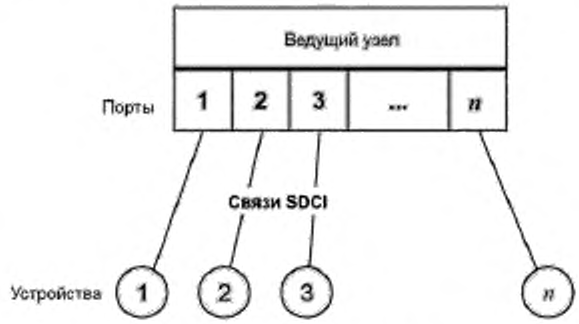


Рисунок 10 — Топология SDCI

5.2 Услуги физического уровня

5.2.1 Обзор

На рисунке 11 даются обзор PL Ведущего узла и его сервисные примитивы.



Рисунок 11 — Физический уровень (Ведущий узел)

PL определяет операции линии C/Q на рисунке 3 и связанный драйвер линии (передатчик) и приемник конкретного порта. Ведущий узел управляет этой линией в трех основных режимах (см. рисунок 11): неактивном, «Коммутирующий сигнал» (DI/DO) или «Кодовая коммутация» (COMx). Услуга PL_SetMode.req отвечает за переключение в один из данных трех режимов.

Если порт находится в неактивном режиме, линия C/Q будет высокоимпедансной (слабонагруженной). В режиме SIO порт может применяться как стандартный входной или выходной интерфейс

в соответствии с определениями МЭК 61131-2 или таблицей 6 соответственно. Уровни связи SDCI обходятся, как показано на рисунке 11; сигналы прямо обрабатываются в приложении Ведущего узла. В режиме SDCI услуга PL_WakeUp.req создает специальный сигнал-шаблон (импульс тока), который может обнаруживаться Устройством с SDCI, подключенным к этому порту (см. 5.3.3.3).

На рисунке 12 даются обзор PL Устройства и его сервисные примитивы.

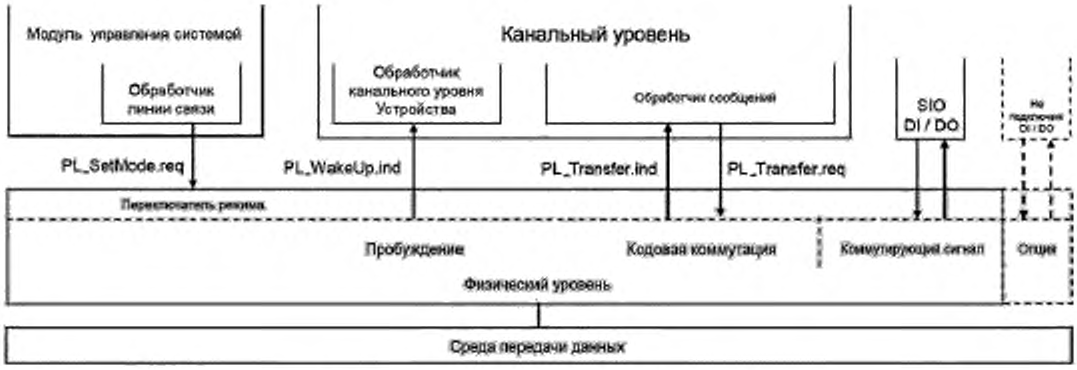


Рисунок 12 — Физический уровень (Устройство)

PL Устройства в соответствии с рисунком 12 следует тем же принципам, за исключением того, что здесь нет неактивного состояния. По умолчанию при включении питания или восстановлении соединения кабеля Устройство будет работать в режиме SIO, как цифровой ввод. Устройство всегда будет способно определить импульс тока пробуждения (запрос пробуждения). Услуга PL_WakeUp.ind сообщает об успешном обнаружении запроса пробуждения (обычно прерывания микроконтроллера), который требуется Устройству для переключения в режим SDCI.

Специальная команда Ведущего узла (fallback), посланная через SDCI, заставляет Устройство переключиться обратно в режим SIO.

После этого определяются услуги, которые предоставляются PL SM и DL (полный обзор данных услуг приводится на рисунках 83 и 94). В таблице 1 приведены назначения Ведущего узла и Устройства как инициатора и приемника для отдельных услуг PL.

Таблица 1 — Назначение услуг Ведущего узла и Устройства

Имя услуги	Ведущий узел	Устройство
PL_SetMode	R	R
PL_WakeUp	R	I
PL_Transfer	I/R	R/I
Обозначения (см. 3.3.4): I — Инициатор услуги; R — Приемник (Ответчик) услуги.		

5.2.2 Услуги Физического уровня

5.2.2.1 Услуга PL_SetMode

Услуга PL_SetMode применяется для установки электрических характеристик и конфигураций PL. Параметры сервисных примитивов приведены в таблице 2.

Таблица 2 — Услуга PL_SetMode

Имя параметра	.req
Argument TargentMode	M M

Argument (Аргумент)
Специфические для услуги параметры запроса услуги передаются в параметре Аргумент.

TargetMode (ЦелевойРежим)
Данный параметр указывает затребованный операционный режим.

Допустимые значения:
- INACTIVE (линия C/Q имеет высокий импеданс);
- DI (линия C/Q находится в режиме цифрового ввода);
- D (линия C/Q находится в режиме цифрового вывода);
- COM1 (линия C/Q находится в режиме COM1);
- COM2 (линия C/Q находится в режиме COM2);
- COM3 (линия C/Q находится в режиме COM3).

5.2.2.2 Услуга PL_WakeUp

Услуга PL_WakeUp инициирует или указывает специальную последовательность, которая подготавливает PL к посылке и приему запросов связи (см. 5.3.3.3). Данная неподтверждаемая услуга не имеет параметров. Ее успех может быть подтвержден только Ведущим узлом через попытку установления связи с Устройством. Сервисные примитивы приведены в таблице 3.

Таблица 3 — Услуга PL_WakeUp

Имя параметра	.req	ind
<отсутствуют>		

5.2.2.3 Услуга PL_Transfer

Услуга PL_Transfer применяется для обмена данными SDCI между DL и PL. Параметры сервисных примитивов приведены в таблице 4.

Таблица 4 — Услуга PL_Transfer

Имя параметра	.req	ind.
Argument Data	M	M
Result (+)		S
Result (–) Status		S M

Argument (Аргумент)
Специфические для услуги параметры запроса услуги передаются в параметр Аргумент.

Data (Данные)
Данный параметр содержит значения данных, которые переданы через интерфейс SDCI.

Допустимые значения 0 … 255.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что запрос услуги был выполнен успешно.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что запрос услуги завершился неуспешно.

Status (Состояние)
Данный параметр содержит дополнительную информацию о состоянии передачи.

Допустимые значения:
- PARITY_ERROR (UART обнаружил ошибку четности);
- FRAMING_ERROR (недопустимый стоповый бит UART);
- OVERRUN (конфликт октетов в UART).

5.3 Передатчик/приемник

5.3.1 Метод описания

Физический уровень определяется посредством электрических и временных требований. Электрические требования определяют уровни сигнала и тока отдельно для Ведущего узла и Устройств в виде эталонных схем. Временные требования определяют процесс передачи сигнала (в частности, в приемнике) и специальную функцию обнаружения сигнала.

5.3.2 Электрические требования

5.3.2.1 Общие положения

Драйвер линии определяется эталонной схемой, соответствующей рисунку 13. На стороне Ведущего узла передатчик включает комбинацию из двух драйверов линии и одного токового стока. На стороне Устройства в простейшем случае передатчик принимает форму драйвера переключения питания. В качестве дополнительного средства может присутствовать дополнительный некоммутирующий драйвер (это также предоставляет возможность работы с двухтактным выходом).

В рабочем состоянии ON описательными переменными являются остаточное напряжение *VRQ*, номинальный ток драйвера *IQ* и пиковый ток *IQPK*. Источник управляется сигналом включения/выключения On/Off. Наличие тока перегрузки указывается на выходе «Перегрузка». Данная возможность может применяться для обнаружения импульса тока (пробуждения).

Приемник определяется эталонной схемой в соответствии с рисунком 14. Приемник выполняет функции компаратора и характеризуется порогом переключения *VTH* и гистерезисом *VHYS* между порогами переключения. Выход показывает логический уровень (Высокий или Низкий) на входе приемника.

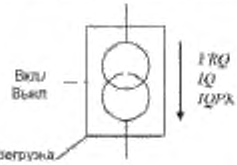


Рисунок 13 — Эталонная схема драйвера линии

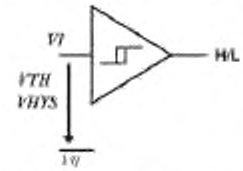
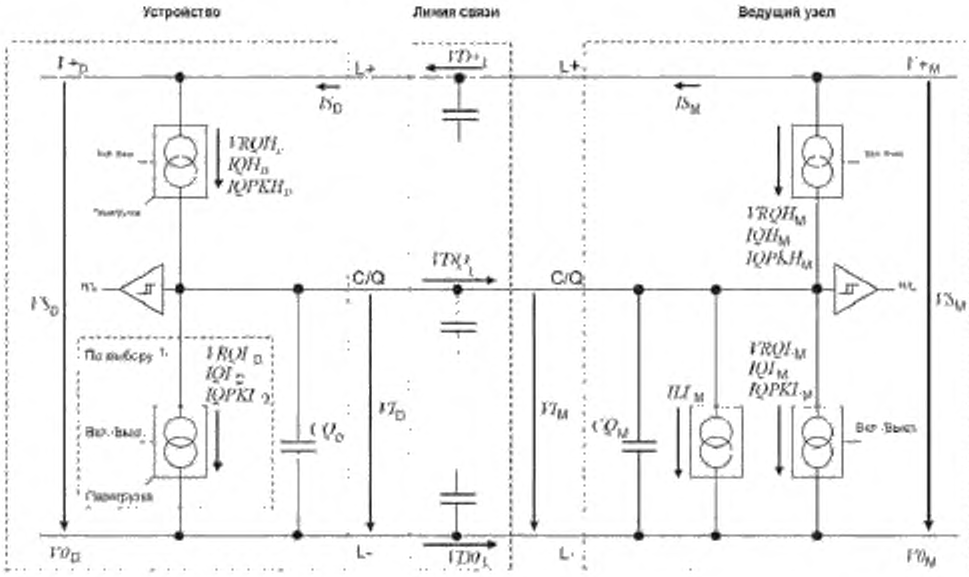


Рисунок 14 — Эталонная схема приемника

На рисунке 15 показана эталонная схема взаимодействия Ведущего узла и Устройства для системы 3-проводного присоединения SDCI.



1) По выбору: драйвер нижнего плеча (только двухтактный).

Рисунок 15 — Эталонная схема системы 3-проводного присоединения SDCI

Следующие рисунки и таблицы параметров относятся к определениям уровня напряжения, показанным на рисунке 16. Индексы параметра относятся к Ведущему узлу (M), Устройству (D) или линии (L). Перепады напряжения в линии $VD+_L$, VDQ_L и $VD0_L$ неявно определяются в подразделе 5.5 через параметры кабеля.

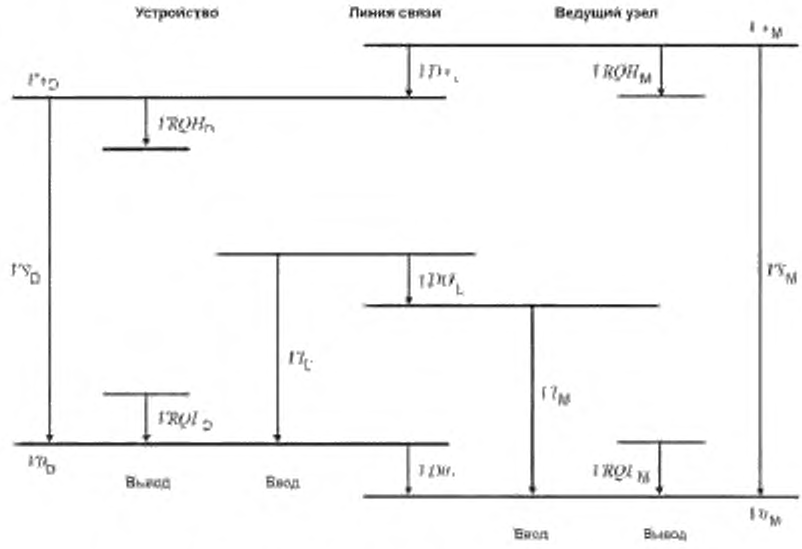


Рисунок 16 — Определения уровня напряжения

5.3.2.2 Приемник

Определения диапазона напряжения и порога переключения одинаковы для Ведущего узла и Устройства. Применяются определения, приведенные в таблице 5.

Таблица 5 — Электрические характеристики приемника

Свойство	Наименование	Минимум	Норма	Максимум	Единица измерения	Примечание
$VTHH_{D,M}$	Входной порог высокого сигнала	10,5	н/д	13	В	См. примечание 1
$VTHL_{D,M}$	Входной порог низкого сигнала	8	н/д	11,5	В	См. примечание 1
$VHYS_{D,M}$	Гистерезис между входными порогами высокого и низкого сигналов	0	н/д	н/д	В	Не должен быть отрицательным. См. примечание 2
$VIL_{D,M}$	Допустимый диапазон напряжений низкого сигнала	$V0_{D,M}$ – 1,0	н/д	н/д	В	Относительно соответствующего отрицательного напряжения питания
$VIH_{D,M}$	Допустимый диапазон напряжений высокого сигнала	н/д	н/д	$V+_{D,M}$ + 1,0	В	Относительно соответствующего положительного напряжения питания
Примечание 1 — Пороги совместимы с определениями цифровых вводов типа 1 в соответствии с МЭК 61131-2. Примечание 2 — Напряжение гистерезиса $VHYS = VTHH – VTHL$.						

На рисунке 17 показаны пороги переключения для определения низкого и высокого сигналов.

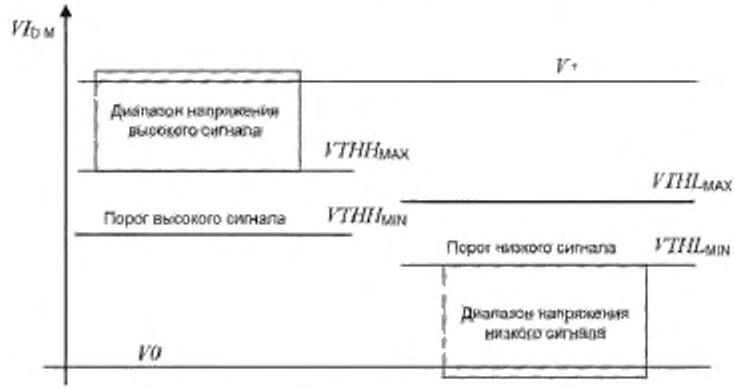


Рисунок 17 — Пороги переключения

5.3.2.3 Порт Ведущего узла

Определения, приведенные в таблице 6, действительны для электрических характеристик порта Ведущего узла.

Таблица 6 — Электрические характеристики порта Ведущего узла

Свойство	Наименование	Минимум	Норма	Максимум	Единица измерения	Примечание
VS_M	Напряжение питания для Устройств	20	24	30	В	См. рисунок 16
IS_M	Ток питания для Устройств	200	н/д	н/д	мА	Для токов свыше 200 мА требуется внешний источник питания
$ISIR_M$	Возможности импульса тока для Устройств	400	н/д	н/д	мА	Возможность тока питания Ведущего узла для минимального периода 50 мс при 18 В после включения питания порта
ILL_M	Ток нагрузки или ток разряда для: 0 В < VIM < 5 В; 5 В < VIM < 15 В; 15 В < VIM < 30 В	 0 5 5	 н/д н/д н/д	 15 15 15	 мА мА мА	См. примечание 1
$VRQH_M$	Остаточное напряжение высокого сигнала	н/д	н/д	3	В	Перепад напряжения относительно $V+_M$ при максимальном токе драйвера IQH_M
$VRQL_M$	Остаточное напряжение низкого сигнала	н/д	н/д	3	В	Перепад напряжения относительно $V0_M$ при максимальном токе драйвера IQL_M
IQH_M	Ток драйвера постоянного тока высокого сигнала	100	н/д	н/д	мА	

Окончание таблицы 6

Свойство	Наименование	Минимум	Норма	Максимум	Единица измерения	Примечание
$IQPKH_M$	Выходной пиковый ток высокого сигнала	500	н/д	н/д	мА	Абсолютная величина. См. примечание 2
IQL_M	Ток драйвера постоянного тока низкого сигнала	100	н/д	н/д	мА	
$IQPKL_M$	Выходной пиковый ток низкого сигнала	500	н/д	н/д	мА	Абсолютная величина. См. примечание 2
CQ_M	Входная емкость	н/д	н/д	1,0	нФ	f = от 0 до 4 МГц
Примечание 1 — Токи совместимы с определениями цифровых вводов типа 1 в МЭК 61131-2. Тем не менее для диапазона 5 В < VI_M < 15 В минимальный ток равен 5 мА вместо 2 мА, чтобы достичь достаточно низких скоростей нарастания в Устройствах, снабженных только драйвером переключения питания. Примечание 2 — Ток запроса пробуждения (см. 5.3.3.3).						

5.3.2.4 Устройство

Определения в таблице 7 действительны для электрических характеристик Устройства.

Таблица 7 — Электрические характеристики Устройства

Свойство	Наименование	Минимум	Норма	Максимум	Единица измерения	Примечание
VS_D	Напряжение питания	18	24	30	В	См. рисунок 16
ΔVS_D	Пульсация	н/д	н/д	1,3	В	Границы полного размаха абсолютного значения не должны быть превышены. f_{ripple} = до 100 кГц
$VRQH_D$	Остаточное напряжение высокого сигнала	н/д	н/д	3	В	Перепад напряжения относительно V_{+D} (МЭК 60947-5-2)
$VRQL_D$	Остаточное напряжение низкого сигнала	н/д	н/д	3	В	Перепад напряжения относительно $V0_D$
IQH_D	Ток на выходе драйвера постоянного тока переключения питания (в состоянии On)	50	н/д	Минимум ($IQPKL_M$)	мА	Минимальное значение из-за возврата к цифровому выводу в соответствии с МЭК 61131-2, тип 2
IQL_D	Ток на выходе драйвера постоянного тока питания (в состоянии On)	0	н/д	Минимум ($IQPKH_M$)	мА	Только для стадий двухтактного выхода
IQQ_D	Ток в рабочей точке к VO_D (состояние Off)	0	н/д	15	мА	Ток при понижающемся напряжении или остаточный ток при деактивированном драйвере
CQ_D	Входная емкость	0	н/д	1,0	нФ	Эффективная емкость между C/Q и L+ или L– Устройства в состоянии приема

Значение 1 нФ применимо для скорости передачи данных 230,4 кбит/с. Входная емкость CQ_D может быть увеличена до 10 нФ в случае двухтактной конструкции при работе на меньших скоростях передачи данных при условии, что все требования к динамическим параметрам, установленные в 5.3.3.2, соблюдены.

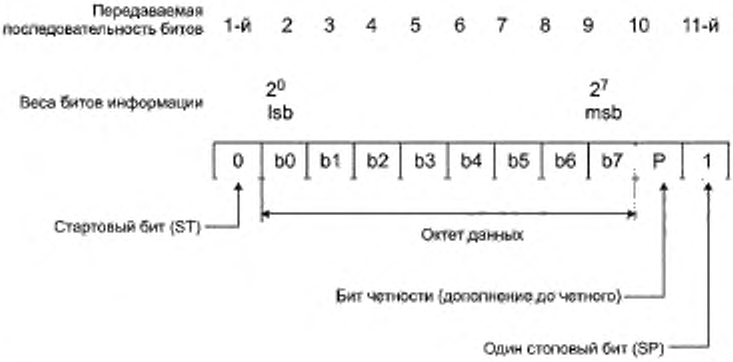
5.3.3 Временные требования

5.3.3.1 Способ передачи данных

Для побитового кодирования применяется модуляция без возврата к нулю. Логическое значение «1» соответствует разности напряжений 0 В между линиями C/Q и L–. Логическое значение «0» соответствует разности напряжений 24 В между линиями C/Q и L–.

Уровень разомкнутой цепи на линии C/Q относительно линии L– равен 0 В. Стартовый бит имеет логическое значение «0», т. е. +24 В.

Для кодирования по октетам данных применяется фрейм UART. Формат фрейма UART в технологии SDCI — это строка битов, структурированная, как показано на рисунке 18.



Обозначения:
lsb — младший бит;
msb — старший бит.

Рисунок 18 — Формат фрейма UART в технологии SDCI

Определение формата фрейма UART основано на ИСО 1177 и ИСО/МЭК 2022.

5.3.3.2 Характеристики передачи данных

Временные характеристики передачи данных демонстрируются в форме глазковой диаграммы с допустимыми диапазонами сигналов (см. рисунок 19). Данные диапазоны применимы для приемника как в Ведущем узле, так и в Устройстве.

Независимо от граничных условий датчик будет генерировать вольтовую характеристику в соединении линии C/Q приемника, которая находится в допустимом диапазоне глазковой диаграммы.

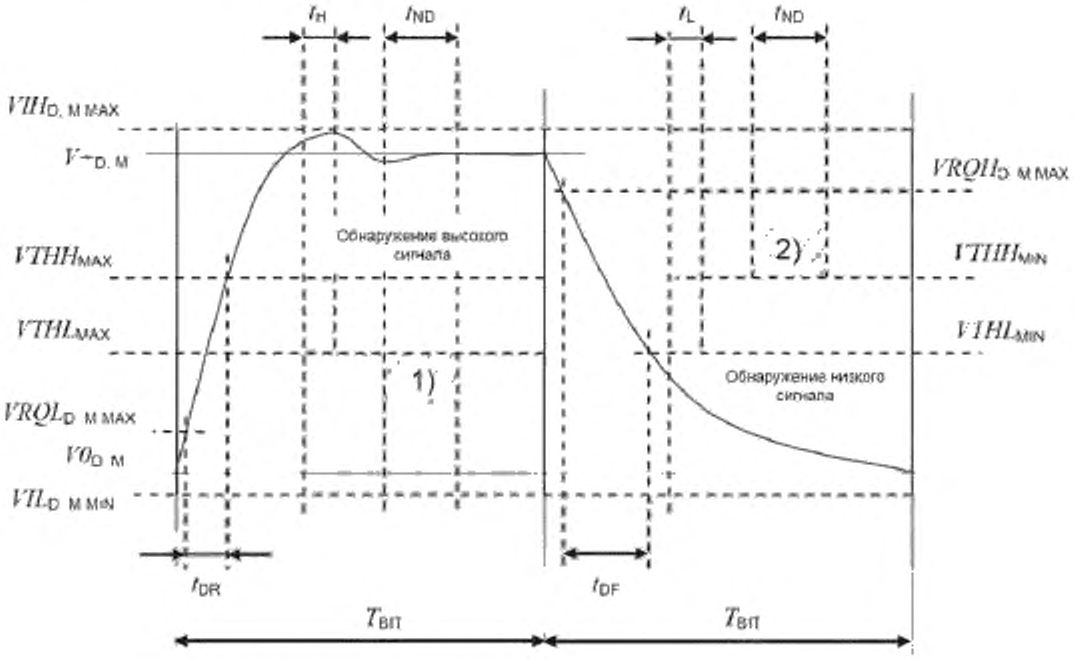
Приемник будет обнаруживать биты, как сигнал допустимой формы в допустимом диапазоне глазковой диаграммы на соединении линии C/Q. Формы сигнала в областях «необнаружения» (ниже $VTHL_{MAX}$ или $VTHH_{MIN}$ и в пределах t_{ND}) не будут приводить к недопустимым битам.

Для того чтобы фреймы UART правильно обнаруживались, на стороне приемника должны быть характеристики сигнала, показанные на рисунке 20. Должно учитываться время задержки сигнала между сигналом линии C/Q и фреймом UART. Время T_{BIT} всегда указывает на скорость передачи битов в приемнике.

Для каждого бита n в битовой последовательности (n = 1 … 11) фрейма UART время $(n - r)T_{BIT}$ (см. таблицу 8 для значений r) определяет время, через которое будет достигаться правильный уровень в диапазонах высокого и низкого сигналов, как продемонстрировано в глазковой диаграмме на рисунке 19. Время $(n - s)$ $TBIT$ (см. таблицу 8 для значений s) описывает время, которое будет проходить до изменения уровня. Если возникают вопросы по характеристикам сигнала, всегда следует обращаться к глазковой диаграмме на рисунке 19.

Данное представление позволяет оценивать влияние таких определяющих параметров приемника, как точность скорости передачи данных, деформация битовой ширины и скорость нарастания.

Динамические характеристики передачи данных приведены в таблице 8.



1) — не обнаружен низкий сигнал; 2) — не обнаружен высокий сигнал.

Рисунок 19 — Глазковая диаграмма
для обнаружения высокого и низкого сигналов

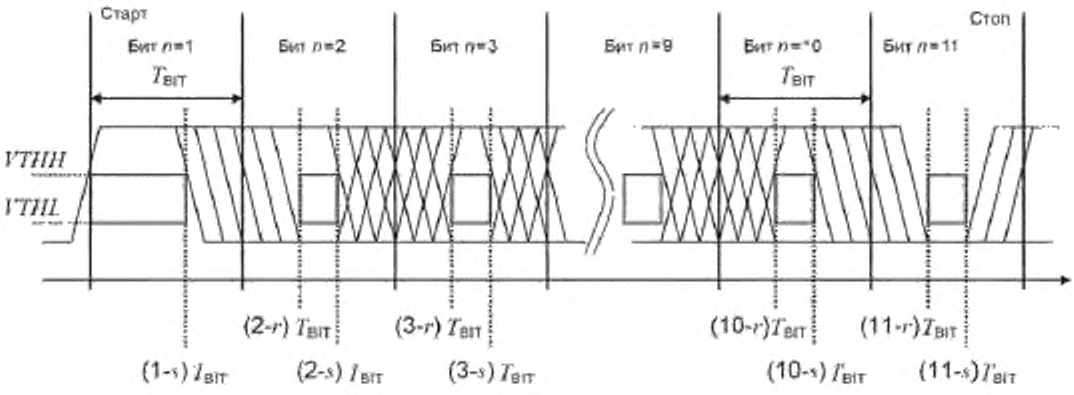


Рисунок 20 — Глазковая диаграмма
для правильного обнаружения фрейма UART

Таблица 8 — Динамические характеристики передачи данных

Свойство	Наименование	Минимум	Норма	Максимум	Единица измерения	Примечание
f_{DTR}	Скорость передачи данных	н/д	4,8 38,4 230,4	н/д	кбит/с	COM1 COM2 COM3
T_{BIT}	Время передачи бита при 4,8 кбит/с при 38,4 кбит/с при 230,4 кбит/с		 208,33 26,04 4,34		 мкс мкс мкс	
Δf_{DTRM}	Точность скорости передачи Ведущего узла при 4,8 кбит/с при 38,4 кбит/с при 230,4 кбит/с	 – 0,1 – 0,1 – 0,1	 н/д н/д н/д	 + 0,1 + 0,1 + 0,1	 % % %	Допуск скорости передачи Ведущего узла $\Delta T_{BIT}/T_{BIT}$
r	Начало времени обнаружения бита относительно переднего фронта стартового бита	0,65	н/д	н/д	—	В каждом случае вычисляется от конца бита при интервале выборки UART, равном 8
s	Конец времени обнаружения бита относительно переднего фронта стартового бита	н/д	н/д	0,22	—	В каждом случае вычисляется от конца бита при интервале выборки UART, равном 8
T_{DR}	Время нарастания при 4,8 кбит/с при 38,4 кбит/с при 230,4 кбит/с	0 0 0 0	н/д н/д н/д н/д	0,20 41,7 5,2 869	T_{BIT} мкс мкс нс	Относительно единицы измерения времени передачи бита
t_{DF}	Время отката при неудаче при 4,8 кбит/с при 38,4 кбит/с при 230,4 кбит/с	 0 0 0 0	 н/д н/д н/д н/д	 0,20 41,7 5,2 869	 T_{BIT} мкс мкс нс	Относительно единицы измерения времени передачи бита
t_{ND}	Время подавления шума	н/д	н/д	1/16	T_{BIT}	Допустимый период времени приемного сигнала выше/ниже порога обнаружения до собственно обнаружения
t_H	Время обнаружения высокого сигнала	1/16	н/д	н/д	T_{BIT}	Период времени приемного сигнала над порогом обнаружения для высокого уровня
t_L	Время обнаружения низкого сигнала	1/16	н/д	н/д	T_{BIT}	Период времени приемного сигнала под порогом обнаружения для низкого уровня

Параметры r и s применяются к соответствующей стороне приемника Ведущего узла или Устройства. Такое определение позволяет дать более гибкое определение точности генератора, деформации бита и скорости нарастания. Общая деформация ширины бита на последнем бите фрейма UART предоставляет правильную оценку уровня в диапазоне рисунка 20.

5.3.3.3 Импульс тока пробуждения

Функциональная возможность пробуждения применяется для запроса на перевод Устройства в режим COMx.

Вызов услуги PL_WakeUp.req на DL инициирует процесс пробуждения (см. 5.2.2.2).

Запрос пробуждения (WURQ) начинается с импульсом тока, возбужденным Ведущим узлом (в порту), на период времени T_{WU}. Запрос пробуждения включает в себя следующие фазы (см. рисунок 21):

a) подача Ведущим узлом тока IQWU в зависимости от уровня в соединении линии C/Q. Для входного сигнала, соответствующего логической «1», это ток источника; для логического сигнала, соответствующего логическому «0», это токовый сток;

b) время задержки на Устройстве до наступления готовности к приему.

WURQ может быть обнаружен Устройством через изменение напряжения в линии C/Q или вычисление тока соответствующего элемента драйвера в период времени T_{WU}. На рисунке 21 показаны примеры Устройства с малой выходной мощностью.

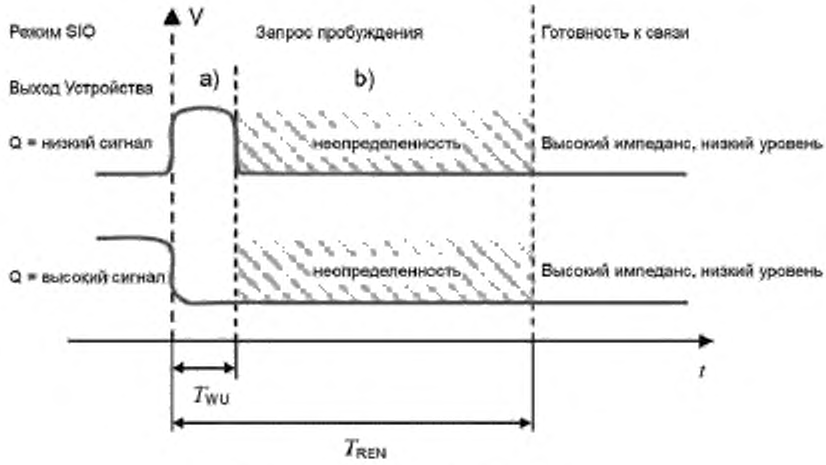


Рисунок 21 — Запрос пробуждения

В таблице 9 определены ток и временные характеристики, связанные с запросом пробуждения. Значения $IQPKL_M$ и $IQPKH_M$ приводятся в таблице 6.

Таблица 9 — Характеристики запроса пробуждения

Свойство	Наименование	Минимум	Норма	Максимум	Единица измерения	Примечания
IQ_{WU}	Амплитуда импульса тока пробуждения Ведущего узла	$IQPKL_M$ или $IQPKH_M$	н/д	н/д	мА	Импульс тока с последующим состоянием переключения Устройства
T_{WU}	Длительность импульса тока пробуждения Ведущего узла	75	н/д	85	мкс	Свойство Ведущего узла
T_{REN}	Задержка готовности к приему	н/д	н/д	500	мкс	Свойство Устройства

5.4 Источник питания

5.4.1 Опции источника питания

Система присоединения SDCI обеспечивает выделенные линии питания в дополнение к линии сигнала. Секция связи Устройства всегда будет питаться Ведущим узлом, применяя линии питания, определенные в системе 3-проводной связи (Источник питания 1).

Максимальный ток питания, доступный от порта Ведущего узла, определяется в таблице 6.

Прикладная часть Устройства может получать питание одним из трех способов:

- через линии питания системы 3-проводного присоединения SDCI (порты класса А), применяя Питание 1;
- через дополнительные линии питания системы 5-проводного присоединения SDCI (порты класса В), применяя дополнительный источник питания в Ведущем узле (Питание 2);
- через локальный источник питания в Устройстве (зависит от конструкции).

Порт класса А допускает потребление тока до 200 мА, как указано в таблице 6. Максимальное потребление энергии в порте класса В зависит от выбранного метода. Соединение М12 позволяет увеличивать ток до 3,5 А.

5.4.2 Требования к подаче питания

На рисунке 22 показано, как поведение подачи питания в Устройстве определяется временем нарастания импульса источника Питания 1 и внутренним для Устройства временем подготовки к операции пробуждения.

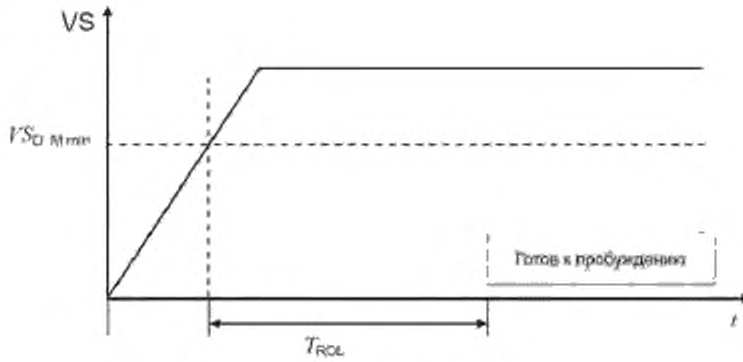


Рисунок 22 — Регламент подачи питания для источника Питания 1

После подачи питания Устройство должно достичь состояния готовности к пробуждению в предельные сроки, указанные в таблице 10.

Таблица 10 — Регламент подачи питания

Свойство	Наименование	Минимум	Норма	Максимум	Единица измерения	Примечания
T_{RDL}	Готовность к пробуждению после подачи питания	н/д	н/д	300	мс	Время нарастания импульса Устройства до его готовности к обнаружению сигнала пробуждения (см. примечание)
Примечание — Эквивалентно времени задержки до готовности в соответствии с МЭК 60947-5-2.						

5.5 Среда передачи данных

5.5.1 Соединители

Назначение контактов Ведущего узла и Устройства основано на спецификациях, приведенных в МЭК 60947-5-2, с расширениями, определенными в разделах ниже. Порты класса А применяют соединители М5, М8 и М12 не более чем с четырьмя контактами. Порты класса В применяют

только соединители М12 с пятью контактами. Соединители М12 имеют код «А» в соответствии с МЭК 61076-2-101.

Примечание — Для наследования и совместимости может применяться прямая проводка различных типов соединений вместо поставляемой при условии, что она не нарушает электрических характеристик и применяет наименование сигналов, установленное в настоящем стандарте.

Гнездовые соединители предназначены для Ведущего узла, а штекерные соединителя — для Устройства. В таблице 11 приведены назначения контактов и на рисунке 23 показано расположение и механическое кодирование для соединений М12, М8 и М5.

Таблица 11 — Схема назначений контактов

Контакт	Сигнал	Наименование	Примечание
1	L+	Источник питания (+)	См. таблицу 7
2	I/Q P24	NC/DI/DO (порт класса А) P24 (порта класса В)	Опция 1: NC (не соединен). Опция 2: DI (цифровой ввод). Опция 3: DI → сконфигурированный DO. Опция 4: Дополнительный источник питания для мощных Устройств (порт класса В)
3	L–	Источник питания (–)	См. таблицу 7
4	C/Q	SIO/SDCI	Режим SIO (DI/DO) или SDCI (электрические характеристики DO см. в таблице 6)
5	NC N24	NC (порт класса А) N24 (порт класса В)	Опция 1: Не соединяется на стороне Ведущего узла (порт класса А). Опция 2: Ссылка на дополнительный источник питания (порт класса В)
Примечание — Соединение М12 всегда исполняется в версии с пятью контактами на стороне Ведущего узла (гнездовое).			

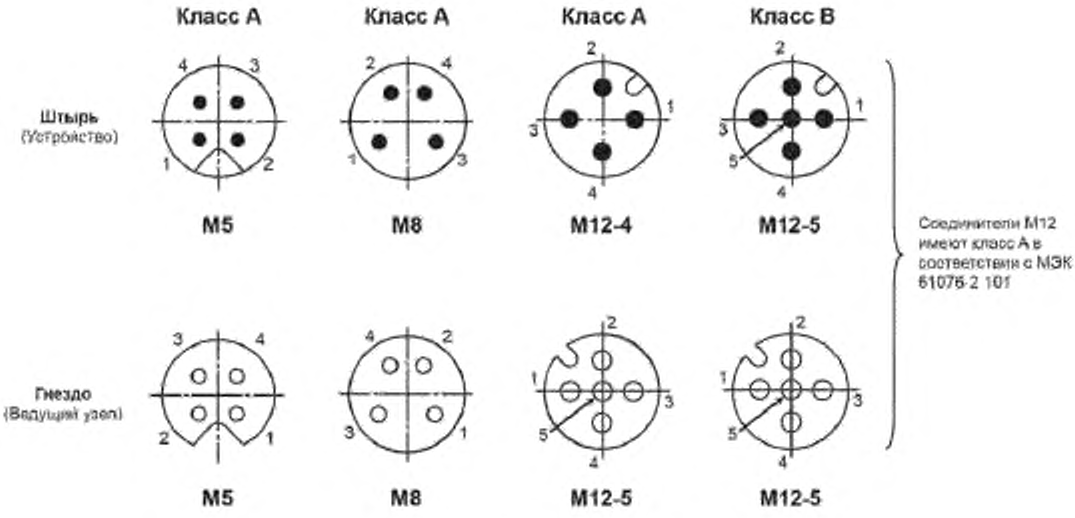


Рисунок 23 — Расположение контактов (вид спереди)

На рисунке 24 показано расположение двух классов портов — A и B. Порты класса B должны быть маркированы для их отличия от портов класса A, так как данные классы портов несовместимы.

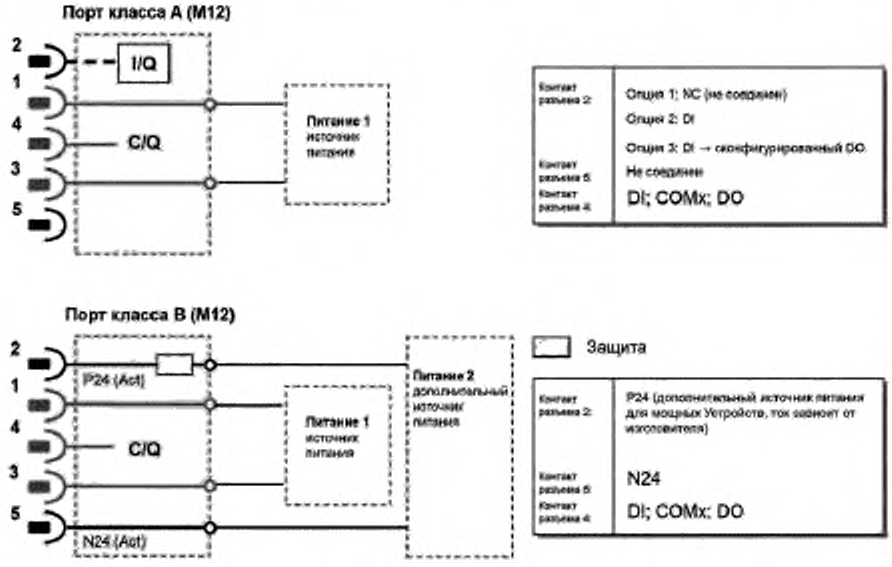


Рисунок 24 — Определения портов классов A и B

5.5.2 Кабель

Среда передачи данных связи SDCI — многожильный кабель с тремя или более жилами. Определения в следующих разделах неявно охватывают определения статического напряжение, показанные в таблице 5 и на рисунке 16. Для обеспечения функциональной надежности свойства кабеля должны соответствовать таблице 12.

Таблица 12 — Характеристики кабеля

Свойство	Минимум	Норма	Максимум	Единица измерения
Длина	0	н/д	20	м
Полное сопротивление шлейфа RL_{eff}	н/д	н/д	6,0	Ом
Эффективная емкость линии CL_{eff}	н/д	н/д	3,0	нФ (< 1 МГц)

Сопротивление шлейфа RL_{eff} и эффективная емкость линии CL_{eff} могут быть измерены, как показано на рисунке 25.

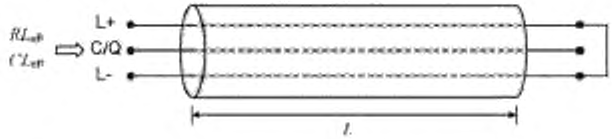


Рисунок 25 — Схема измерения эффективной емкости линии и сопротивления шлейфа

В таблице 13 приведены токоведущие жилы кабеля и назначенные им цветовые коды.

Таблица 13 — Схема кабельного соединителя

Сигнал	Наименование	Цвет	Примечание
L–	Источник питания (–)	Синий[a)]	Систем 3-проводного соединения SDCI
C/Q	Сигнал связи	Черный[a)]	Систем 3-проводного соединения SDCI
L+	Источник питания (+)	Коричневый[a)]	Систем 3-проводного соединения SDCI
I/Q	Цифровой ввод или цифровой вывод	Белый[a)]	Необязательный
P24	Дополнительный источник питания (+)	Любой другой	Необязательный
N24	Дополнительный источник питания (–)	Любой другой	Необязательный
a) В соответствии с МЭК 60947-5-2.			

6 Стандартный ввод и вывод (SIO)

На рисунках 83 и 94 показано, как режим SIO позволяет Устройству обходить уровни связи SDCI и отображать сигналы цифрового ввода DI или цифрового вывода DO прямо в сообщение обмена данными шины или системы более высокого уровня. Переход между режимами SDCI и SIO определяется конфигурацией пользователя или (неявно) услугами приложений Ведущего узла. SM следит за соответствующей инициализацией или деактивацией уровней связи SDCI и PL (переключатель режима). Характеристики интерфейсов сигналов DI и DO извлекаются из характеристик, установленных в МЭК 61131-2 для типа 1.

7 Канальный уровень (DL)

7.1 Общие положения

Канальные уровни SDCI обеспечивают доставку сообщений между Ведущим узлом и Устройством через физический канал. DL применяет несколько типов M-последовательностей («последовательностей сообщений») для различных категорий данных.

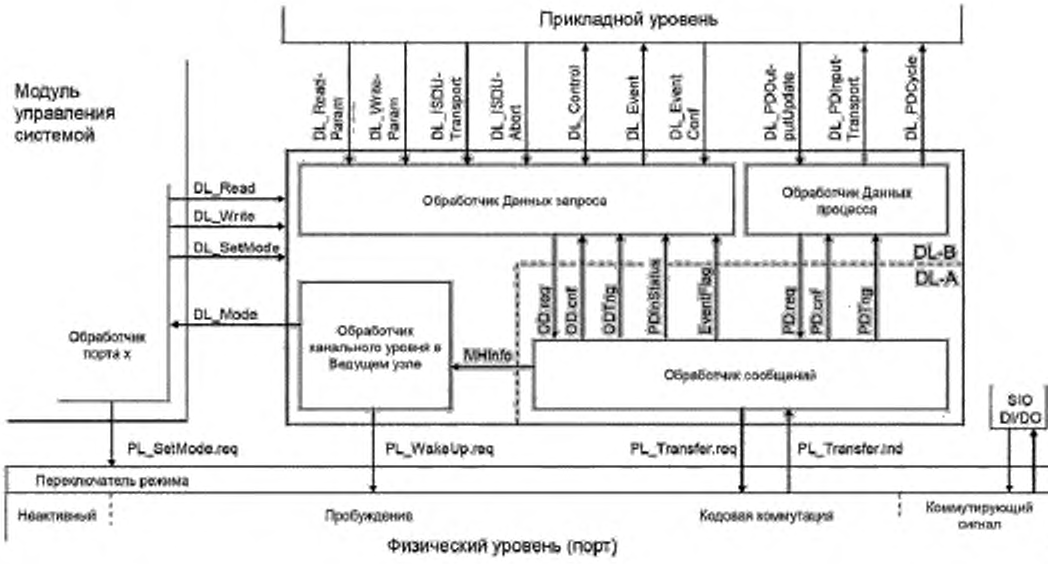
Набор услуг DL доступен AL для обмена PD и OD. Другой набор услуг DL доступен SM для поиска и выборки идентификационных параметров Устройства и установки конечных машин на DL. DL применяет услуги PL для управления PL и для обмена фреймами UART. DL следит за обнаружением ошибок в сообщениях (как внутренних, так сообщенных из PL) и предпринимает подходящие меры по их устранению.

DL структурируются по категории данных, в обработчике PD и OD, которые, в свою очередь, применяют обработчик сообщений для требуемой передачи сообщений. Специальные режимы портов Ведущего узла, такие как пробуждение, COMx и SIO (деактивация связи), требуют выделенных обработчиков DL в DL Ведущего узла. Модуляция сигнала пробуждения требует обнаружения сигнала на стороне Устройства и, следовательно, обработчиков DL на стороне Устройства. Каждый обработчик включает собственную конечную машину.

DL подразделяется на секцию DL-A с собственными внутренними услугами и секцию DL-B с внешними услугами.

DL применяет дополнительные внутренние административные вызовы между обработчиками, которые определяются в секции «внутренних элементов» соответствующих таблиц переходов состояний.

На рисунке 26 показан обзор структуры и услуг DL Ведущего узла.



Примечание — Используются условные обозначения пункта 3.3.5.

Рисунок 26 — Структура и услуги DL (Ведущий узел)

На рисунке 27 показан обзор структуры и услуг DL Устройства.

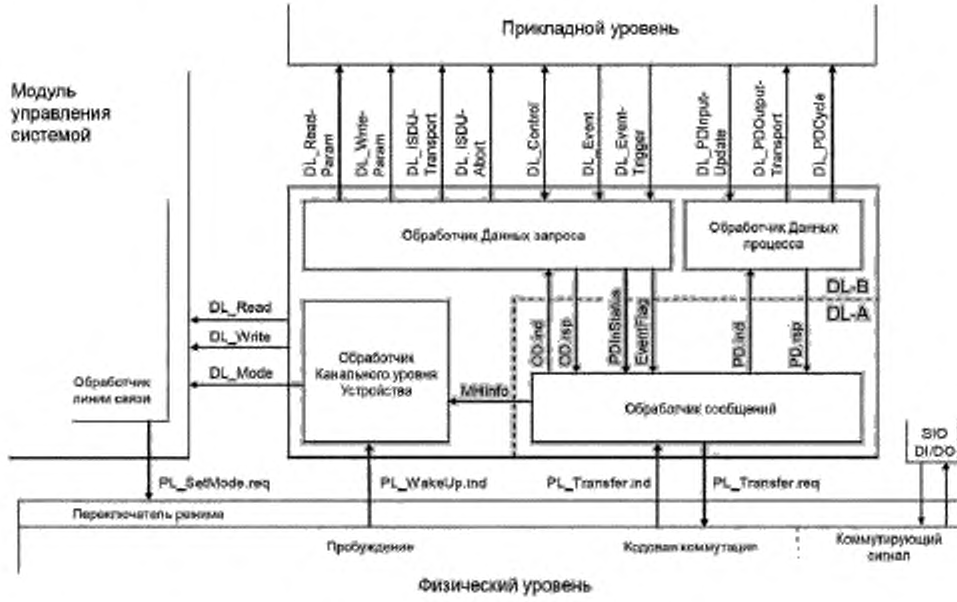


Рисунок 27 — Структура и услуги DL (Устройство)

7.2 Услуги канального уровня

7.2.1 Услуги секции DL-B канального уровня

7.2.1.1 Обзор услуг в Ведущем узле и Устройстве

В разделе 7 устанавливаются услуги DL, предоставляемые AL и SM через внешние интерфейсы DL. В таблице 14 приведены распределения Ведущего узла и Устройства их ролям инициатора и приемника для отдельных услуг DL. Пустые поля указывают, что данная услуга отсутствует на Ведущем узле или Устройстве.

Таблица 14 — Распределение услуг в Ведущем узле и Устройстве

Имя услуги	Ведущий узел	Устройство
DL_ReadParam	R	I
DL_WriteParam	R	I
DL_ISDUTransport	R	I
DL_ISDUAbort	R	I
DL_PDOutputUpdate	R	
DL_PDOutputTransport		I
DL_PDInputUpdate		R
DL_PDInputTransport	I	
DL_PDCycle	I	I
DL_SetMode	R	
DL_Mode	I	I
DL_Event	I	R
DL_EventConf	R	
DL_EventTrigger		R
DL_Control	I/R	R/I
DL_Read	R	I
DL_Write	R	I
Обозначения (см. 3.3.4): I — Инициатор услуги; R — Приемник (ответчик) услуги.		

В подразделе 3.3 приводятся условные обозначения и объясняется, как читать описания услуг, приведенных в 7.2, 8.2, 9.2.2 и 9.3.2.

7.2.1.2 Услуга DL_ReadParam

Услуга DL_ReadParam применяется AL для чтения значений параметров из Устройства через канал связи страниц. Параметры сервисных примитивов приведены в таблице 15.

Таблица 15 — Услуга DL_ReadParam

Имя параметра	.req	.cnf	.ind
Argument	M		M
Address	M		M
Result (–)		S	
Value		M	
Result (–)		S	
ErrorInfo		M	

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

Address (Адрес)
Данный параметр содержит адрес параметра, затребованного Устройством, т. е. адреса параметров Устройства на канале связи страницы (см. таблицу В.1).

Допустимые значения 0 … 31.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Value (Значение)
Данный параметр содержит считанные значения параметров Устройства.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неуспешно.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
NO_COMM (нет доступной связи);
STATE_CONFLICT (в текущем состоянии услуга недоступна).

7.2.1.3 Услуга DL_WriteParam

Услуга DL_WriteParam применяется AL для записи значения параметра в Устройства через канал связи страниц. Параметры сервисных примитивов приведены в таблице 16.

Таблица 16 — Услуга DL_WriteParam

Имя параметра	.req	.cnf	.ind
Argument	M		M
Address	M		M
Value	M		M
Result (–)		S	
Result (–)		S	
ErrorInfo		M	

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

Address (Адрес)
Данный параметр содержит адрес параметра, затребованного Устройством, т. е. адреса параметров Устройства на канале связи страницы.

Допустимые значения от 16 до 31 в соответствии с правами доступа параметра Устройства.

Value (Значение)
Данный параметр содержит подлежащее записи значение параметра Устройства.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неуспешно.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
NO_COMM (нет доступной связи);
STATE_CONFLICT (в текущем состоянии услуга недоступна).

7.2.1.4 Услуга DL_Read

Услуга DL_Read применяется SM для чтения значения параметра Устройства через канал связи страниц. Параметры сервисных примитивов приведены в таблице 17.

Таблица 17 — Услуга DL_Read

Имя параметра	.req	.cnf	.ind
Argument	M		M
Address	M		M
Result (+)		S	
Value		M	
Result (–)		S	
ErrorInfo		M	

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

Address (Адрес)
Данный параметр содержит адрес параметра, затребованного Устройством, т. е. адреса параметров Устройства на канале связи страницы (см. таблицу В.1).

Допустимые значения от 0 до 15 в соответствии с правами доступа параметра Устройства.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Value (Значение)
Данный параметр содержит считанные значения параметров Устройства.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

ErrorInfor (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
NO_COMM (нет доступной связи);
STATE_CONFLICT (в текущем состоянии услуга недоступна).

7.2.1.5 Услуга DL_Write

Услуга DL_Write применяется SM для записи значения параметра в Устройство через канал связи страниц. Параметры сервисных примитивов приведены в таблице 18.

Таблица 18 — Услуга DL_Write

Имя параметра	.req	.cnf	.ind
Argument	M		M
Address	M		M
Value	M		M
Result (+)		S	
Result (–)		S	
ErrorInfo		M	

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

Address (Адрес)
Данный параметр содержит адрес параметра, затребованного Устройством, т. е. адреса параметров Устройства на канале связи страницы.

Допустимые значения от 0 до 15 в соответствии с правами доступа параметра Устройства.

Value (Значение)
Данный параметр содержит подлежащее записи значение параметра Устройства.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неуспешно.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
NO_COMM (нет доступной связи);
STATE_CONFLICT (в текущем состоянии услуга недоступна).

7.2.1.6 Услуга DL_ISDUTransport

Услуга DL_ISDUTransport применяется для передачи ISDU. Данная услуга применяется Ведущим узлом для посылки запроса услуги из AL Ведущего узла в Устройство. Она также применяется Устройством для посылки ответа на запрос услуги из AL Устройства. Параметры сервисных примитивов приведены в таблице 19.

Таблица 19 — Услуга DL_ISDUTransport

Имя параметра	.req	.ind	.cnf	.res
Argument	M	M		
ValueList	M	M		
Result (+)			S	S
Data			C	C
Descriptor			M	M
Result (–)			S	S
ISDUTransportErrorInfo			M	M

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

ValueList (СписокЗначений)
Данный список содержит используемые рабочие параметры.
Тип параметра: Запись.

Index (Индекс)
Допустимые значения 2 … 65 535 (ограничения описаны в B.2.1).

Subindex (Субиндекс)
Допустимые значения 0 … 255.

Data (Данные)
Тип параметра: Строка октетов.

Direction (Направление)
Допустимые значения:
READ (операция Чтения);
WRITE (операция Записи).

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Data (Данные)
Тип параметра: Строка октетов.

Descrirtor (Описатель)
Допустимые значения: ответ на запрос I-Service Устройства в соответствии с таблицей A.12.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

ISDUTransportErrorInfo (ИнформацияОбОшибкеТранспортаISDU)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
NO_COMM (нет доступной связи);
STATE_CONFLICT (в текущем состоянии услуга недоступна);
ISDU_TIMEOUT (время подтверждения ISDU истекло, см. таблицу 97);
SDU_NOT_SUPPORTED (ISDU не реализован);
VALUE_OUT_OF_RANGE (значение параметра услуги нарушает границы диапазона).

7.2.1.7 Услуга DL_ISDUAbort

Услуга DL_ISDUAbort прекращает текущую передачу ISDU. Данная услуга не имеет параметров. Сервисные примитивы приведены в таблице 20.

Таблица 20 — Услуга DL_ISDUAbort

Имя параметра	.req	.cnf
<отсутствуют>		

Услуга возвращает подтверждение после прекращения передачи ISDU.

7.2.1.8 Услуга DL_PDOutputUpdate

Прикладной слой Ведущего узла применяет услугу DL_PDOutputUpdate для модификации данных вывода (PD от Ведущего узла к Устройству) на DL. Параметры сервисных примитивов приведены в таблице 21.

Таблица 21 — Услуга DL_PDOutputUpdate

Имя параметра	.req	.cnf
Argument	M	
OutputData	M	
Result (+)		S
TransportStatus		M
Result (–)		S
ErrorInfo		M

Argument (Аргумент)

Определяемые услугой данные передаются в данном параметре.

OutputData (ДанныеВывода)

Данный параметр содержит PD, предоставленные прикладным слоем.

Тип параметра: Строка октетов.

Result (+) (ПоложительныйРезультат)

Данный параметр выбора указывает, что услуга выполнена успешно.

TransportStatus (СостояниеТранспорта)

Данный параметр показывает, находится ли транспортный уровень в состоянии, разрешающем передачу данных партнеру(ам) по связи.

Допустимые значения:
YES (передача данных разрешена);
NO (передача данных не разрешена).

Reslut (–) (ОтрицательныйРезультат)

Данный параметр выбора указывает, что услуга завершилась неуспешно.

ErrorInfor (ИнформацияОбОшибке)

Данный параметр содержит информацию об ошибке.

Допустимые значения:
NO_COMM (нет доступной связи);
STATE_CONFLICT (в текущем состоянии услуга недоступна).

7.2.1.9 Услуга DL_PDOutputTransport

DL на Устройстве применяет услугу DL_PDOutputTransport для передачи содержания выходных PD AL (от Ведущего узла к Устройству). Параметры сервисных примитивов приведены в таблице 22.

Таблица 22 — Услуга DL_PDOutputTransport

Имя параметра	.ind
Argument	M
OutputData	M

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

OutputData (ДанныеВывода)
Данный параметр содержит PD, передаваемые прикладному слою.

Тип параметра: Строка октетов.

7.2.1.10 Услуга DL_PDInputUpdate

Прикладной слой Ведущего узла применяет услугу DL_PDInputUpdate для модификации входных данных (PD от Устройства к Ведущему узлу) на DL. Параметры сервисных примитивов приведены в таблице 23.

Таблица 23 — Услуга DL_PDInputUpdate

Имя параметра	.req	.cnf
Argument	M	
InputData	M	
Result (+)		S
TransportStatus		M
Result (–)		S
ErrorInfor		M

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

InputData (ДанныеВвода)
Данный параметр содержит PD, предоставленные прикладным слоем.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

TransportStatus (СостояниеТранспорта)
Данный параметр показывает, находится ли транспортный уровень в состоянии, разрешающем передачу данных партнеру(ам) по связи.

Допустимые значения:
YES (передача данные разрешена);
NO (передача данные не разрешена).

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неуспешно.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
NO_COMM (нет доступной связи);
STATE_CONFLICT (в текущем состоянии услуга недоступна).

7.2.1.11 Услуга DL_PDInputTransport

DL на Устройстве применяет услугу DL_PDInputTransport для передачи содержания входных PD (от Устройства к Ведущему узлу) AL. Параметры сервисных примитивов приведены в таблице 24.

Таблица 24 — Услуга DL_PDInputTransport

Имя параметра	.ind
Argument	M
InputData	M

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

InprtData (ДанныеВвода)
Данный параметр содержит PD, передаваемые прикладному слою.

Тип параметра: Строка октетов.

7.2.1.12 Услуга DL_PDCycle
DL применяет услугу DL_PDCycle для указания конца цикла PD AL.
Данная услуга не имеет параметров. Сервисные примитивы приведены в таблице 25.

Таблица 25 — Услуга DL_PDCycle

Имя параметра	.ind
<отсутствуют>	

7.2.1.13 Услуга DL_SetMode
Услуга DL_SetMode применяется управлением системой для создания конечной машины канального слоя и посылки характеристических значений, требуемых для работы на DL. Параметры сервисных примитивов приведены в таблице 26.

Таблица 26 — Услуга DL_SetMode

Имя параметра	.req	.cnf
Argument	M	
Mode	M	
ValueList	U	
Result (+)		S
Result (–)		S
ErrorInfro		M

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

Mode (Режим)
Данный параметр указывает требуемый режим DL Ведущего узла в отдельном порте.

Допустимые значения:
INACTIVE (обработчик изменяет состояние на INACTIVE);
STARTUP (обработчик изменяет состояние на STARTUP);
PREOPERATE (обработчик изменяет состояние на PREOPERATE);
OPERATE (обработчик изменяет состояние на OPERATE).

ValueList (СписокЗначений)
Данный список содержит используемые рабочие параметры.

Структура данных: Запись.

M-SequenceTime (ДлительностьМ-последовательности): (для передачи обработчику сообщений).
M-SequenceType (ТипМ-последовательности): (для передачи обработчику сообщений).

Допустимые значения:
TYPE_0;
TYPE_1_1, TYPE_1_2, TYPE_1_V;
TYPE_2_1, TYPE_2_2, TYPE_2_3, TYPE_2_4, TYPE_2_5, TYPE_2_6, TYPE_2_V;
(TYPE_1_1 задает режим расслоения при передаче PD и OD, см. 7.3.4.2).

PDInputLength (ДлинаВходныхДП): (для передачи обработчику сообщений).
PDOutputLength (ДлинаВыходныхДП): (для передачи обработчику сообщений).
OnReqDataLengthPerMessage (ДлинаДЗНаСообщение): (для передачи обработчику сообщений).

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неуспешно.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
STATE_CONFLICT (в текущем состоянии услуга недоступна);
PARAMETER_CONFLICT (нарушена корректность набора параметров).

7.2.1.14 Услуга DL_Mode

AL применяет услугу DL_Mode для уведомления SM о достижении определенного рабочего состояния. Параметры сервисных примитивов приведены в таблице 27.

Таблица 27 — DL_Mode

Имя параметра	.ind
Argument	M
RealMode	M

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

RealMode (РеальныйРежим)
Данный параметр указывает на текущее состояние обработчика DL.

Допустимые значения:
INACTIVE (обработчик изменяет состояние INACTIVE);
COM1 (установлен режим COM1);
COM2 (установлен режим COM2);
COM3 (установлен режим COM3);
COMLOST (связь потеряна);
ESTABCOM (обработчик изменил состояние на EstablishCom);
STARTUP (обработчик изменил состояние на STARTUP);
PREOPERATE (обработчик изменил состояние на PREOPERATE);
OPERATE (обработчик изменил состояние на OPERATE).

7.2.1.15 Услуга DL_Event

Услуга DL_Event показывает состояние ожидания или информацию об ошибке. Причина События расположена в Устройстве, и приложение Устройства запускает передачу События. Параметры сервисных примитивов приведены в таблице 28.

Таблица 28 — DL_Event

Имя параметра	.req	.ind
Argument	M	M
Instance	M	M
Type	M	M
Mode	M	M
EventCode	M	M
EventsLeft		M

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

Instance (Экземпляр)
Данный параметр указывает источник События.

Допустимые значения: Приложение (см. таблицу А.17).

Type (Тип)
Данный параметр указывает категорию События.

Допустимые значения ERROR, WARNING, NOTIFICATION (см. таблицу А.19).

Mode (Режим)
Данный параметр указывает режим События.

Допустимые значения SINGLESHOT, APPEARS, DISAPPEARS (см. таблицу А.20).

EventCode (КодСобытия)
Данный параметр содержит код, идентифицирующий определенное Событие (см. таблицу D.1).

Тип параметра: 16-битовое целое без знака.

EventsLeft (ОсталосьСобытий)
Данный параметр указывает число необработанных Событий.

7.2.1.16 Услуга DL_EventConf

Услуга DL_EventConf подтверждает переданные События через обработчик Событий. Данная услуга не имеет параметров. Сервисные примитивы приведены в таблице 29.

Таблица 29 — Услуга DL_EventConf

Имя параметра	.req	.cnf
<отсутствуют>		

7.2.1.17 Услуга DL_EventTrigger

Запрос DL_EventTrigger начинает сигнализацию События [см. Event flag (флаг События) на рисунке А.3] и «замораживает» память События на DL. Подтверждение возвращается после того, как

обработаны активированные События. Дополнительные запросы услуги DL_EventTrigger игнорируются до подтверждения предыдущего события (см. 7.3.8, 8.3.3 и рисунок 64). Данная услуга не имеет параметров. Сервисные примитивы приведены в таблице 30.

Таблица 30 — DL_EventTrigger

Имя параметра	.req	.cnf
<отсутствуют>		

7.2.1.18 Услуга DL_Control

Ведущий узел применяет услугу DL_Control для передачи управляющей информации через механизм команд Ведущего узла соответствующему приложению Устройства, зависящему от используемой технологии, и получает управляющую информацию через механизм флага состояния PD (см. A.1.5) и услугу PDInStatus (см. 7.2.2.5). Параметры сервисных примитивов приведены в таблице 31.

Таблица 31 — Услуга DL_Control

Имя параметра	.req	.cnf
Argument	M	M
ControlCode	M	M(=)

Argument (Аргумент)

Определяемые услугой данные передаются в данном параметре.

ControlCode (УправляющийКод)

Данный параметр указывает статус описателя PD.

Допустимые значения:
VALID (входные PD допустимы; см. 7.2.2.5, 8.2.2.12);
INVALID (входные PD недопустимы);
PDOUTVALID (выходные PD допустимы; см. 7.3.7.1);
PDOUTINVALID (выходные PD недопустимы или отсутствуют).

7.2.2 Услуги уровня DL-A

7.2.2.1 Обзор

В соответствии с подразделом 7.1 данные DL разделяются на верхний уровень DL–B и нижний уровень DL-A. Уровень DL-A включает обработчик сообщений, как показано на рисунках 26 и 27.

Обработчик событий Ведущего узла кодирует команды и данные в сообщения и посылает Данные сообщения в подключенное Устройство на PL. Он получает сообщения от Устройства на PL и переправляет их содержание соответствующим обработчикам в форме подтверждения. Если в событии Устройства установлен флаг События «Event flag» (см. A.1.5), обработчик сообщений Ведущего узла вызывает услугу EventFlag, чтобы проинструктировать обработчик Событий.

Обработчик сообщений Ведущего узла применяет стратегию повторных попыток для поврежденных сообщений, т. е. сообщений с неправильной контрольной суммой от Устройства или с отсутствующей контрольной суммой. В данных случаях Ведущий узел два раза повторяет сообщение ведущего узла (см. таблицу 97). Если повторные попытки также неуспешны, будет предоставлено отрицательное подтверждение и Ведущий узел будет повторно инициировать связь через обработчик порта x, начиная с пробуждения.

После фазы запуска обработчик сообщений выполняет циклическую операцию с M-последовательностью и временем цикла, предоставленным услугами DL_SetMode.

В таблице 32 приведены распределения Ведущего узла и Устройства по их ролям, как инициатора (I) или получателя (R) в контексте исполнения их конкретных услуг DL-A.

Таблица 32 — Услуги уровня DL-A на Ведущем узле и Устройстве

Имя услуги	Ведущий узел	Устройство
OD	R	I
PD	R	I
EventFlag	I	R
PDInStatus	I	R
MHInfo	I	I
ODTrig	I	
PDTrig	I	

7.2.2.2 Услуга OD

Услуга OD применяется для подготовки OD для следующего посылаемого сообщения. В свою очередь, подтверждение услуги содержит данные от приемника. Параметры сервисных примитивов приведены в таблице 33.

Таблица 33 — Услуга OD

Имя параметра	.req	.ind	.rsp	.cnf
Argument	M	M		
RWDirection	M	M		
ComChannel	M	M		
AddressCtrl	M	M		
Length	M	M		
Data	C	C		
Result (+)			S	S
Data			C	C(=)
Length			M	M
Result (–)			S	S
ErrorInfo			M	M(=)

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

RWDirection (НаправлениеЧЗ)
Данный параметр указывает на направление передачи: чтение или запись.

Допустимые значения:
READ (операция Чтения);
WRITE (операция Записи).

ComChannel (КаналСвязи)
Данный параметр указывает выбранный канал связи для передачи.

Допустимые значения: DIAGNOSIS, PAGE, ISDU (см. таблицу А.1).

AddressCtrll (АдресКонтроль)
Данный параметр содержит адрес или значение управления потоком (см. А.1.2).
Допустимые значения от 0 до 31.

Length (Длина)
Данный параметр содержит длину передаваемых данных.

Допустимые значения от 0 до 32.

Data (Данные)
Данный параметр содержит передаваемые данные.

Тип данных: Строка октетов.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Data (Данные)
Данный параметр содержит значения прочитанных данных.

Length (Длина)
Данный параметр содержит длину полученного пакета данных.

Допустимые значения от 0 до 32.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неуспешно.

ErrorInfor (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
NO_COMM (нет доступной связи);
STATE_CONFLICT (в текущем состоянии услуга недоступна).

7.2.2.3 Услуга PD

Услуга PD применяется для подготовки PD, посылаемых по каналу связи процесса. Подтверждение услуги содержит данные от приемника. Параметры сервисных примитивов приведены в таблице 34.

Таблица 34 — Услуга PD

Имя параметра	.req	.ind	.rsp	.cnf
Argument	M	M		
PDInAddress	C	C(=)		
PDInLength	C	C(=)		
PDOut	C	C(=)		
PDOutAddress	C	C(=)		
PDOutLength	C	C(=)		
Result (+)			S	S
PDIn			C	C(=)
Result (–)			S	S
ErrorInfo			M	M(=)

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

PDInAddress (ВхДанныеПроцессаАдрес)
Данный параметр содержит адрес затребованных входных PD (см. 7.3.4.2).

PDInLength (ВхДанныеПроцессаДлина)
Данный параметр содержит адрес затребованных входных PD.

Допустимые значения от 0 до 32.

PDOut (ВыхДанныеПроцесса)
Данный параметр содержит PD, подлежащие передаче из Ведущего узла к Устройству.

Тип данных: Строка октетов.

PDOutAddress (ВыхДанныеПроцессаАдрес)
Данный параметр содержит адрес переданных выходных PD (см. 7.3.4.2).

PDOutLength (ВыхДанныеПроцессаДлина)
Данный параметр содержит длину переданных выходных PD.

Допустимые значения от 0 до 32.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

PDIn (ВхДанныеПакета)
Данный параметр содержит PD, подлежащие передаче из Устройства к Ведущему узлу.

Тип данных: Строка октетов.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неуспешно.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
NO_COMM (нет доступной связи);
STATE_CONFLICT (в текущем состоянии услуга недоступна).

7.2.2.4 Услуга EventFlag

Услуга EventFlag устанавливает или сообщает состояние флага События «Event flag» (см. A.1.5) во время циклической связи. Параметры сервисных примитивов приведены в таблице 35.

Таблица 35 — Услуга EventFlag

Имя параметра	.req	.ind
Argument Flag	M	M

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

Flag (Флаг)
Данный параметр содержит значение флага События «Event flag».

Допустимые значения:
TRUE («Event flag» = 1);
FALSE («Event flag» = 0).

7.2.2.5 Услуга PDInStatus

Услуга PDInStatus устанавливает и сообщает значение допустимости входных PD. Параметры сервисных примитивов приведены в таблице 36.

Таблица 36 — PDInStatus

Имя параметра	.req	.ind
Argument Status	M	M

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

Status (Состояние)
Данный параметр содержит индикатор допустимости переданных входных PD.

Допустимые значения:
VALID [входные PD допустимы на основе флага состояния PD (см. A.1.5); см. 7.2.1.18];
INVALID (входные PD недопустимы).

7.2.2.6 Услуга MHInfo

Услуга MHInfo сообщает об исключительных операциях в обработчике сообщений. Параметры сервисных примитивов приведены в таблице 37.

Таблица 37 — Услуга MHInfo

Имя параметра	.ind
Argument MHInfo	M

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

MHInfo (ИнформацияОбработчикаСообщений)
Данный параметр содержит указания об исключительных ситуациях в обработчике сообщений.

Допустимые значения:
COMLOST (связь потеряна);
ILLEGAL_MESSAGETYPE (обнаружена M-последовательность неожиданного типа);
CHECKSUM_MISMATCH (обнаружена ошибка контрольной суммы).

7.2.2.7 Услуга ODTrig

Услуга ODTrig доступна только на Ведущем узле. Услуга запускает обработчик OD, и затем обработчики ISDU, Команд и Событий несут ответственность за предоставление OD (через услугу OD) для следующего сообщения Ведущего узла. Параметры сервисных примитивов приведены в таблице 38.

Таблица 38 — Услуга ODTrig

Имя параметра	.ind
Argument DataLength	M

Argument (Аргумент)
Определяемые услугой данные передаются в данном параметре.

DataLength (ДлинаДанных)
Данный параметр содержит доступное пространство для OD на сообщение.

7.2.2.8 Услуга PDTrig

Услуга PDTrig доступна только на Ведущем узле. Услуга запускает обработчик PD для формирования PD для следующего сообщения Ведущего узла. Параметры сервисных примитивов приведены в таблице 39.

Таблица 39 — Услуга PDTrig

Имя параметра	.ind
Argument DataLength	M M

Argument (Аргумент)
Определяемые услугой данные передаются в аргументе.

DataLength (ДлинаДанных)
Данный параметр содержит доступное пространство для PD на сообщение.

7.3 Протокол канального уровня

7.3.1 Обзор

На рисунках 26 и 27 показаны структура DL и его компоненты; обработчик режима DL, обработчик сообщений, обработчик PD и обработчик OD для предоставления определенных услуг. В 7.3.2—7.3.8 устанавливается поведение (динамика Данных обработчиков) средствами конечных машин языка UML и таблиц переходов.

Обработчик OD поддерживает три независимых типа данных: ISDU, команда и Событие. Поэтому три дополнительные конечные машины работают вместе с конечной машиной обработчика OD, как показано на рисунке 28. Дополнительная последовательность или диаграммы деятельности демонстрируют некоторые сценарии использования. См. IEC/TR 62390 и ИСО/МЭК 19505.

Элементы, с которыми работает каждый обработчик, такие как сообщения, процедуры пробуждения, режим расслоения, ISDU (индексированные сервисные блоки данных) и События, определяются в контексте соответствующего обработчика.



Рисунок 28 — Конечные машины канального уровня

7.3.2 Обработчик канального уровня

7.3.2.1 Общие положения

Обработчик DL, показанный на рисунке 26, несет ответственность за установку связи SDCI, применяя услуги PL и внутренние административные вызовы, и контролирует обработчик сообщений, а также состояния других обработчиков.

Обработчик DL, показанный на рисунке 27, несет ответственность за обнаружение запроса пробуждения и установление связи. Он получает команды Ведущего узла для синхронизации с состояниями STARTUP, PREOPERATE и OPERATE обработчика DL Ведущего узла и управляет активацией и деактивацией обработчиков сообразно обстоятельствам.

7.3.2.2 Процедуры пробуждения и правила соответствия Устройства

Управление системой запускает следующие действия на DL с помощью услуги DL_SetMode (затребованный режим = STARTUP).

Обработчик DL Ведущего узла пытается установить связь через запрос пробуждения (PL_WakeUp. req) с М-последовательностью TYPE_0 (чтение «MinCycleTime») в соответствии с последовательностью, показанной на рисунке 29.

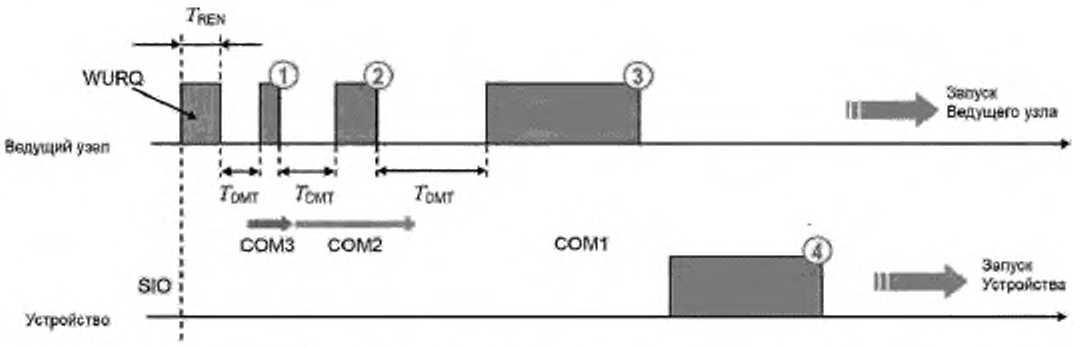


Рисунок 29 — Пример попытки установления связи

После запроса пробуждения (WURQ), определенного в 5.3.3.3, обработчик DL требует, чтобы обработчик сообщений послал первое тестовое сообщение через промежуток времени T_{REN} (см. таблицу 9) и T_{DMT} (см. таблицу 40). Установленные скорости передачи данных портов COM1, COM2 и COM3 применяются в нисходящем порядке, пока не будет получен ответ, как показано на рисунке 29:

Шаг (1): сообщение Ведущего узла со скоростью передачи данных порта COM3 (см. таблицу 8);

Шаг (2): сообщение Ведущего узла со скоростью передачи данных порта COM2 (см. таблицу 8);

Шаг (3): сообщение Ведущего узла со скоростью передачи данных порта COM1 (см. таблицу 8);

Шаг (4): ответное сообщение Устройства со скоростью передачи порта COM1.

Перед инициацией (нового) сообщения обработчик канального вывода ожидает по меньшей мере в течение периода времени T_{DMT}. Значение T_{DMT} устанавливается в таблице 40.

В отношении поддержки скоростей передачи данных для Устройства применяется следующее правило соответствия:

- Устройство будет поддерживать только одну из скоростей передачи для портов COM1, COM2 или COM3.

Если попытка установления связи не достигает успеха, обработчик DL Ведущего узла не будет начинать новую попытку процедуры пробуждения, пока не истечет период времени T_{DWU}, как показано на рисунке 30 и установлено в таблице 40.

Ведущий узел будет выдавать до n_{WU}+1 последовательных запросов пробуждения, как показано на рисунке 31. Если эта начальная последовательность повторных попыток пробуждения терпит неудачу, Устройство возвратит линию C/Q в режим SIO через промежуток времени T_{DSIO} (T_{DSIO} повторно запускается в Устройстве после каждого обнаруженного запроса пробуждения). Ведущий узел не будет запускать новую последовательность попыток пробуждения через промежуток времени T_{SD}.

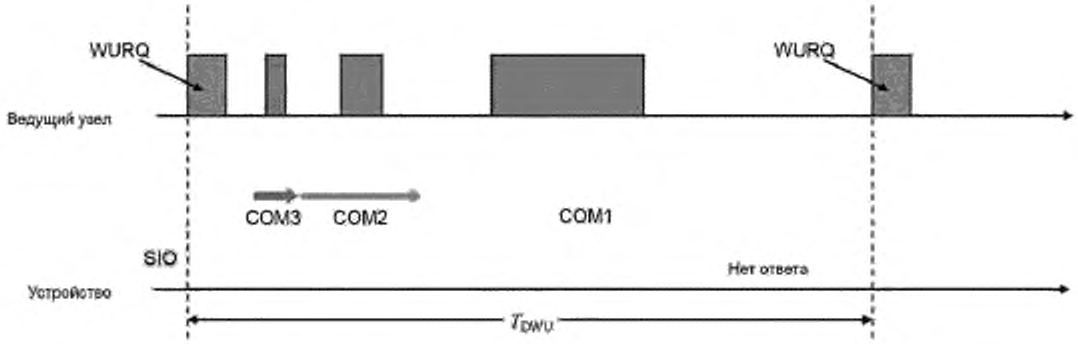


Рисунок 30 — Неудачная попытка установления связи

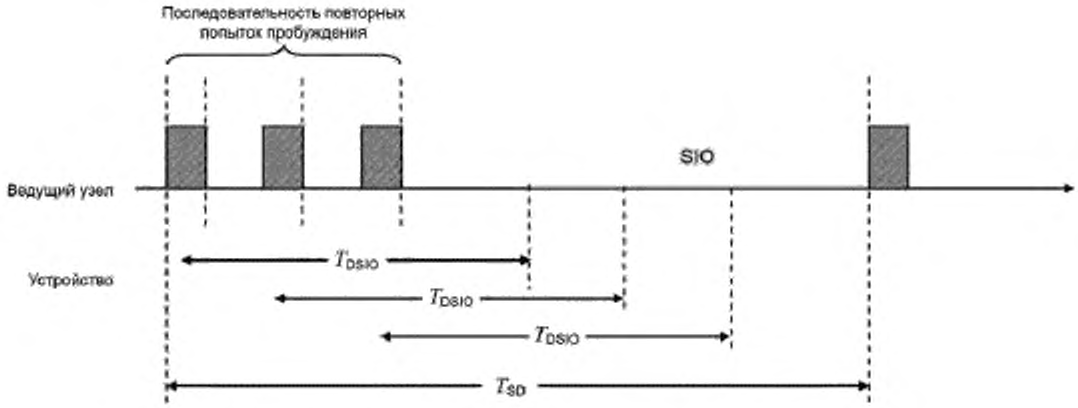


Рисунок 31 — Стратегия повторных попыток установления связи

DL Ведущего узла пошлет запрос PL на переход в режим SIO после неудачной последовательности повторных попыток пробуждения.

Значения для согласования по времени процедур пробуждения и повторных попыток установлены в таблицах 9 и 40. Они определяются с точки зрения Ведущего узла.

Таблица 40 — Процедура пробуждения и характеристики повторной попытки

Свойство	Наименование	Минимум	Норма	Максимум	Единица измерения	Примечание
T_{DMT}	Задержка сообщения Ведущего узла	27	н/д	37	T_{BIT}	Время прохождения бита повторной передачи данных
T_{DSIO}	Задержка SIO	60	н/д	300	мс	Через T_{DSIO} Устройство возвращается в режим SIO (если он поддерживается)
T_{DWU}	Задержка повторной попытки пробуждения	30	н/д	50	мс	Через T_{DWU} Ведущий узел повторяет запрос пробуждения

Окончание таблицы 40

Свойство	Наименование	Минимум	Норма	Максимум	Единица измерения	Примечание
n_{WU}	Счетчик повторных попыток пробуждения	2	2	2		Число повторных попыток запроса пробуждения
T_{SD}	Время обнаружения Устройства	0,5	н/д	1	с	Время между двумя последовательностями запроса пробуждения (см. примечание)
Примечание — Характеристика Ведущего узла.						

DL Ведущего узла прекратит процедуру установления связи, как только он найдет Устройство, поддерживающее связь, и сообщит в SM об обнаруженном режиме COMx, применяя услугу DL. Если процедура закончится неудачно, о соответствующей ошибке будет сообщено с использованием той же услуги.

7.3.2.3 Процедура возврата в исходный режим

SM вызывает следующие действия на DL с помощью услуги DL_SetMode (режим = INACTIVE):

- команда Ведущего узла Fallback (Возврат в исходное состояние, см. таблицу B.2) заставляет Устройство перейти в режим SIO;

- Устройство завершит переход в режим SIO через три времени цикла Ведущего узла MasterCycleTimes и/или не позднее, чем через 500 мс после команды Ведущего узла. Это делает возможными повторные попытки, если команда Ведущего узла закончилась неудачно из-за отрицательного ответа Устройства.

На рисунке 32 показаны процедура возврата в исходный режим, ее повторные попытки и временные ограничения.

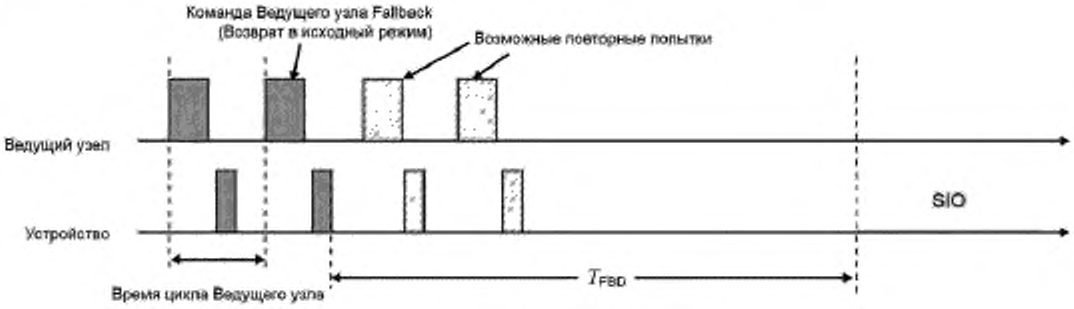


Рисунок 32 — Процедура возврата в исходный режим

В таблице 41 устанавливаются временные характеристики возврата в исходный режим. Подробное описание приводится в А.2.6.

Таблица 41 — Временные характеристики возврата в исходный режим

Свойство	Наименование	Минимум	Норма	Максимум	Единица измерения	Примечание
T_{FBD}	Задержка возврата в исходный режим	Три времени цикла Ведущего узла (OPERATE) или 3 Tinitcyc (PREOPE-RATE)	н/д	500	мс	Через время T_{FBD} Устройство будет переключено в режим SIO (см. рисунок 32)

7.3.2.4 Конечная машина обработчика DL Ведущего узла

На рисунке 32 показана конечная машина обработчика DL Ведущего узла.

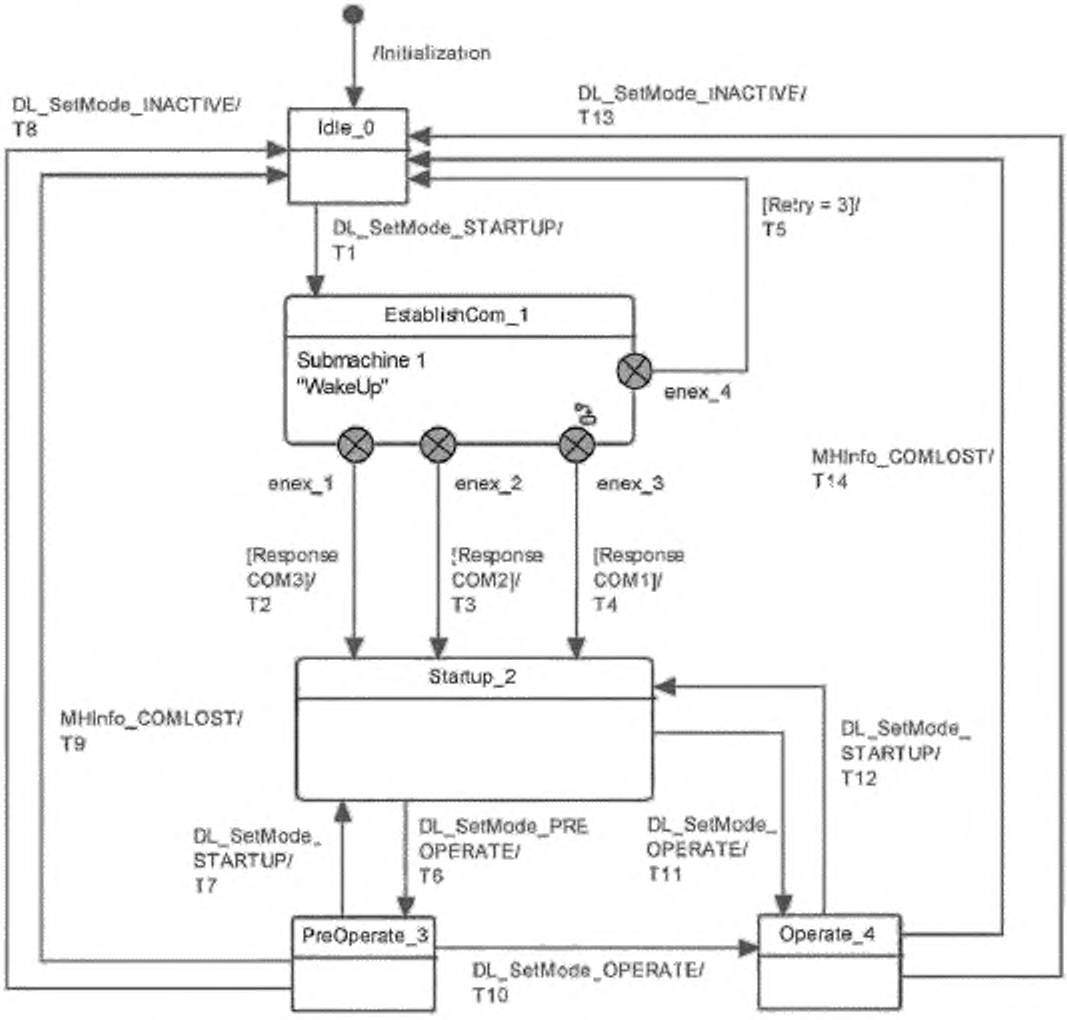


Рисунок 33 — Конечная машина обработчика канального уровня Ведущего узла

Примечание — Условные обозначения диаграмм языка UML определены в 3.3.7.

После получения услуги DL_SetMode_STARTUP из SM обработчик DL вначале создаст импульс тока пробуждения через услугу PL_WakeUp и затем установит связь. Данная процедура определена в функциональном узле 1 на рисунке 34.

Цель состояния «Startup_2» — это проверка идентичности Устройства через данные страницы Непосредственных параметров (см. рисунок 5). В состоянии «PreOperate_3» Ведущий узел назначает параметры Устройству, применяя ISDU. Циклический обмен PD осуществляется в состоянии «Operate». В данном состоянии дополнительные OD, такие как ISDU, команды и События могут передаваться, применяя подходящие типы M-последовательностей (см. рисунок 37).

В состояниях «PreOperate_3» и «Operate_4» в Ведущем узле активируются различные наборы обработчиков.

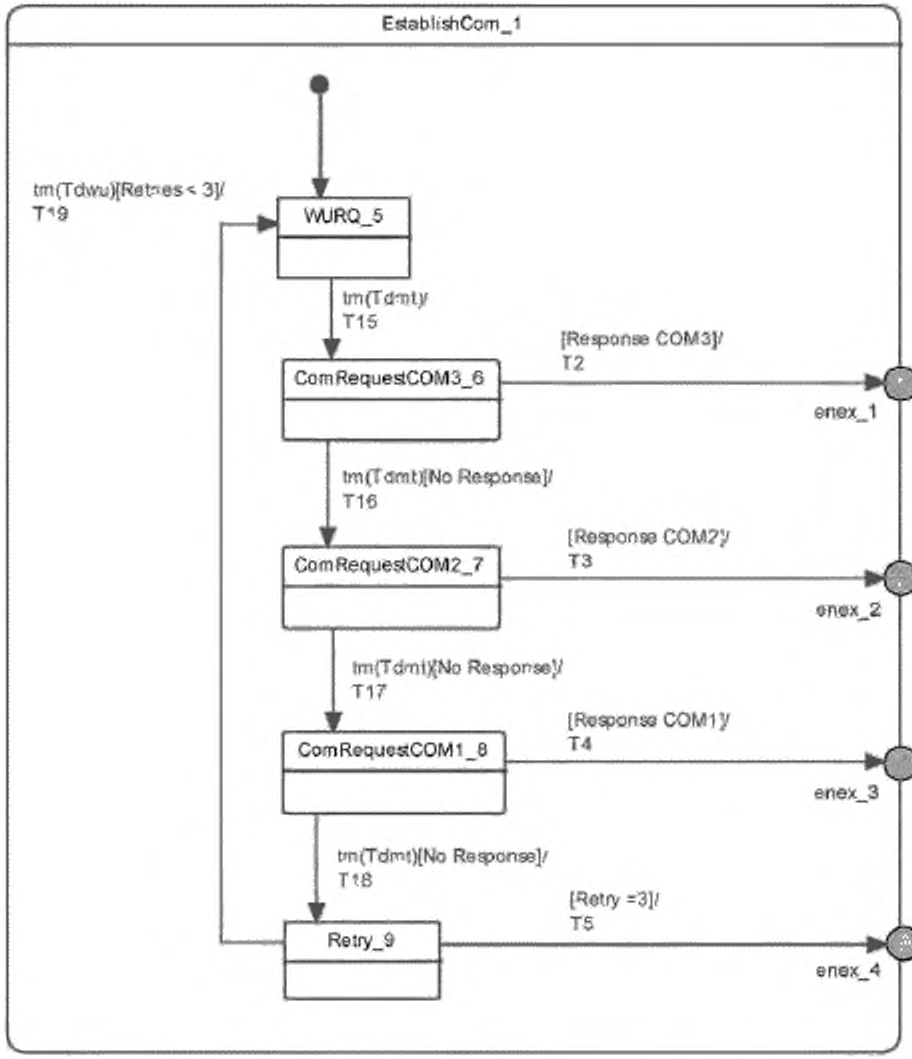


Рисунок 34 — Функциональный узел 1 для установки связи

В таблице 42 показаны таблицы перехода состояний обработчика DL на Ведущем узле.

Таблица 42 — Таблицы перехода состояний обработчика DL на Ведущем узле

Имя состояния	Описание состояния
Idle_0	Ожидание запроса пробуждения от SM: DL_SetMode (STARTUP)
EstablishComm_1	Выполнить процедуру пробуждения (функциональный узел 1)
Startup_2	SM применяет состояние STARTUP для идентификации Устройства, проверки и конфигурирования связи (см. рисунок 69)
Preoperate_3	ODE (параметры, команды, События) без PD
Operate_4	PD и OD (параметры, команды, События)

Продолжение таблицы 42

Имя состояния		Описание состояния	
SM: WURQ_5		Создать импульс тока пробуждения: Вызвать услугу PL_WakeUp (см. рисунок 11 и 5.3.3.3) и ожидать T_{DMT} (см. таблицу 40)	
SM: ComRequestCOM3_6		Послать тестовое сообщение со скоростью передачи порта COM3 через обработчик сообщений: Вызвать услугу MH_Conf_COMx (см. рисунок 38) и ожидать T_{DMT} (см. таблицу 40)	
SM: ComRequestCOM2_7		Послать тестовое сообщение со скоростью передачи порта COM2 через обработчик сообщений: Вызвать MH_Conf_COMx (см. рисунок 38) и ожидать T_{DMT} (см. таблицу 40)	
SM: ComRequestCOM1_8		Послать тестовое сообщение со скоростью передачи порта COM1 через обработчик сообщений: Вызвать MH_Conf_COMx (см. рисунок 38) и ожидать T_{DMT} (см. таблицу 40)	
SM: Retry_9		Проверить число повторных попыток	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	Обнулить число попыток: Retry = 0
T2	1	2	Скорость передачи порта COM3 проверена успешно. Обработчик сообщений активирован и сконфигурирован на COM3 (см. рисунок 38, Переход T2). Активировать обработчик команд (вызов CH_Conf_ACTIVE на рисунке 51). Вернуть Return DL_Mode.ind (STARTUP) и DL_Mode.ind (COM3) в SM
T3	1	2	Скорость передачи порта COM2 проверена успешно. Обработчик сообщений активирован и сконфигурирован на COM2 (см. рисунок 38, Переход T2). Активировать обработчик команд (вызов CH_Conf_ACTIVE на рисунке 51). Вернуть Return DL_Mode.ind (STARTUP) и DL_Mode.ind (COM2) в SM
T4	1	2	Скорость передачи порта COM1 проверена успешно. Обработчик сообщений активироеан и сконфигурирован на COM1 (см. рисунок 38, Переход T2). Активировать обработчик команд (вызов CH_Conf_ACTIVE на рисунке 51). Вернуть DL_Mode.ind (STARTUP) и DL_Mode.ind (COM1) в SM
T5	1	0	Вернуть DL_Mode.ind (INACTIVE) в SM
T6	2	3	SM затребовал режим OPERATE. Активировать обработчик OD (вызов OH_Conf_ACTIVE на рисунке 46), обработчик ISDU (вызов IH_Conf_ACTIVE на рисунке 49) и обработчик Событий (вызов EH_Conf_ACTIVE на рисунке 53). Изменить состояние обработчика сообщений на PREOPERATE (вызов MH_Conf_MPERATE на рисунке 38). Вернуть DL_Mode.ind (PREOPERATE) в SM
T7	3	2	SM затребовал режим STARTUP. Изменить состояние обработчика сообщений на STARTUP (вызов MH_Conf_STARTUP на рисунке 38). Деактивировать обработчик OD (вызов OH_Conf_INACTIVE на рисунке 46), обработчик ISDU (вызов IH_Conf_INACTIVE на рисунке 49) и обработчик Событий (вызов EH_Conf_INACTIVE на рисунке 53). Вернуть DL_Mode.ind (STARTUP) в SM
T8	3	0	SM затребовал режим SIO. Деактивировать все обработчики (вызов xx_Conf_INACTIVE). Вернуть DL_Mode.ind (INACTIVE) в SM
T9	3	0	Обработчик сообщений информирует о потерянной связи через услугу DL-A MHInfo (COMLOST). Деактивировать все обработчики (вызов xx_Conf_INACTIVE). Вернуть DL_Mode.ind (COMLOST) в SM
T10	3	4	SM затребовал режим OPERATE. Активировать обработчик PD (вызов PD_Conf_SINGLE, если тип M-последовательности = TYPE_2_x, или вызов PD_Conf_INTERLEAVE, если тип M-последовательности = TYPE_1_1 на рисунке 44). Изменить состояние обработчика сообщений на OPERATE (вызов MH_Conf_OPERATE на рисунке 38). Вернуть DL_Mode.ind () в SM

Окончание таблицы 42

Переход	Исходное состояние	Конечное состояние	Действие
T11	2	4	SM затребовал режим OPERATE. Активировать обработчик PD (вызов PD_Conf_SINGLE или вызов PD_Conf_INTERLEAVE на рисунке 44 в соответствии с конфигурацией порта Ведущего устройства). Активировать обработчик OD (вызов OH_Conf_ACTIVE на рисунке 46), обработчик ISDU (вызов IH_Conf_ACTIVE на рисунке 49) и обработчик Событий (вызов EH_Conf_ACTIVE на рисунке 53). Изменить состояние обработчика сообщений на OPERATE (вызов MH_Conf_OPERATE на рисунке 38). Вернуть DL_Mode.ind () в SM
T12	4	2	SM затребовал режим STARTUP. Изменить состояние обработчика сообщений на STARTUP (вызов MH_Conf_STARTUP на рисунке 38). Деактивировать обработчики PD (вызов PD_Conf_INACTIVE на рисунке 44), OD (вызов OH_Conf_INACTIVE на рисунке 46), обработчик ИСДБ (вызов IH_Conf_INACTIVE на рисунке 49) и обработчик Событий (вызов EH_Conf_INACTIVE на рисунке 53). Вернуть DL_Mode.ind (STARTUP) в SM
T13	4	0	SM затребовал состояние SIO. Деактивировать все обработчики (вызов xx_Conf_INACTIVE). Вернуть DL_Mode.ind (INACTIVE) в SM
T14	4	0	Обработчик сообщений информирует о потерянной связи через услугу DL-A MHInfo (COMLOST). Деактивировать все обработчики (вызов xx_Conf_INACTIVE). Вернуть DL_Mode.ind (COMLOST) в SM
T15	5	6	Установить скорость передачи порта COM3
T16	6	7	Установить скорость передачи порта COM2
T17	7	8	Установить скорость передачи порта COM1
T18	8	9	Увеличить счетчик попыток Retry
T19	9	5	—
Внутренние элементы		Тип	Определение
MH_Conf_COMx		Вызов	Данный вызов заставляет обработчик сообщений послать сообщение с требуемой скоростью передачи порта COMx и M-последовательностью типа TYPE_0 (см. таблицу 44)
MH_Conf_STARTUP		Вызов	Данный вызов заставляет обработчик сообщения переключиться в состояние STARTUP (см. рисунок 38)
MH_Conf_ PREOPERATE		Вызов	Данный вызов заставляет обработчик сообщения переключиться в состояние PREOPERATE (см. рисунок 38)
MH_Conf_OPERATE		Вызов	Данный вызов заставляет обработчик сообщения переключиться в состояние OPERATE (см. рисунок 38)
xx_Conf_ACTIVE		Вызов	Данный вызов активирует соответствующий обработчик. xx заменяется на MH (обработчик сообщений), OH (обработчик OD), IH (обработчик ISDU), CH (обработчик команд) и/или EH (обработчик событий)
xx_Conf_I NACTIVE		Вызов	Данный вызов деактивирует соответствующий обработчик. xx заменяется на MH (обработчик сообщений), OH (обработчик OD), IH (обработчик ISDU), CH (обработчик команд) и/или EH (обработчик событий)
Retry		Переменная	Число повторных попыток установления связи

7.3.2.5 Конечная машина обработчика DL на Устройстве

На рисунке 35 показана конечная машина обработчика DL на Устройстве. В состояниях PreOperate_3 и Operate_4 в Устройстве активируются различные наборы обработчиков.

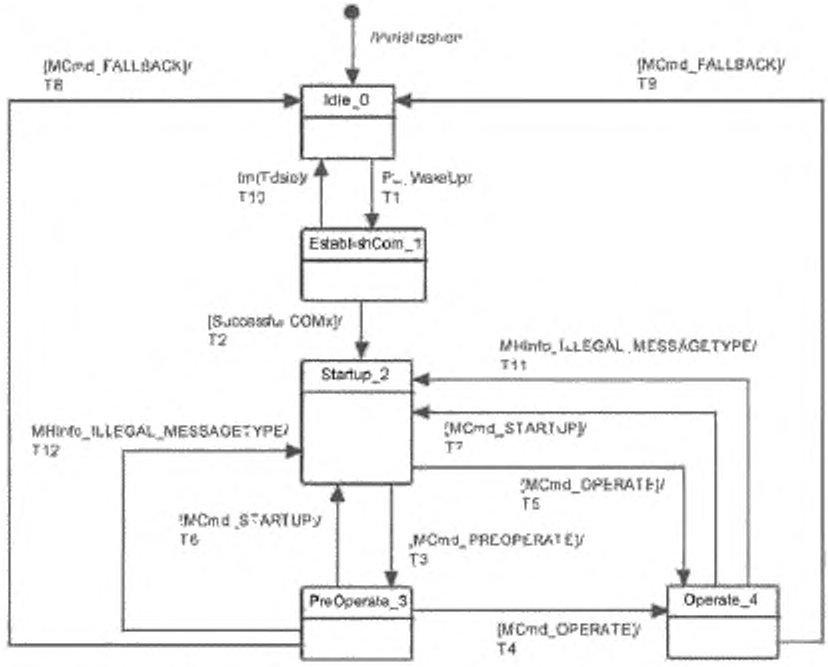


Рисунок 35 — Конечная машина обработчика канального уровня на Устройстве

Ведущий узел применяет команды Ведущего узла (см. таблицу 42) для изменения состояний SIO, STARTUP, PREOPERATE и OPERATE. При каждом обнаружении обработчиком сообщений неправильных (неожиданных) типов M-последовательности он будет заставлять обработчик DL менять состояние на STARTUP и указывать на данное состояние модулю управления системой (см. 9.3.3.2) в целях синхронизации Ведущего узла и Устройства.

В таблице 43 показаны таблицы перехода состояний обработчика DL на Устройстве.

Таблица 43 — Таблицы перехода состояний обработчика DL на Устройстве

Имя состояния		Описание состояния	
Idle_0		Ожидание обнаруженного импульса тока пробуждения (PL_WakeUp.ind)	
EstablishComm_1		Обработчик сообщений активирован и ожидает тестового сообщения COMx (см. таблицу 42)	
Startup_2		Проверка совместимости (см. 9.2.3.3)	
Preoperate_3		Обмен OD (параметры, команды, События) без PD	
Operate_4		Обмен PD и OD (параметры, команды, События)	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	Обнаружен импульс тока пробуждения. Активировать обработчик сообщений (вызов MH_Conf_ACTIVE на рисунке 42). Сообщить состояние через услугу DL_Mode.ind (ESTABCOM) в SM
T2	1	2	Установлена одна из трех скоростей передачи: для порта COM3, COM2 или COM1. Активировать обработчики OD (вызов OH_Conf_ACTIVE на рисунке 47) и команд (вызов CH_Conf_ACTIVE на рисунке 52). Сообщить состояние через услугу DL_Mode.ind (COM1, COM2 или COM3) в SM

Продолжение таблицы 43

Переход	Исходное состояние	Конечное состояние	Действие
T3	2	3	Обработчик команд Устройства получил команду (MCmd_PREOPERATE) Ведущего узла. Активировать обработчик ISDU (вызов IH_Conf_ACTIVE на рисунке 50) и обработчик Событий (вызов EH_Conf_ACTIVE на рисунке 54). Сообщить состояние через услугу DL_Mode.ind (PREOPERATE) в SM
T4	3	4	Обработчик команд Устройства получил команду (MCmd_OPERATE) Ведущего узла. Активировать обработчик PD (вызов MH_Conf_ACTIVE на рисунке 45). Сообщить состояние через услугу DL_Mode.ind (OPERATE) в SM
T5	2	4	Обработчик команд Устройства получил команду (MCmd_OPERATE) Ведущего узла. Активировать обработчик PD (вызов OH_Conf_ACTIVE на рисунке 45), обработчик ISDU (вызов IH_Conf_ACTIVE на рисунке 50) и обработчик Событий (вызов EH_Conf_ACTIVE на рисунке 54). Сообщить состояние через услугу DL_Mode.ind (OPERATE) в SM
T6	3	2	Обработчик команд Устройства получил команду (MCmd_STARTUP) Ведущего узла. Деактивировать обработчик ISDU (вызов IH_Conf_INACTIVE на рисунке 50) и обработчик Событий (вызов EH_Conf_INACTIVE на рисунке 54). Сообщить состояние через услугу DL_Mode.ind (STARTUP) в SM
T7	4	2	Обработчик команд Устройства получил команду (MCmd_STARTUP) Ведущего узла. Деактивировать обработчик PD (вызов OH_Conf_INACTIVE на рисунке 45), обработчик ISDU (вызов IH_Conf_INACTIVE на рисунке 50) и обработчик Событий (вызов EH_Conf_INACTIVE на рисунке 54). Сообщить состояние через услугу DL_Mode.ind (STARTUP) в SM
T8	3	0	Обработчик команд Устройства получил команду (MCmd_FALLBACK) Ведущего узла. Ждать окончания промежутка времени T_{FBD} и затем отключить все обработчики (вызов xx_Conf_INACTIVE). Сообщить состояние через услугу DL_Mode.ind (INACTIVE) в SM (см. рисунок 79 и таблицу 93)
T9	4	0	Обработчик команд Устройства получил команду (MCmd_FALLBACK) Ведущего узла. Ждать окончания промежутка времени T_{FBD} и затем отключить все обработчики (вызов xx_Conf_INACTIVE). Сообщить состояние через услугу DL_Mode.ind (INACTIVE) в SM (см. рисунок 79 и таблицу 93)
T10	1	0	После неудачных процедур пробуждения (см. рисунок 30) Устройство устанавливает сконфигурированный режим SIO после прохождения периода времени T_{DSIO} (см. рисунок 31). Деактивировать все обработчики (вызов xx_Conf_INACTIVE). Сообщить состояние через услугу DL_Mode.ind (INACTIVE) в SM
T11	4	2	Обработчик сообщений обнаружил неправильный тип M-последовательности. Деактивировать обработчик PD (вызов OH_Conf_INACTIVE на рисунке 45), обработчик ISDU (вызов IH_Conf_INACTIVE на рисунке 50) и обработчик Событий (вызов EH_Conf_INACTIVE на рисунке 54). Сообщить состояние через услугу DL_Mode.ind (INACTIVE) в SM (см. рисунок 79 и таблицу 93)
T12	3	2	Обработчик сообщений обнаружил неправильный тип M-последовательности. Деактивировать обработчик ISDU (вызов IH_Conf_INACTIVE на рисунке 50) и обработчик Событий (вызов EH_Conf_INACTIVE на рисунке 54). Сообщить состояние через услугу DL_Mode.ind (INACTIVE) в SM (см. рисунок 79 и таблицу 93)

Окончание таблицы 43

Переход	Исходное состояние	Конечное состояние	Действие
T_{FBD}		Время	См. таблицу 41
TDSIO		Время	См. рисунок 31
MCmd_XXXXXXX		Вызов	Любая команда Ведущего узла, полученная обработчиком команд Устройства (см. таблицу 42 и рисунок 52, состояние «CommandHandler_2»)

7.3.3 Обработчик сообщений

7.3.3.1 Общие положения

Роль обработчика сообщений установлена в подразделе 7.1 и 7.2.2.1. Структура и типы М-сообщений и поведение (динамика) обработчика сообщений установлены в 7.3.3.

7.3.3.2 М-последовательности

Ведущий узел и Устройства обмениваются данными посредством последовательности сообщений (М-последовательности). М-последовательность включает сообщение от Ведущего узла с последующим сообщением от Устройства, как показано на рисунке 36. Каждое сообщение состоит из фреймов UART.

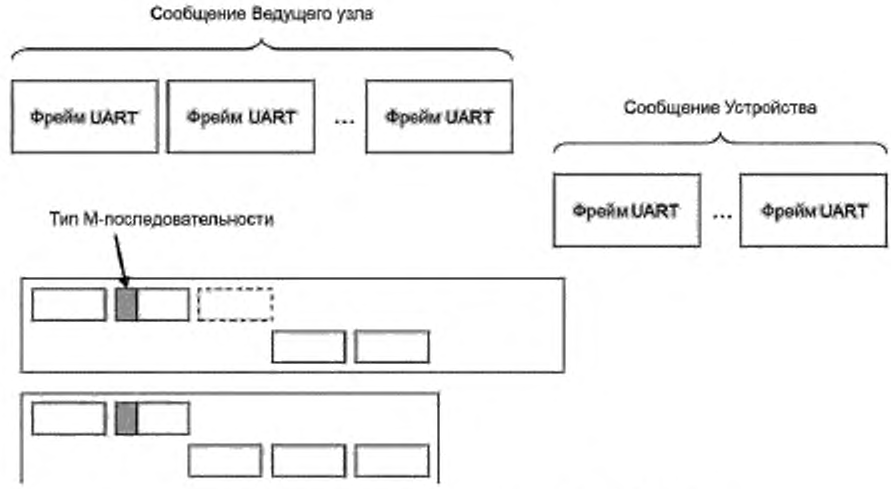


Рисунок 36 — Последовательности сообщений SDCI

Все многооктетные типы данных передаются как последовательности с обратным порядком, т. е. самый старший октет (MSO) посылается первым, за ним следуют менее значимые октеты в порядке убывания, и самый младший октет (LSO) посылается последним, как показано на рисунке 2.

Сообщение Ведущего узла начинается с октета «Управления М-последовательностью» (MC), далее следуют октет и необязательные октеты PD. Сообщение Устройства, в свою очередь, начинается с необязательных октетов PD, OD, за которыми следует октет «CHECK/STAT» (CKS).

Для удовлетворения конкретных требований исполнительного устройства или датчика (скорость сканирования, объем PD) могут выбираться различные типы М-последовательности. Длина сообщений Ведущего узла и Устройства может меняться в зависимости от типа сообщений и направления передачи данных (см. рисунок 36).

На рисунке 37 представлен обзор установленных типов М-последовательности. Части, отмеченные пунктиром, зависят от направления операции (чтение или запись) в управляющем октете М-последовательности.

Фиксированные типы М-последовательности состоят из TYPE_0, TYPE_1_1, TYPE_1_2 и от TYPE_2_1 до TYPE_2_6. Переменные типы М-последовательности состоят из TYPE_1_V и TYPE_2_V.

Различные типы М-последовательностей удовлетворяют различным требованиям датчиков и исполнительных устройств относительно ширины их PD и соответствующих условий. Подробное описание типов М-последовательностей приводится в разделе А.2. В разделе А.3 описываются временные ограничения М-последовательностей.

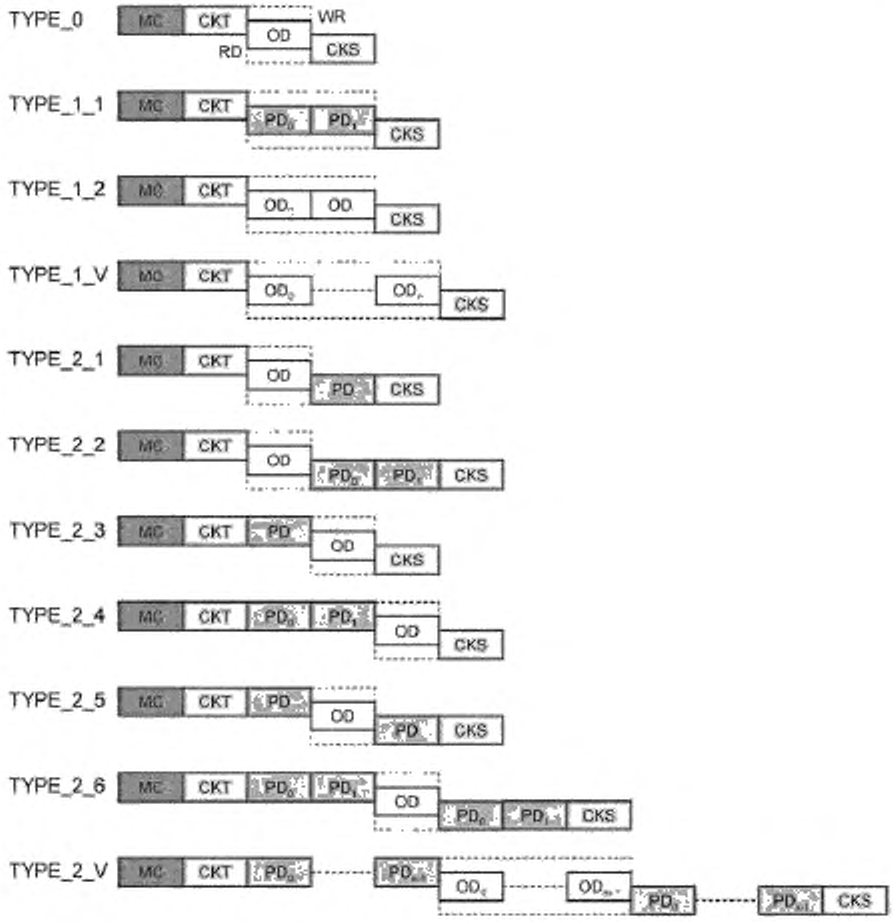


Рисунок 37 — Обзор типов М-последовательностей

7.3.3.3 Ограничения времени цикла Ведущего узла

В состояниях STARTUP и PREOPERATE устройство может осуществлять связь ациклически. Для обнаружения отключения Устройства рекомендуется, чтобы Ведущий узел поддерживал периодическую связь (сообщения контроля активности) через ациклические М-последовательности на DL. Минимальное время восстановления для ациклической связи, установленное в А.2.6, будет рассмотрено ниже.

После трех фаз циклическая связь PD может начинаться Ведущим узлом через услугу DL_SetMode (OPERATE). Типы М-последовательности для циклического обмена данными будут применяться на данной фазе связи для обмена PD и OD с Устройством (см. таблицы А.9 и А.10).

Ведущий узел будет применять для времени t_{CYC} значение, указанное в параметре MasterCycleTime (см. таблицу В.1) с относительным допуском от 0 % до плюс 10 % (включая флуктуации синхронизации).

Если Устройство должно переключаться обратно в режим SIO после параметризации, Ведущий узел будет посылать команду Fallback возврата в исходный режим (см. таблицу В.2), за которой следует подтверждение от Устройства.

7.3.3.4 Конечная машина обработчика сообщений Ведущего узла

На рисунке 38 показана конечная машина обработчика DL Ведущего узла. Три функциональных узла, описывающих реакции на ошибки связи, показаны на рисунках 39, 40 и 41.

Обработчик сообщений следит за соблюдением специальных требований к связи в состояниях «EstablishCom», «Startup», «PreOperate» и «Operate» обработчика DL.

Внутренний административный вызов MH_Conf_COMx в состоянии «Inactive_0» заставляет обработчик сообщений посылать тестовое сообщение с М-последовательностью типа TYPE_0 и скоростями передачи различных портов COM3, COM2 и COM1 во время последовательности установления связи.

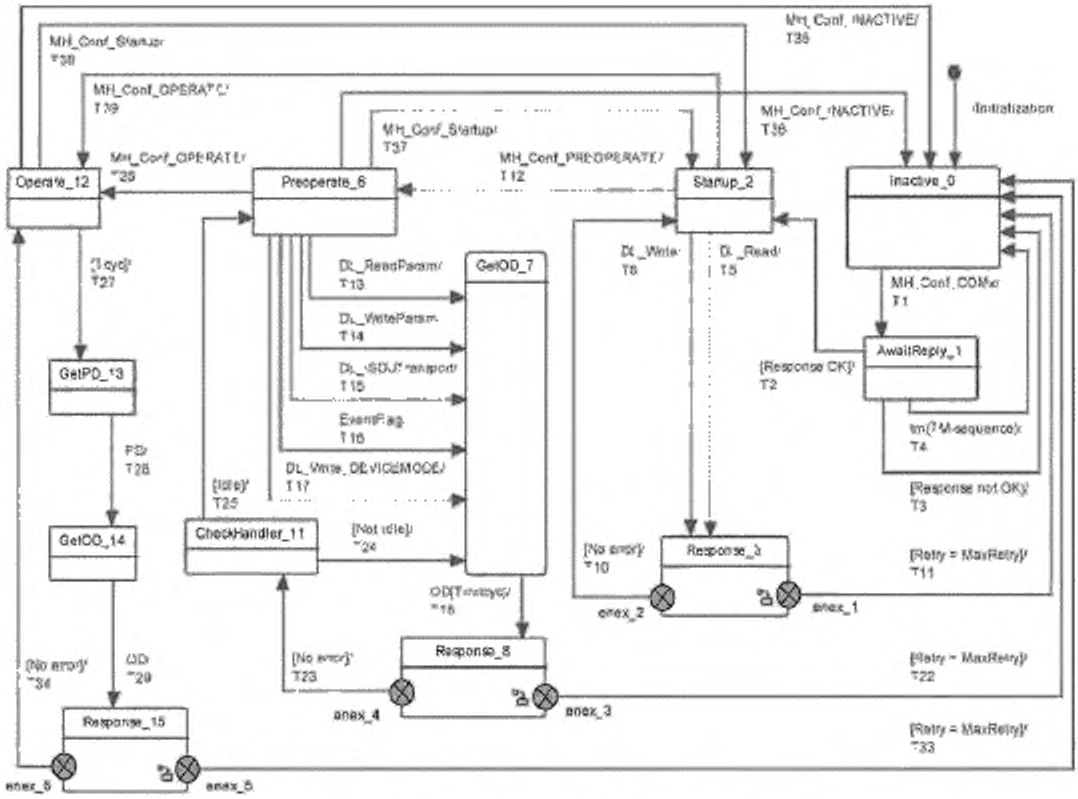


Рисунок 38 — Конечная машина обработчика сообщений Ведущего узла

В состоянии «Startup_2» предоставляются все средства связи для поддержки проверки идентичности SM с помощью услуг DL_Read и DL_Write. Обработчик сообщений ожидает появления данных услуг для посылки и получения сообщений (ациклическая связь).

Состояние «Preoperate_6» служит контрольной точкой для всех действий с OD, таких как ISDU, команды и События, с целью поддержки параметризации Устройства. Обработчик сообщений ожидает появления данных услуг, показанных на рисунке 38, для посылки и получения сообщений (ациклическая связь).

Состояние «Operate_12» является контрольной точкой для обмена циклическими PD. В зависимости от типа М-последовательности обработчик сообщений генерирует сообщения Ведущего узла с PD, полученными от обработчика прерываний PD через услугу PD, и при необходимости с OD, полученными от обработчика OD через услугу OD.

На рисунке 39 показан функциональный блок состояния «Response 3».

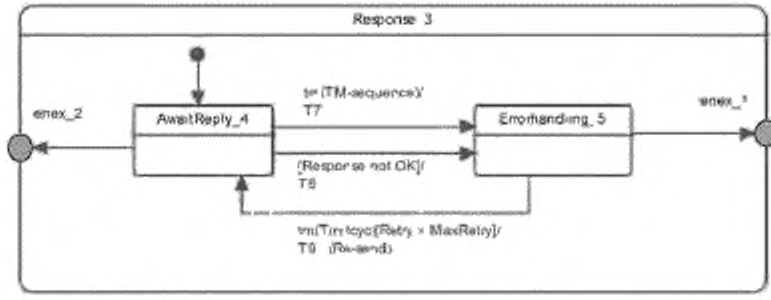


Рисунок 39 — Функциональный блок «Response 3» обработчика прерываний

На рисунке 40 показан функциональный блок состояния «Response 8».

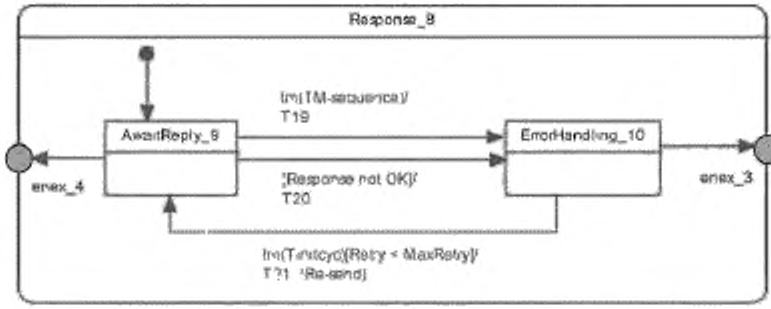


Рисунок 40 — Функциональный блок «Response 8» обработчика прерываний

На рисунке 41 показан функциональный блок состояния «Response 15».

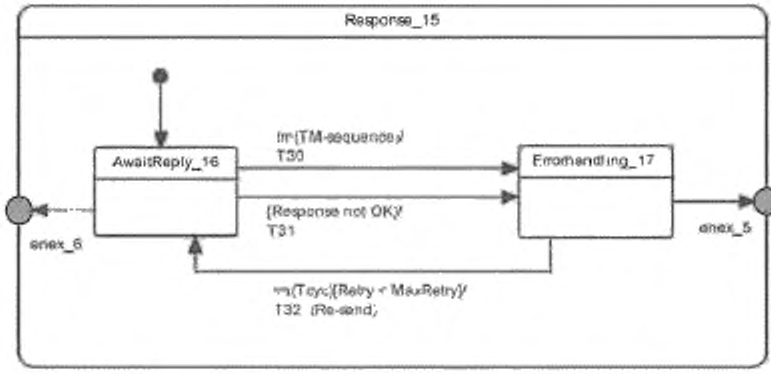


Рисунок 41 — Функциональный блок «Response 15» обработчика прерываний

В таблице 44 показаны таблицы перехода состояний обработчика DL на Ведущем узле.

Таблица 44 — Таблицы перехода состояний обработчика сообщений на Ведущем узле

Имя состояния	Описание состояния
Inactive_0	Ожидание по требованию тестового сообщения через вызов MH_Conf_COMx (см. рисунок 34 и таблицу 42) от обработчика DL

Продолжение таблицы 44

Имя состояния	Описание состояния
AwaitReply_1	Ожидание ответа Устройства на тестовое сообщение. Возврат в состояние Inactive_0 по истечении времени $T_{M\text{-}sequence}$ без ответа от Устройства или когда ответ на тестовое сообщение не может быть дешифрован. В случае корректного ответа от Устройства обработчик сообщений изменяет состояние на Startup_2
Startup_2	При входе через переход T2 это состояние отвечает за управление обменом ациклическими OD в соответствии с условиями, установленными в таблице A.7. Любая услуга DL_Write или DL_Read из SM вызывает такой переход
Response_3	Услуга OD заставила обработчик сообщений послать соответствующее сообщение. Функциональный блок в этом псевдосостоянии ожидает ответа и проверяет его правильность
SM: AwaitReply_4	Данное состояние следит за истечением промежутка времени $T_{M\text{-}sequence}$ и правильностью ответа
SM: ErrorHandling_5	В случае неправильного ответа обработчик сообщений повторно посылает сообщение через промежуток времени $T_{initcyc}$. После слишком большого числа повторных попыток обработчик сообщений перейдет в положение Inactive_0
Preoperate_6	При получении вызова MH_Conf_PREOPERATE обработчик сообщений переходит в данное состояние. Теперь обработчик сообщений несет ответственность за управление ациклическим обменом OD в соответствии с условиями, определенными в таблице A.8. Любая из услуг DL_ReadParam, DL_WriteParam, DL_ISDUTransport, DL_Write или EventFlag вызывает переход
GetOD_7	Обработчик сообщений использовал услугу ODTrig для получения OD из обработчика OD. Обработчик сообщений ожидает, когда услуга пошлет сообщения через промежуток времени $T_{initcyc}$
Response_8	Услуга OD заставила обработчик сообщений послать соответствующее сообщение. Функциональный блок в данном псевдосостоянии ожидает ответа и проверяет его правильность
SM: AwaitReply_9	Данное состояние следит за истечением промежутка времени $T_{M\text{-}sequence}$ и правильностью ответа
SM: ErrorHandling_10	В случае неправильного ответа обработчик сообщений повторно посылает сообщение через промежуток времени $T_{initcyc}$. После слишком большого числа повторных попыток обработчик сообщений перейдет в положение Inactive_0
CheckHandler_11	Некоторые услуги требуют нескольких циклов получения OD при обмене OD. Всякий раз, когда задействованный обработчик OD, обработчик ISDU или обработчик События возвращается в состояние простоя, обработчик сообщений может выйти из цикла получения OD
Operate_12	При получении вызова MH_Conf_OPERATE обработчик сообщений переходит в данное состояние и через начальное время $T_{initcyc}$ он становится ответственным за контроль обмена циклическими PD и OD в соответствии с условиями, установленными в таблицах A.9 и A.10. Обработчик сообщений повторно начинает новый цикл сообщений по истечении промежутка времени t_{CYC}
GetPD_13	Обработчик сообщений использовал услугу PDTrig для получения PD из обработчика PD. Обработчик сообщений ожидает услуги PD и затем переходит в состояние GetOD_14
GetOD_14	Обработчик сообщений использовал услугу ODTrig для получения OD из обработчика OD. Обработчик сообщений ожидает услуги OD для комплектования уже полученных PD и посылки сообщения с уже полученными PD или OD
Response_15	Обработчик сообщений послал сообщение с полученными PD или OD. Функциональный блок в данном псевдосостоянии ожидает ответа и проверяет его правильность

Продолжение таблицы 44

Имя состояния		Описание состояния	
SM: AwaitReply_16		В данном состоянии проверяется истечение промежутка времени $T_{M\text{-}sequence}$ и правильность ответа	
SM: ErrorHandling_17		В случае неправильного ответа обработчик сообщений повторно посылает сообщение через время ожидания t_{CYC}. После слишком большого числа повторных попыток обработчик сообщений перейдет в положение Inactive_0	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	Посылает сообщение с требуемой скоростью передачи порта COMx и М-последовательностью типа TYPE_0: Считывает страницу 1 Непосредственных параметров, адрес 0x02 (MinCycleTime); компилирует команду Ведущего узла по адресу 0xA2 в М-последовательность (см. А.1.2). Запускает таймер с установленным значением $T_{M\text{-}sequenc}$
T2	1	2	Возвращает значение «MinCycleTime» через подтверждение услуги DL_Read
T3	1	0	Сбрасывает таймер ($T_{M\text{-}sequence}$)
T4	1	0	Сбрасывает таймер ($T_{M\text{-}sequence}$)
T5	2	3	Посылает сообщение, применяя установленную скорость передачи, канал связи страницы и опцию доступа чтения (см. А.1.2). Запускает таймер с установленным значением $T_{M\text{-}sequence}$
T6	2	3	Посылает сообщение, применяя установленную скорость передачи, канал связи страницы и опцию доступа записи (см. А.1.2). Запускает таймер с установленным значением $T_{M\text{-}sequence}$
T7	4	5	Сбрасывает таймер ($T_{M\text{-}sequence}$)
T8	4	5	Сбрасывает таймер ($T_{M\text{-}sequence}$)
T9	5	4	Повторно посылает сообщение через промежуток времени $T_{initcyc}$. Повторно запускает таймер с установленным значением $T_{M\text{-}sequence}$
T10	3	2	Возвращает подтверждение услуги DL_Read или DL_Write соответственно в SM
T11	3	0	Обработчик сообщений возвращает MH_Info (COMLOST) обработчику DL
T12	2	6	—
T13	6	7	Обработчик сообщений вызывает услугу ODTrig для обработчика Запроса (см. рисунок 46), который находится в состоянии «ISDU_1». В данном состоянии он заставляет обработчик ISDU предоставить услугу OD в соответствии с услугой DL_ReadParam (см. рисунок 49, Переход T13)
T14	6	7	Обработчик сообщений вызывает услугу ODTrig для обработчика Запроса (см. рисунок 46), который находится в состоянии «ISDU_1». В данном состоянии он заставляет обработчик ISDU предоставить услугу OD в соответствии с услугой DL_WriteParam (см. рисунок 49, Переход T13)
T15	6	7	Обработчик сообщений вызывает услугу ODTrig для обработчика Запроса (см. рисунок 46), который находится в состоянии «ISDU_1». В данном состоянии он заставляет обработчик ISDU предоставить услугу OD в соответствии с услугой DL_ISDUTransort (см. рисунок 49, Переход T2). Обработчику сообщений может потребоваться несколько циклов, пока обработчик ISDU возвратится в состояние «простоя»
T16	6	7	Обработчик сообщений вызывает услугу ODTrig для обработчика Запроса (см. рисунок 46), который находится в состоянии «Event_4». В данном состоянии он заставляет обработчик Событий предоставить услугу OD в соответствии с услугой EventFlag (см. рисунок 53, Переход T2). Обработчику сообщений может потребоваться несколько циклов, пока обработчик Событий возвратится в состояние «простоя»

Продолжение таблицы 44

Переход	Исходное состояние	Конечное состояние	Действие
T17	6	7	Обработчик сообщений вызывает услугу ODTrig для обработчика Запроса (см. рисунок 46), который находится в состоянии «ISDU_1». В данном состоянии он заставляет обработчик ISDU предоставить услугу OD в соответствии с услугой DL_Write (см. рисунок 49, Переход T13)
T18	7	8	Посылает сообщение после истечения времени восстановления $T_{initcyc}$, вызванного услугой OD.req. Запускает таймер с установленным значением $T_{M\text{-}sequence}$
T19	9	10	Сбрасывает таймер ($T_{M\text{-}sequence}$)
T20	9	10	Сбрасывает таймер ($T_{M\text{-}sequence}$)
T21	10	9	Повторно посылает сообщение через промежуток времени $T_{initcyc}$. Повторно запускает таймер с установленным значением $T_{M\text{-}sequence}$
T22	8	0	Обработчик сообщений изменяет состояние на Inactive_0 и возвращает MH_Info (COMLOST) обработчику DL
T23	8	11	—
T24	11	7	Получает OD через вызов услуги ODTrig к обработчику OD, который, в свою очередь, запускает текущий дежурный обработчик через вызов ISDU или вызов EventTrig
T25	11	6	Возвращает результат через сервисный примитив OD.cnf
T26	6	12	Обработчик сообщений изменяет состояние на Operate_12
T27	12	13	Запускает таймер с установленным значением t_{CYC}. Получает PD через вызов услуги PDTrig к обработчику PD (см. рисунок 44)
T28	13	14	Получает OD через вызов услуги ODTrig к обработчику OD (см. рисунок 46)
T29	14	15	PD и OD становятся готовыми через услугу PD.req из обработчика PD или через услугу OD.req обработчика OD. Обработчик сообщений посылает сообщение. Запускает таймер с установленным временем $T_{M\text{-}sequence}$
T30	16	17	Сбрасывает таймер (TM-sequence)
T31	16	17	Сбрасывает таймер (TM-sequence)
T32	17	16	Повторно посылает сообщение через промежуток времени t_{CYC}. Повторно запускает таймер с установленным значением TM-sequence
T33	15	0	Обработчик сообщений изменяет состояние на Inactive_0 и возвращает MH_Info (COMLOST) обработчику DL
T34	15	12	Ответное сообщение Устройства — правильное. Возвращает PD через услугу PD.cnf и через вызов PDTrig к обработчику PD (см. таблицу 46). Возвращает OD через услугу OD.cnf или через вызов ODTrig к обработчику OD, который перенаправляет их к соответствующему обработчику ISDU (см. таблицу 51), обработчику команд (см. таблицу 54) или обработчику Событий 57)
T35	12	0	Обработчик сообщений изменяет состояние на Inactive_0 и возвращает MH_Info (COMLOST) обработчику DL
T36	6	0	Обработчик сообщений изменяет состояние на Inactive_0 и возвращает MH_Info (COMLOST) обработчику DL
T37	6	2	—
T38	12	2	—
T39	2	12	—

Окончание таблицы 44

Внутренние элементы	Тип	Определение
Retry	Переменная	Счетчик повторных попыток
MaxRetry	Константа	MaxRetry = 2, см. таблицу 97
t M-sequence	Время	См. формулу (A.6)
t CYC	Время	Услуга DL_SetMode предоставляет это значение с параметром «M-sequenceTime». См. формулу (A.7)
t initcyc	Время	См. A.2.6
MH_Conf_xxx	Вызов	См. таблицу 42

7.3.3.5 Конечная машина обработчика сообщений Устройства

На рисунке 42 показана конечная машина обработчика DL на Устройстве.

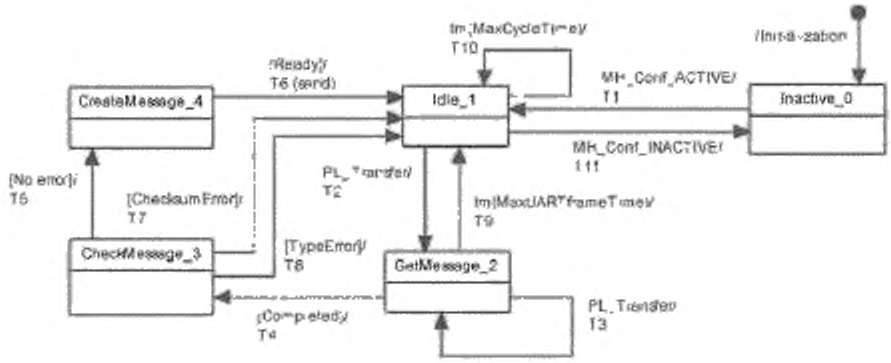


Рисунок 42 — Конечная машина обработчика сообщений Устройства

В таблице 45 показаны таблицы перехода состояний обработчика DL на Устройстве.

Таблица 45 — Таблицы перехода состояний обработчика сообщений на Устройстве

Имя состояния	Описание состояния
Inactive_0	Ожидание активации обработчиком DL через услугу MH_Conf_ACTIVE (см. таблицу 43, Переход T1)
Idle_1	Ожидание первого фрейма UART сообщения Ведущего узла через индикацию услуги PL_Transfer. Проверка времени истечения максимального времени цикла «MaxCycleTime»
GetMessage_2	Получение фрейма UART сообщения Ведущего узла. Проверка числа полученных фреймов UART (Устройство знает тип M-последовательности и, следовательно, число фреймов UART). Проверка времени истечения максимального времени фрейма UART «MaxUARTframeTime»
CheckMessage_3	Проверка типа M-последовательности и контрольной суммы полученного сообщения
Create Message_4	Составление сообщения из услуг OD.rsp, PD.rsp, EventFlag и PDStatus

Окончание таблицы 45

Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	2	Запустить отсчет максимального времени фрейма UART «MaxUARTframeTime» и максимального времени цикла «MaxCycleTime» при нахождении в состоянии OPERATE
T3	2	2	Повторно запустить таймер максимального времени фрейма UART «MaxUARTframeTime»
T4	2	3	Сбросить таймер максимального времени фрейма UART «MaxUARTframeTime»
T5	3	4	Вызвать индикацию услуг OD.ind и PD.ind
T6	4	1	Компилировать и вызвать ответ на услугу PL_Transfer.rsp (Устройство посылает ответное сообщение)
T7	3	1	Сообщить об ошибке обработчику DL через MHInfo (CHECKSUM_MISMATCH)
T8	3	1	Сообщить об ошибке обработчику DL через MHInfo (ILLEGAL_MESSAGETYPE)
T9	2	1	Сбросить оба таймера «MaxUARTframeTime» и «MaxCycleTime»
T10	1	1	Сообщить об ошибке обработчику DL через MHInfo (COMLOST). Исполнительные устройства будут следить за этой информацией и предпринимать соответствующие действия (см. 10.2 и 10.7.3)
T11	1	0	Обработчик сообщений Устройства изменяет состояние на Inactive_0
Внутренние элементы		Тип	Определение
MaxUARTFrameTime		Время	Время для передачи фрейма UART (11 TBIT) плюс максимум из t_1 (1 T_{BIT}) = 11 T_{BIT}
MaxCycleTime		Время	Проверить для таймера «MaxCycleTime», не занял ли обмен циклическими PD слишком много времени и не был ли он прерван. MaxCycleTime должно быть больше, чем MasterCycleTime (см. A.3.7)
TypeError		Условие	Обнаружена одна из следующих ошибок: ILLEGAL_MESSAGETYPE или COMLOST
ChecksumError		Условие	Обнаружена ошибка контрольной суммы сообщения

7.3.4 Обработчик Данных процесса

7.3.4.1 Общие положения

Транспортировка выходных PD выполняется, применяя услугу DL_OutputUpdate, а транспортировка входных PD — применяя услугу DL_InputTransport (см. рисунок 26). Цикл PD завершен, когда весь набор PD передан между Ведущим узлом и Устройством в нужном направлении. Данный цикл может потребовать более одной M-последовательности.

Все PD передаются в одной M-последовательности при использовании M-последовательностей типа TYPE_2_x (см. рисунок 37). В данном случае время исполнения цикла PD равно времени цикла t_{CYC}.

7.3.4.2 Режим расслоения

В данном случае все PD и OD передаются с несколькими перемежающимися M-последовательностями типов TYPE_1_1 (PD) и TYPE_1_2 (OD), как показано на рисунке 43. Это демонстрируется сообщениями Ведущего узла, записывающими выходные PD на Устройство. Параметр PDOutAddress услуги показывает расчленение выходных PD, подлежащих передаче (см. 7.2.2.3). Для входных PD параметр PDInAddress услуги соответственно указывает на расчленение входных PD. В цикле PD все входные PD будут считываться вначале с последующими выходными PD, подлежащими записи. Цикл PD включает все времена циклов, требуемых для передачи полных PD.

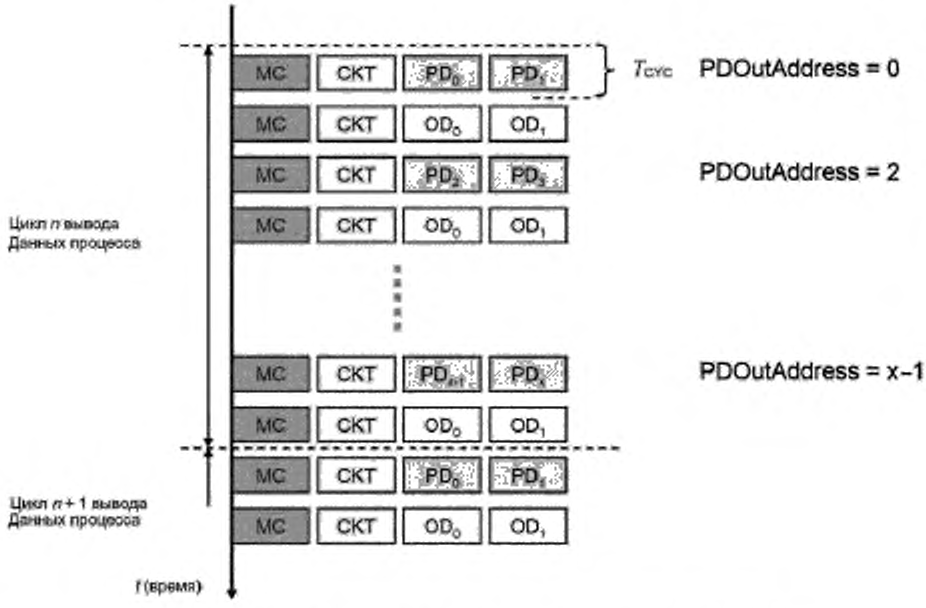


Рисунок 43 — Режим расслоения для сегментированной передачи PD

Режим расслоения применяется только для устаревших Устройств.

7.3.4.3 Конечная машина обработчика Данных процесса Ведущего узла

На рисунке 44 показана конечная машина обработчика PD Ведущего узла.

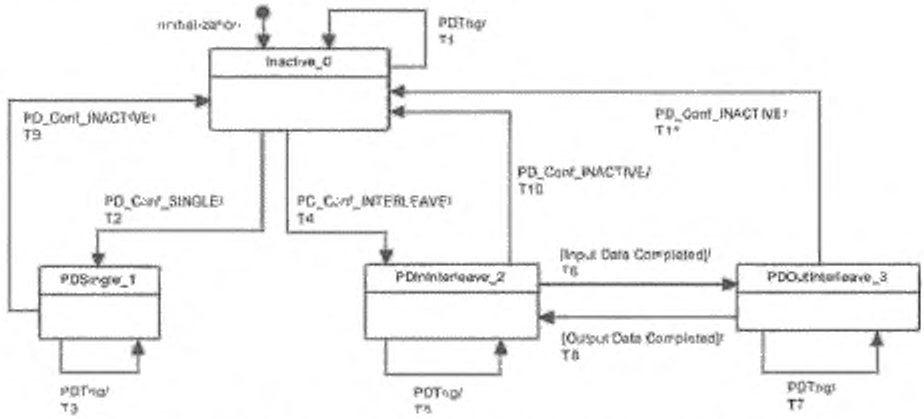


Рисунок 44 — Конечная машина обработчика PD Ведущего узла

В таблице 46 показаны таблицы перехода состояний обработчика PD на Ведущем узле.

Таблица 46 — Таблицы перехода состояний обработчика PD на Ведущем узле

Имя состояния	Описание состояния
Inactive_0	Ожидание активации
PDSingle_1	Передача PD в одной-единственной М-последовательности

Окончание таблицы 46

Имя состояния		Описание состояния	
PDInInterleave_2		Передача входных PD в режиме расслоения	
PDOutInterleave_3		Передача выходных PD в режиме расслоения	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	0	Вызывает услугу PD.req без PD
T2	0	1	Примечание — Обработчик DL сконфигурировал обработчик PD для передачи одной порции PD (см. таблицу 42, переходы T10 или T11)
T3	1	1	Берет данные из услуги DL_PDOutputUpdate и вызывает PD.req для передачи данных обработчику сообщений. Берет данные из PD.cnf и вызывает DL_PDInputTransport.ind и DL_PDCycle.ind для передачи входных PD на AL
T4	0	2	Примечание — Сконфигурировано для передачи PD с расслоением (см. таблицу 42, переходы T10 или T11)
T5	2	2	Вызывает PD.req и применяет PD.cnf для подготовки DL_PDInput Transport.ind
T6	2	3	Вызывает DL_PDInputTransport.ind и DL_PDCycle.ind для передачи входных PD на AL (см. 7.2.1.11)
T7	3	3	Берет данные из услуги DL_PDOutputUpdate и вызывает PD.req для передачи данных обработчику сообщений
T8	3	2	Вызывает DL_PDCycle.ind для указания конца цикла PD AL (см. 7.2.1.12)
T9	1	0	—
T10	2	0	—
T11	3	0	—
Внутренние элементы		Тип	Определение
<отсутствуют>			

7.3.4.4 Конечная машина обработчика Данных процесса Устройства

На рисунке 45 показана конечная машина обработчика PD Устройства.

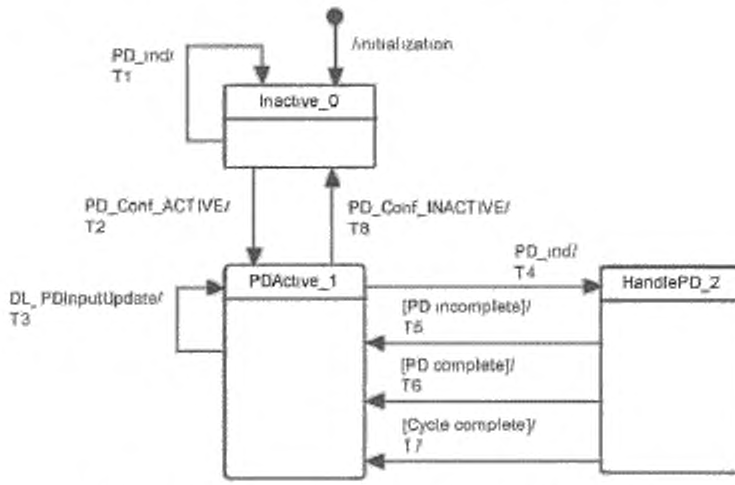


Рисунок 45 — Конечная машина обработчика PD Устройства

См. диаграмму последовательностей на рисунках 65 и 66.

В таблице 47 показаны таблицы перехода состояний обработчика PD на Устройстве.

Таблица 47 — Таблицы перехода состояний обработчика PD на Устройстве

Имя состояния		Описание состояния	
Inactive_0		Ожидание активации	
PDActive_1		Обработчик активен и ожидает следующего запроса обработчика сообщений через службу PD или службу DL_PDInputUpdate из AL	
HandlePD_2		Проверка PD на полноту в режиме расслоения	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	0	Игнорирует PD
T2	0	1	—
T3	1	1	Подготавливает входные PD для PD.rsp для формирования следующего запроса обработчика сообщений
T4	1	2	Обработчик сообщений запрашивает входные PD через услугу PD.ind и доставляет выходные PD или сегмент выходных PD. Вызывает PD.rsp с входными PD, когда находится не в режиме расслоения (см. 7.2.2.3)
T5	2	1	—
T6	2	1	Вызывает DL_PDOutputTransport.ind (см. 7.2.1.9)
T7	2	1	Вызывает DL_PDCycle.ind (см. 7.2.1.12)
T8	1	0	—
Внутренние элементы		Тип	Определение
PD_ind		Метка	Вызов услуги PD.ind, приходящий от обработчика сообщений

7.3.5 Обработчик Данных запроса

7.3.5.1 Общие положения

Обработчик OD Ведущего узла — это подчиненная конечная машина, активная в состояниях «Startup_2», «PreOperate_3» и «Operate_4» обработчика канального режима (см. рисунок 33). Конечная машина управляет тремя другими конечными машинами, так называемым обработчиком ISDU, обработчиком команд и обработчиком Событий. По умолчанию конечная машина всегда запускается в режиме обработчика ISDU.

При любом получении услуги EventFlag.ind конечная машина переходит в обработчик Событий. После окончания считывания информации о Событии конечная машина возвращается в состояние обработчика ISDU.

При любом получении от услуг DL_Control.req или PDInStatus.ind, когда машина находится в режиме обработчика ISDU или обработчика Событий, конечная машина переходит в режим обработчика команд. После того как команда обслужена, конечная машина возвращается в предыдущее активное состояние (обработчика ISDU или обработчика Событий).

7.3.5.2 Конечная машина обработчика Данных запроса Устройства

На рисунке 46 показана конечная машина Ведущего узла обработчика OD.

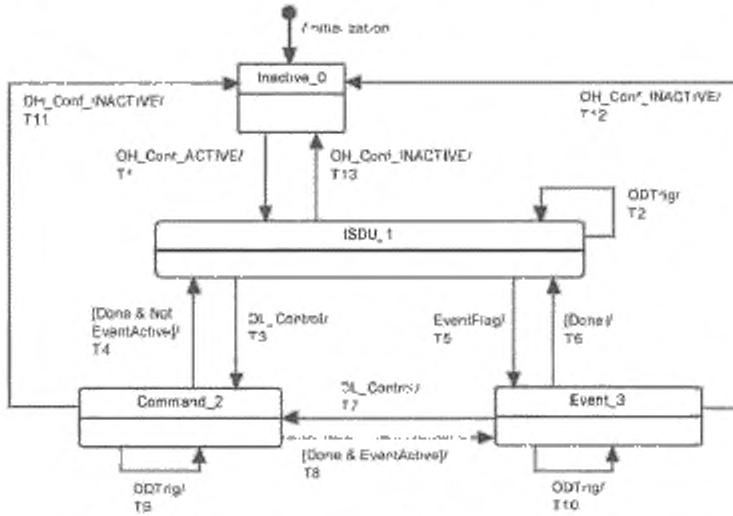


Рисунок 46 — Конечная машина обработчика OD Ведущего узла

Обработчик OD перенаправляет сервисный примитив ODTrig.ind для содержания следующего сообщения к текущему активному подконтрольному обработчику (ISDU, команд или Событий). Это осуществляется вызовом одной из услуг: ISDUTrig, CommandTrig или EventTrig.

В таблице 48 показаны таблицы перехода состояний обработчика OD на Ведущем узле.

Таблица 48 — Таблицы перехода состояний обработчика OD на Ведущем узле

Имя состояния		Описание состояния	
Inactive_0		Ожидание активации	
ISDU_1		Состояние по умолчанию — обработчик OD (низший приоритет)	
Command_2		Состояние управления Устройством с помощью команд с высшим приоритетом	
Event_3		Состояние передачи данных о Событии (ошибки, предупреждения, уведомления) с более высоким приоритетом	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	1	Обработчик OD передает услугу ODTrig.ind, называемую теперь ISDUTrig, обработчику ISDU (см. рисунок 49). В случае услуг DL_Read, DL_Write, DL_ReadParam или DL_WriteParam обработчик ISDU будет применять различные переходы (см. рисунок 49, переход T13)
T3	1	2	—
T4	2	1	—
T5	1	3	EventActive = TRUE
T6	3	1	EventActive = FALSE
T7	3	2	—
T8	2	3	—
T9	2	2	Обработчик OD передает услугу ODTrig.ind, называемую теперь CommandTrig, обработчику команд (см. рисунок 51)
T10	3	3	Обработчик OD передает услугу ODTrig.ind, называемую теперь EventTrig, обработчику Событий (см. рисунок 53)

Окончание таблицы 48

Переход	Исходное состояние	Конечное состояние	Действие
T11	2	0	—
T12	3	0	—
T13	1	0	—
Внутренние элементы		Тип	Определение
EventActive		Bool	Флаг указания направления возврата после прерывания обработки События высокоприоритетным запросом команды

7.3.5.3 Конечная машина обработчика Данных запроса Устройства

Обработчик OD на Устройстве получает информацию по каналу связи и параметр или адрес FlowCTRL через услугу OD.ind. Каналы связи совершенно независимы. В случае правильного доступа соответствующая конечная машина обработчика ISDU, обработчика команд или обработчика Событий адресуется через соответствующий канал связи.

Устройство будет отвечать на запросы чтения из нереализованных диапазонов адресов со значением «0». Оно будет игнорировать запросы записи в нереализованные диапазоны адресов.

На рисунке 47 показана конечная машина обработчика OD на Устройстве.

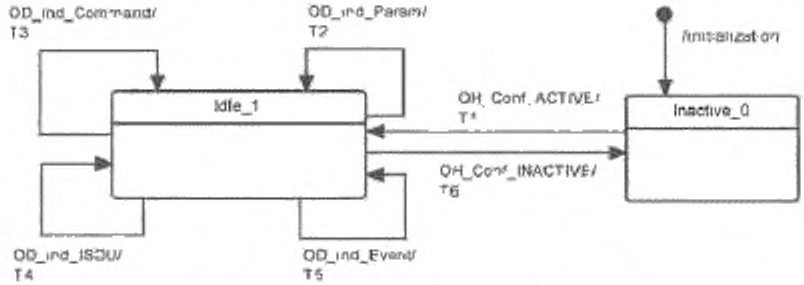


Рисунок 47 — Конечная машина обработчика OD на Устройстве

В случае доступа ISDU в Устройстве, не поддерживающем ISDU, Устройство будет отвечать «No Service» (Нет обслуживания) (см. таблицу А.12). Сообщение об ошибке не генерируется.

Примечание — Если нет OD, ожидающих передачи, то сообщением по умолчанию является OD.ind (R, ISDU, FlowCTRL = IDLE).

В таблице 49 показаны таблицы перехода состояний обработчика OD на Устройстве.

Таблица 49 — Таблицы перехода состояний обработчика OD на Устройстве

Имя состояния		Описание состояния	
Inactive_0		Ожидание активации	
Idle_1		Ожидание сообщений с OD через индикацию услуги OD. Декомпозиция и анализ	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	1	Перенаправляет к обработчику ISDU (страница Непосредственных параметров)
T3	1	1	Перенаправляет к обработчику команд
T4	1	1	Перенаправляет к обработчику ISDU
T5	1	1	Перенаправляет к обработчику Событий
T6	1	0	—

Окончание таблицы 49

Внутренние элементы	Тип	Определение
OD_ind_Param	Услуга	Альтернативное имя услуги OD.ind (R/W, PAGE, от 16 до 31, Data) в случае DL_ReadParam или DL_WriteParam
OD_ind_Command	Услуга	Альтернативное имя для услуги OD.ind (W, PAGE, 0, MasterCommand)
OD_ind_ISDU	Услуга	Альтернативное имя для услуги OD.ind (R/W, ISDU, FlowCtrl, Data)
OD_ind_Event	Услуга	Альтернативное имя для услуги OD.ind (R/W, DIAGNOSIS, n, Data)

7.3.6 Обработчик ISDU

7.3.6.1 Индексированный сервисный блок данных (ISDU)

Общая структура ISDU показана на рисунке 48 и подробно описана в подразделе А.5.



Рисунок 48 — Структура ISDU

Последовательность элементов соответствует последовательности передачи. Элементы ISDU могут принимать различные формы в зависимости от типа I-Service (см. А.5.2 и таблицу А.12).

ISDU позволяет получать доступ к объектам данных (параметры и команды), подлежащим передаче (см. рисунок 5). Объекты данных адресуются с помощью индекса «Index».

Все многооктетные типы данных будут передаваться как последовательность с обратным порядком следования октетов, т. е. старший октет (MSO) будет посылаться первым, затем менее значащие октеты в убывающем порядке, и младший октет (LSO) будет послан последним, как показано на рисунке 2.

7.3.6.2 Передача ISDU

ISDU передается по каналу связи ISDU (см рисунок 7 и А.1.2). Для выполнения такой передачи, как правило, требуется несколько сообщений (сегментация). Ведущий узел передает ISDU, посылая запрос I-Service (Read/Write) Устройству через канал связи ISDU. Затем по данному каналу он получает ответ Устройства.

В канале связи ISDU элемент адреса Address в управляющем октете M-последовательности размещает счетчик (= FlowCTRL). FlowCTRL контролирует поток сегментированных данных (см. А.1.2), подсчитывая элементы ISDU (по модулю 16) во время передачи.

Ведущий узел применяет элемент длины Length из ISDU и FlowCTRL для проверки завершения передачи.

Допустимые значения FlowCTRL приведены в таблице 50.

Таблица 50 — Определения FlowCTRL

FlowCTRL	Определение
0x00 to 0x0F	COUNT Счетчик элементов в ISDU. Увеличивается начиная с 1 после запуска ISDU ISDU START. Возвращается от 15 к 0 при Событии переполнения

Окончание таблицы 50

FlowCTRL	Определение
0x10	START Запуск I-Service ISDU, т. е. запуск запроса или ответа. При запуске запроса любые ранее незавершенные услуги могут быть отброшены. При запуске запроса, связанного с ответом, Устройство будет посылать сообщение «No Service» (Нет обслуживания) до тех пор, пока приложение не возвратит данные для ответа (см. таблицу А.12)
0x11	IDLE 1 Нет запроса на передачу ISDU
0x12	IDLE 2: Зарезервировано для будущего использования Нет запроса на передачу ISDU
0x13 to 0x1E	Зарезервировано
0x1F	ABORT Отменить обслуживание. Ведущий узел отвечает отбрасыванием полученных данных ответа. Устройство отвечает отбрасыванием полученных OD и может сгенерировать аварийное прекращение

В состоянии Idle_1 значения от 0x12 до 0x1F не будут приводить к ошибке связи.

7.3.6.3 Конечная машина обработчика ISDU на Ведущем узле

На рисунке 49 показана конечная машина обработчика ISDU на Ведущем узле.

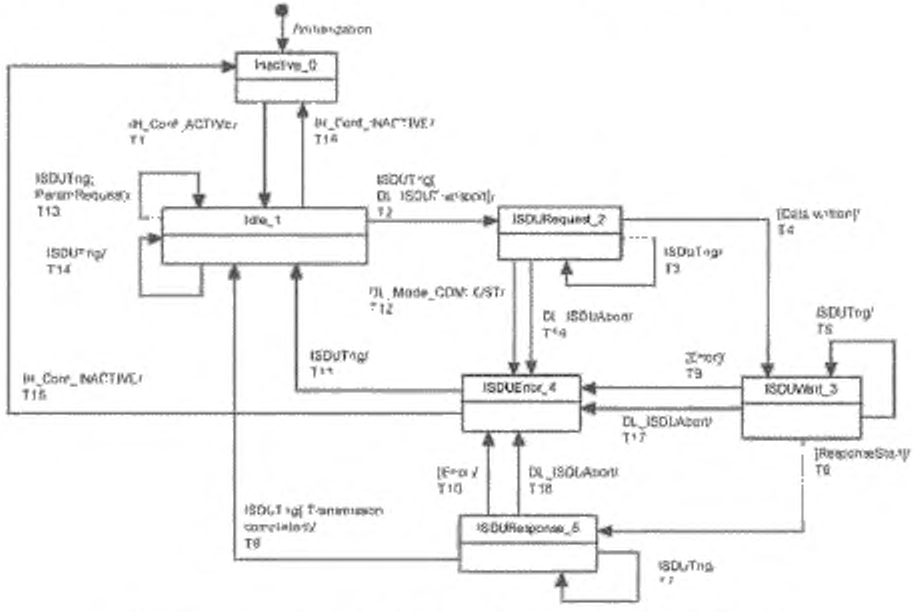


Рисунок 49 — Конечная машина обработчика ISDU на Ведущем узле

В таблице 51 показаны таблицы перехода состояний обработчика ISDU на Ведущем узле.

Таблица 51 — Таблицы перехода состояний обработчика ISDU на Ведущем узле

Имя состояния	Описание состояния
Inactive_0	Ожидание активации
Idle_1	Ожидание передачи следующих OD
ISDURequest_2	Передача OD ISDU

Окончание таблицы 51

Имя состояния		Описание состояния	
ISDUWait_3		Ожидание ответа от Устройства. Наблюдение за временем ISDU ISDUTime	
ISDUError_4		Обработка ошибки после ее обнаружения: Вызов отрицательного ответа DL_ISDU_Transport с информацией об ошибке при передаче ISDU ISDUTransportErrorInfo	
ISDUResponse_5		Получение ответа от Устройства	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	2	Вызывает OD.req с условием запуска записи ISDU: OD.req (W, ISDU, flowCtrl = START, данные)
T3	2	2	Вызывает OD.req записью данных ISDU и FlowCTRL при условиях, приведенных в таблице 50
T4	2	3	Запускает таймер ISDU (ISDUTime)
T5	3	3	Вызывает OD.req с условием запуска чтения ISDU: OD.req (R, ISDU, flowCtrl = START)
T6	3	5	Останавливает таймер (ISDUTime)
T7	5	5	Вызывает OD.req с чтением данных ISDU и FlowCTRL при условиях, приведенных в таблице 50
T8	5	1	Вызывает положительное подтверждение DL_ ISDUTransport
T9	3	4	—
T10	5	4	—
T11	4	1	Вызывает OD.req с прекращением ISDU: OD.req (R, ISDU, flowCtrl = ABORT). Вызывает отрицательное подтверждение DL_ ISDUTransport
T12	2	4	—
T13	1	1	Вызывает OD.req с соответствующими данными. Вызывает положительное подтверждение DL_ReadParam/DL_WriteParam
T14	1	1	Вызывает OD.req с сообщением о простое: OD.req (R, ISDU, flowCtrl = IDLE)
T15	4	1	В случае потери связи обработчик сообщений информирует обработчик DL, который, в свою очередь, применяет административный вызов IH_Conf_INACTIVE. Во время этой передачи не требуется никаких действий
T16	1	0	—
T17	3	4	—
T18	5	4	—
T19	2	4	—
Внутренние элементы		Тип	Определение
ISDUTime		Время	Измерение времени ответа Устройства (сторожевая схема, см. таблицу 97)
ResponseStart		Услуга	OD.cnf (данные, отличные от ISDU_BUSY)
ParamRequest		Услуга	DL_ReadParam или DL_WriteParam
Error		Переменная	Любая определимая ошибка в передаче ISDU, или запросы DL_ISDUAbort, или любые нарушения времени подтверждения ISDU (см. таблицу 97)

7.3.6.4 Конечная машина обработчика ISDU на Устройстве

На рисунке 50 показана конечная машина обработчика ISDU на Устройстве.

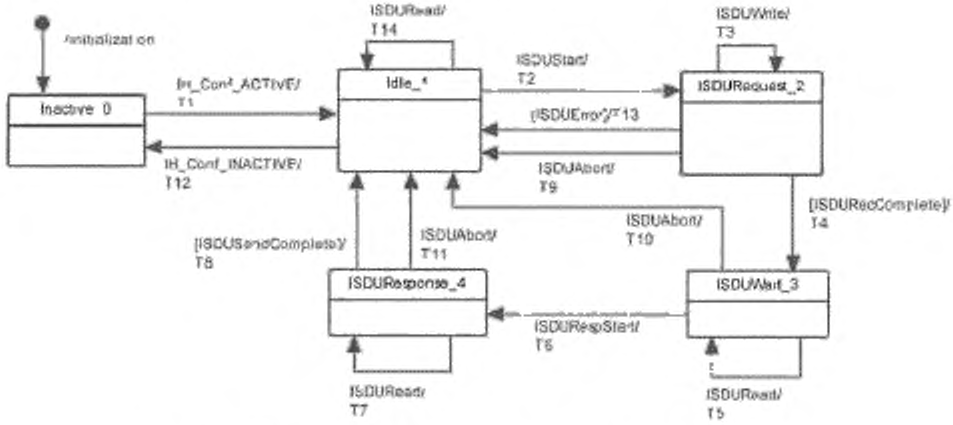


Рисунок 50 — Конечная машина обработчика ISDU на Устройстве

В таблице 52 показаны таблицы перехода состояний обработчика ISDU на Устройстве.

Таблица 52 — Таблицы перехода состояний обработчика ISDU на Устройстве

Имя состояния		Описание состояния	
Inactive_0		Ожидание активации	
Idle_1		Ожидание передачи следующего ISDU	
ISDURequest_2		Прием запроса ISDU	
ISDUWait_3		Ожидание данных для передачи из AL (см. DL_ISDUTransport)	
ISDUResponse_4		Передача данных ответа ISDU	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	2	Начинает получение OD ISDU
T3	2	2	Получает OD ISDU
T4	2	3	Вызывает DL_ ISDUTransport.ind к AL (см. 7.2.1.6)
T5	3	3	Вызывает OD.rsp с индикацией «busy» («занято») (см. таблицу А.14)
T6	3	4	—
T7	4	4	Вызывает OD.rsp с данными ответа ISDU
T8	4	1	—
T9	2	1	—
T10	3	1	Вызывает DL_ ISDUAbort
T11	4	1	Вызывает DL_ ISDUAbort
T12	1	0	—
T13	2	1	—
T14	1	1	Вызывает OD.rsp с индикацией «no service» («нет обслуживания») (см. таблицы А.12 и А.14)

Окончание таблицы 52

Внутренние элементы	Тип	Определение
ISDUStart	Услуга	OD.ind(W, ISDU, Start, Data)
ISDUWrite	Услуга	OD.ind(W, ISDU, FlowCtrl, Data)
ISDURecComplete	Условие	Если получено OD.ind(R, ISDU, Start, ...)
ISDURespStart	Услуга	DL_ ISDUTransport.rsp()
ISDURead	Услуга	OD.ind(R, ISDU, Start or FlowCtrl, ...)
ISDUSendComplete	Условие	Если получено OD.ind(R, ISDU, IDLE, ...)
ISDUAbort	Услуга	OD.ind(R/W, ISDU, Abort, ...)
ISDUError	Условие	Если структура ISDU неправильная

7.3.7 Обработчик команд

7.3.7.1 Общие положения

Обработчик команд посылает управляющий код (PDOUTVALID или PDOUTINVALID), содержащийся в сервисном примитиве DL_Control.req, к циклически работающему обработчику сообщений через услугу OD.req и команды Ведущего узла. Обработчик сообщений применяет канал связи страниц.

Допустимые управляющие коды для выходных PD приведены в таблице 53.

Таблица 53 — Управляющие коды

Управляющий код	Команда ведущего узла	Описание
PDOUTVALID	ProcessDataOutputOperate	Допустимые выходные PD
PDOUTINVALID	DeviceOperate	Недопустимые или отсутствующие выходные PD

Обработчик команд получает информацию о состоянии входных PD через услугу PDInStatus и распространяет ее в пределах сервисного примитива DL_Control.ind.

Кроме того, обработчик команд транслирует запросы на изменение режима Устройства от SM в соответствующие команды Ведущего узла (см. таблицу В.2).

7.3.7.2 Конечная машина обработчика команд Ведущего узла

На рисунке 51 показана конечная машина обработчика команд на Ведущем узле.

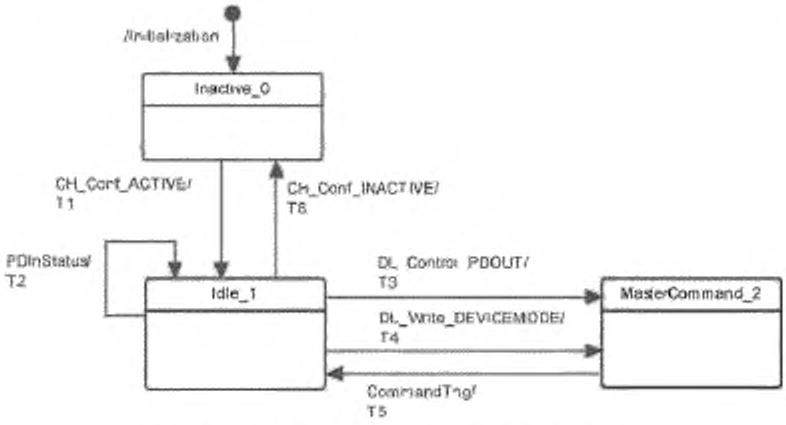


Рисунок 51 — Конечная машина обработчика команд на Ведущем узле

В таблице 54 показаны таблицы перехода состояний обработчика команд на Ведущем узле.

Таблица 54 — Таблицы перехода состояний обработчика команд на Ведущем узле

Имя состояния		Описание состояния	
Inactive_0		Ожидание активации обработчиком DL	
Idle_1		Ожидание новой команды из AL: DL_Control (состояние или выходные PD), или от SM: DL_Write (изменение режима Устройства, например, на режим OPERATE), или ожидание сервисного примитива PDInStatus.ind	
MasterCommand_2		Подготовка данных для сервисного примитива OD.req. Ожидание запроса от обработчика OD (CommandTrig)	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	1	Если услуга PDInStatus.ind = VALID, вызывает DL_Control.ind (VALID) для сообщения AL о допустимых входных PD. Если услуга PDInStatus.ind = INVALID, вызывает DL_Control.ind (INVALID) для сообщения AL о недопустимых входных PD
T3	1	1	Если услуга DL_Control.req = PDOUTVALID, вызывает OD.req (WRITE, PAGE, 0, 1, MasterCommand = 0x98). Если услуга DL_Control.req = PDOUTINVALID, вызывает OD.req (WRITE, PAGE, 0, 1, MasterCommand = 0x98). См. таблицу B.2
T4	1	2	Услуги DL_Write_DEVICEMODE транслируют в: INACTIVE: OD.req (WRITE, PAGE, 0, 1, MasterCommand = 0x5A) STARTUP: OD.req (WRITE, PAGE, 0, 1, MasterCommand = 0x97) REOPERATE: OD.req (WRITE, PAGE, 0, 1, MasterCommand = 0x9A) OPERATE: OD.req (WRITE, PAGE, 0, 1, MasterCommand = 0x99)
T5	2	1	Вызов услуги CommandTrig из обработчика OD заставляет обработчик команд вызвать сервисный примитив OD.req и затем обработчик сообщений для посылки соответствующей команды Ведущего узла в Устройство
T6	1	0	—
Внутренние элементы		Тип	Определение
DEVICEMODE		Метка	Любой из следующих режимов Устройства: INACTIVE, STARTUP, PREOPERATE или OPERATE
PDOUT		Метка	Любой из двух выходных управляющих кодов: PDOUTVALID или PDOUTINVALID (см. таблицу 53)

7.3.7.3 Конечная машина обработчика команд на Устройстве

На рисунке 52 показана конечная машина обработчика команд на Устройстве. Как правило, она контролируется командами Ведущего узла из обработчика команд Ведущего узла для управления режимами Устройства и состоянием выходных PD. Она также контролирует состояние входных PD через услугу PDInStatus.

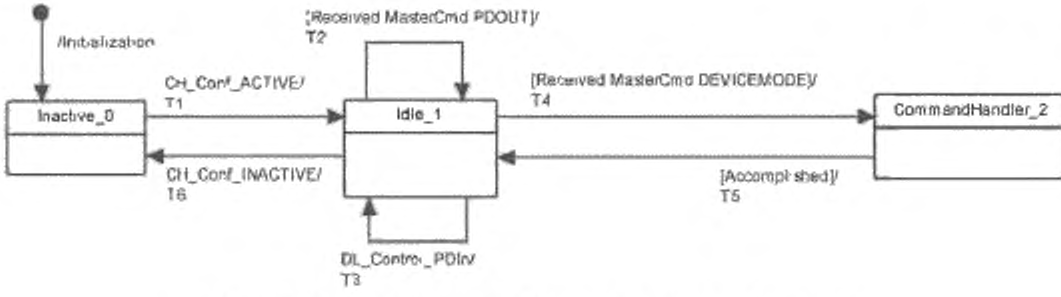


Рисунок 52 — Конечная машина обработчика команд на Устройстве

В таблице 55 показаны таблицы перехода состояний обработчика команд на Устройстве.

Таблица 55 — Таблицы перехода состояний обработчика команд на Устройстве

Имя состояния		Описание состояния	
Inactive_0		Ожидание активации	
Idle_1		Ожидание следующей команды Ведущего узла	
CommandHandler_2		Разложение команды Ведущего узла и вызов конкретных действий (см. B.1.2): если команда Ведущего узла = 0x5A, то изменить состояние Устройства на INACTIVE. Если команда Ведущего узла = 0x97, то изменить состояние Устройства на STARTUP. Если команда Ведущего узла = 0x9A, то изменить состояние Устройства на PREOPE-RATE. Если команда Ведущего узла = 0x99, то изменить состояние Устройства на OPERATE	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	1	Вызывает DL_Control.ind (PDOUTVALID), если полученная команда Ведущего узла = 0x98. Вызывает DL_Control.ind (PDOUTINVALID), если полученная команда Ведущего узла = 0x99
T3	1	1	Если услуга DL_Control.req (VALID), то вызывает PDInStatus.req (VALID). Если услуга DL_Control.req (INVALID), то вызывает PDInStatus.req (INVALID). Обработчик сообщений применяет услугу PDInStatus для установки или сброса флага состояния PD (см. A.1.5)
T4	1	2	—
T5	2	1	—
T6	1	0	—
Внутренние элементы		Тип	Определение
<отсутствуют>			

7.3.8 Обработчик Событий

7.3.8.1 События

Имеется два типа Событий: один — без деталей (т. е. подробной информации), другой — с деталями. События без деталей могут быть реализованы в устаревших Устройствах, но они не будут применяться для Устройств в соответствии с настоящим стандартом. Однако все Ведущие узлы будут поддерживать обработку обоих типов событий: с деталями и без деталей.

Общая структура и кодирование Событий установлены в разделе A.6. Коды событий без деталей установлены в таблице A.16. Коды событий EventCodes с деталями установлены в приложении D. Структура памяти Событий для кодов событий с деталями установлена в таблице 56.

Таблица 56 — Память Событий

Адрес	Номер области памяти События	Имя параметра	Описание
0x00		StatusCode	Сводка состояния и информации об ошибке. Применяется также для управления доступом на чтение для отдельных сообщений
0x01	1	EventQualifier 1	Тип, режим и источник События
0x02		EventCode 1	16-битовый код События
0x03			
0x04	2	EventQualifier 2	Тип, режим и источник События
0x05		EventCode 2	16-битовый код События
0x06			

Окончание табилицы 56

Адрес	Номер области памяти События	Имя параметра	Описание
...			
0x10	6	EventQualifier 6	Тип, режим и источник События
0x11		EventCode 6	16-битовый код События
0x12			
0x13 to 0x1F			Зарезервировано для будущего использования

7.3.8.2 Обработка событий

AL Устройства записывает Событие в память Событий и затем устанавливает бит флага события Event flag, который посылается в Ведущий узел в следующем сообщении в пределах октета контрольной суммы CKS (см. 7.3.3.2 и А.1.5).

При получении ответного сообщения Устройства с битом флага События Event flag = 1 Ведущий узел переключится из режима обработчика ISDU в режим обработчика Событий. Обработчик Событий начинает чтение кода состояния StatusCode.

Если бит «Event Details» установлен (см. рисунок А.23), Ведущий узел будет читать подробную информацию о Событиях, указанную в коде состояния StatusCode из памяти Событий. После чтения деталей События он вызовет услугу DL_Event.ind. После получения услуги DL_EventConf Ведущий узел запишет произвольную информацию в код состояния StatusCode, чтобы сбросить бит флага события Event flag. Обработка События на Ведущем узле будет завершена вне зависимости от содержания полученных данных события (EventQualifier, EventCode).

Если бит деталей события Event Details не установлен (см. рисунок А.22), Ведущий узел сгенерирует стандартизованные События в соответствии с таблицей А.16, начиная со старшего бита в коде События EventCode.

Доступ на запись к коду состояния StatusCode указывает на конец обработки События для Устройства. Устройство будет игнорировать данные этого доступа Ведущего узла на запись. Затем устройство сбрасывает бит флага события Event flag и может теперь изменять содержание полей в памяти Событий.

7.3.8.3 Конечная машина обработчика Событий Ведущего узла

На рисунке 53 показана конечная машина обработчика Событий на Ведущем узле.

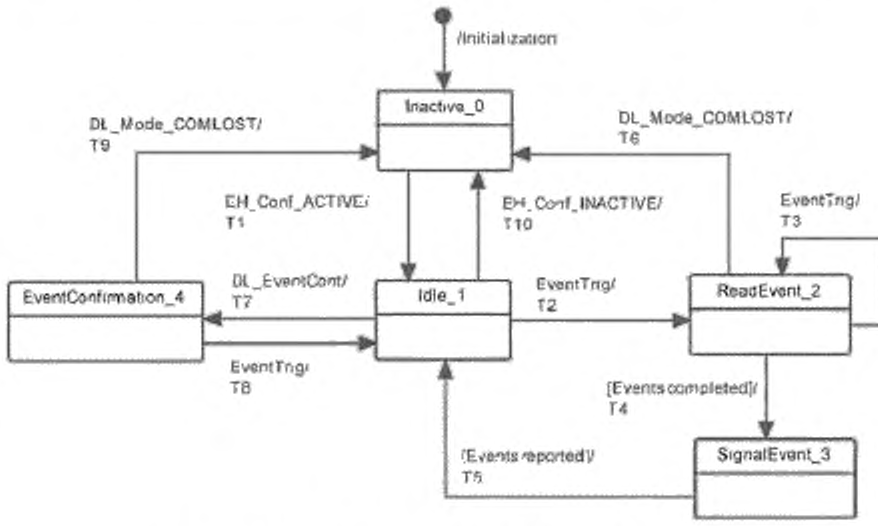


Рисунок 53 — Конечная машина обработчика Событий на Ведущем узле

В таблице 57 показаны таблицы перехода состояний обработчика Событий на Ведущем узле.

Таблица 57 — Таблицы перехода состояний обработчика Событий на Ведущем узле

Имя состояния		Описание состояния	
Inactive_0		Ожидание активации	
Idle_1		Ожидание следующей индикации События (услуга EventTrig через обработчик OD) или подтверждения События через услугу DL_EventConf из AL Ведущего узла	
ReadEvent_2		Чтение набора данных События из сообщения Устройства с использованием адреса памяти Событий. Проверка кода состояния StatusCode на предмет числа активированных Событий (см. таблицу 56)	
SignalEvent_3		Анализ данных События и вызов индикации DL_Event к AL Ведущего узла (см. 7.2.1.15) для каждого доступного События	
Event Confirmation_4		Ожидание передачи подтверждения События через услугу OD.req в Устройство	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	2	Чтение октета кода состояния StatusCode через услугу OD.req (R, DIAGNOSIS, Event memory address = 0, 1)
T3	2	2	Чтение октета из памяти Событий через услугу OD.req (R, DIAGNOSIS, увеличенный адрес памяти Событий, 1)
T4	2	3	—
T5	3	1	—
T6	2	0	—
T7	1	4	—
T8	4	1	Вызывает OD.req (W, DIAGNOSIS, 0, 1, любые данные) с доступом на запись к коду состояния StatusCode (см. таблицу 56), чтобы подтвердить Устройству факт считывания События
T9	4	0	—
T10	1	0	—
Внутренние элементы		Тип	Определение
<отсутствуют>			

7.3.8.4 Конечная машина обработчика Событий на Устройстве

На рисунке 54 показана конечная машина обработчика событий на Устройстве.

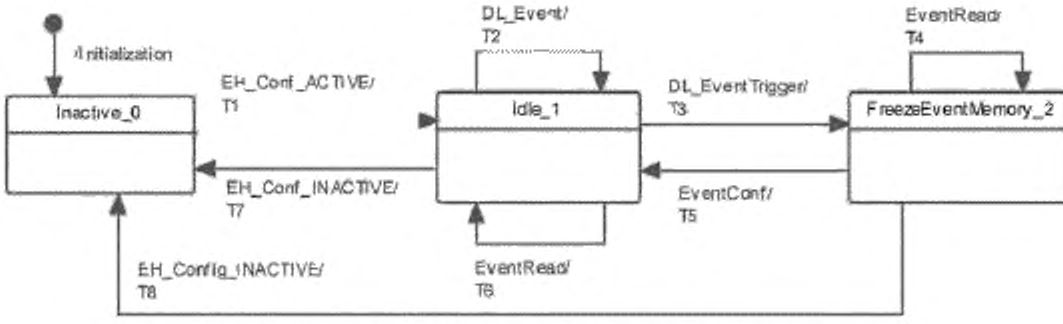


Рисунок 54 — Конечная машина обработчика Событий на Устройстве

В таблице 58 показаны таблицы перехода состояний обработчика Событий на Устройстве.

Таблица 58 — Таблицы перехода состояний обработчика Событий на Устройстве

Имя состояния		Описание состояния	
Inactive_0		Ожидание активации	
Idle_1		Ожидание услуги DL_Event из AL, предоставляющей данные о Событии, и услуги DL_EventTrigger для установки бита флага события Event flag (см. A.1.5)	
FreezeEventMemory_2		Ожидание чтения памяти Событий и подтверждение считывания памяти Событий через доступ на запись в код состояния StatusCode	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	1	Изменение входов памяти Событий новыми данными События (см. таблицу 56)
T3	1	2	Вызывает услугу EventFlag.req (Flag = TRUE) для индикации Ведущему узлу активации События через бит флага события Event flag. Отметить все входы памяти Событий как неизменяемые
T4	2	2	Ведущий узел требует данные памяти Событий через услугу EventRead (= OD.ind). Посылает данные События OD.rsp с данными из указанных адресов памяти Событий
T5	2	1	Вызывает услугу EventFlag.req (Flag = FALSE) для индикации Ведущему узлу факта деактивации События через бит флага события «Event flag». Отмечает все адреса в памяти Событий как недоступные в соответствии с A.6.3
T6	1	1	Посылает содержимое памяти Событий, вызывая услугу OD.rsp с данными События
T7	1	0	—
T8	2	0	Сбрасывает данные памяти Событий
Внутренние элементы		Тип	Определение
EventRead		Услуга	OD.ind (R, DIAGNOSIS, адрес памяти События, длина, данные)
EventConf		Услуга	OD.ind (W, DIAGNOSIS, address = 0, данные = безразлично какие)

8 Прикладной уровень (AL)

8.1 Общие положения

На рисунке 55 приводятся структура и услуги AL Ведущего узла.



Рисунок 55 — Структура и услуги AL (Ведущий узел)

На рисунке 56 приводятся структура и услуги AL Устройства.



Рисунок 56 — Структура и услуги AL (Устройство)

8.2 Услуги прикладного уровня

8.2.1 Услуги прикладного уровня на Ведущем узле и Устройстве

В разделе 8 устанавливаются услуги AL, предоставляемые приложениям Ведущего узла и Устройства и модулю управления системой через его внешние интерфейсы. В таблице 59 приводятся функции Ведущего узла и Устройства в соответствии с их ролями инициатора и получателя отдельных услуг AL. Пустые поля указывают, что данная услуга отсутствует на Ведущем узле или на Устройстве.

Таблица 59 — Услуги AL на Ведущем узле и на Устройстве

Имя услуги	Ведущий узел	Устройство
AL_Read	R	I
AL_Write	R	I
AL_Abort	R	I
AL_GetInput	R	
AL_NewInput	I	
AL_SetInput		R
AL_PDCycle	I	I
AL_GetOutput		R
AL_NewOutput		I
AL_SetOutput	R	
AL_Event	I/R	R
AL_Control	I/R	I/R
Обозначения (см. 3.3.4): I — Инициатор услуги; R — Приемник (Ответчик) услуги.		

8.2.2 Услуги прикладного уровня

8.2.2.1 Услуга AL_Read

Услуга AL_Read применяется для чтения OD из Устройства, подключенного к конкретному порту. Параметры сервисных примитивов приведены в таблице 60.

Таблица 60 — AL_Read

Имя параметра	.req	ind	.rsp	.cnf
Argument	M	M		
Port	M			
Index	M	M		
Subindex	M	M		
Result (+)			S	S(=)
Port				M
Data			M	M(=)
Result (–)			S	S(=)
Port				M
ErrorInfo			M	M(=)

Argument (Аргумент)

Определяемые услугой параметры передаются в данном параметре.

Port (Порт)

Параметр содержит номер порта для считываемых OD.

Тип параметра: Unsigned8.

Index (Индекс)
Данный параметр указывает адрес объектов OD, считываемых из Устройства. Индекс 0 вместе с Субиндексом 0 адресует весь набор Непосредственных параметров от 0 до 15 (см. страницу 1 Непосредственных параметров в таблице B.1) или вместе с Субиндексами от 1 до 16 отдельные параметры от 0 до 15. Индекс 1 вместе с Субиндексом 0 адресует весь набор Непосредственных параметров с адресами от 16 до 31 (см. страницу 2 Непосредственных параметров в таблице B.1) или вместе с Субиндексами от 1 до 16 непосредственные параметры от 16 до 31. Он применяет канал связи страниц (см. рисунок 6) для обеих страниц и всегда возвращает положительный результат. Для всех других индексов (см. B.2) применяется канал связи ISDU.

Допустимые значения от 0 до 65 535 (ограничения описаны в пункте B.2.1).

Subindex (Субиндекс)
Данный параметр указывает номер элемента в структурированном объекте OD. Значение 0 указывает на весь набор элементов.

Допустимые значения от 0 до 255.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Port (Порт)
Данный параметр содержит номер порта затребованных OD.

Data (Данные)
Данный параметр содержит считанные значения OD.

Тип параметра: Строка октетов.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

Port (Порт)
Данный параметр содержит номер порта для запрошенных OD.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения см. приложение C.

Примечание — AL отображает информацию об ошибке DL на свою собственную информацию об ошибке, применяя приложение C.

8.2.2.2 Услуга AL_Write

Услуга AL_Write применяется для записи OD на Устройство, подключенное к указанному порту. Параметры сервисных примитивов приведены в таблице 61.

Таблица 61 — Услуга AL_Write

Имя параметра	.req	.ind	.rsp	.cnf
Argument	M	M		
Port	M			
Index	M	M		
Subindex	M	M		
Data		M(=)		
Result (+)			S	S(=)
Port				M
Result (–)			S	S(=)
Port				M
ErrorInfo			M	M(=)

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

Port (Порт)
Параметр содержит номер порта для записываемых OD.

Тип параметра: Unsigned8.

Index (Индекс)
Данный параметр указывает адрес объектов OD, записываемых на Устройство. Индекс 0 всегда возвращает отрицательный результат. Индекс 1 вместе с Субиндексом 0 адресует весь набор Непосредственных параметров с адресами от 16 до 31 (см. страницу 2 Непосредственных параметров в таблице B.1) или вместе с Субиндексами от 1 до 16 отдельные параметры от 16 до 31. Он применяется для канала связи страниц (см. рисунок 6) в случае Индекса 1 и всегда возвращает отрицательный результат. Для всех других индексов (см. подраздел B.2) применяется канал связи ISDU.

Допустимые значения от 1 до 65 535 (см. таблицу 97).

Subindex (Субиндекс)
Данный параметр указывает номер элемента в структурированном объекте OD. Значение 0 указывает на весь набор элементов.

Допустимые значения от 0 до 255.
Data (Данные)
Данный параметр содержит значения OD.

Тип параметра: Строка октетов.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Port (Порт)
Данный параметр содержит номер порта OD.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

Port (Порт)
Данный параметр содержит номер порта OD.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения см. приложение C.

8.2.2.3 Услуга AL_Abort

Услуга AL_Abort служит для прерывания услуг AL_Read или AL_Write на указанном порту. Вызов этой услуги отменяет ответ на услугу AL_Read или AL_Write, выполняемую на Ведущем узле. Параметры сервисных примитивов приведены в таблице 62.

Таблица 62 — AL_Abort

Имя параметра	.req	.ind
Argument	M	M
Port	M	

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

Port (Порт)
Данный параметр содержит номер порта прерываемой услуги.

8.2.2.4 Услуга AL_GetInput

Услуга AL_GetInput считывает входные данные в PD, предоставленных DL Устройства, подключенного к указанному порту. Параметры сервисных примитивов приведены в таблице 63.

Таблица 63 — Услуга AL_GetInput

Имя параметра	.req	.cnf
Argument	M	
Port	M	
Result (+)		S
Port		M
InputData		M
Result (–)		S
Port		M
ErrorInfo		M

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

Port (Порт)
Параметр содержит номер порта для считываемых PD.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Port (Порт)
Данный параметр содержит номер порта PD.

InputData (ВходныеДанные)
Данный параметр содержит значения затребованных входных PD указанного порта.

Тип параметра: Строка октетов.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

Port (Порт)
Данный параметр содержит номер порта PD.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
NO_DATA (DL не предоставляет PD).

8.2.2.5 Услуга AL_NewInput

Локальная услуга AL_NewInput указывает на получение обновленных входных данных в PD Устройства, подключенного к указанному порту. Параметры сервисных примитивов приведены в таблице 64.

Таблица 64 — Услуга AL_NewInput

Имя параметра	.ind
Argument	M
Port	M

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

Port (Порт)
Данный параметр содержит номер порта OD.

8.2.2.6 Услуга AL_SetInput

Локальная услуга AL_SetInput обновляет входные данные в PD Устройства. Параметры сервисных примитивов приведены в таблице 65.

Таблица 65 — Услуга AL_SetInput

Имя параметра	.req	.cnf
Argument	M	
InputData	M	
Result (+)		S
Result (–)		S
ErrorInfo		M

Argument (Аргумент)

Определяемые услугой параметры передаются в данном параметре.

InputData (ВходныеДанные)

Данный параметр содержит значения PD входных данных, подлежащих передаче.

Тип параметра: Строка октетов.

Result (+) (ПоложительныйРезультат)

Данный параметр выбора указывает, что услуга выполнена успешно.

Result (–) (ОтрицательныйРезультат)

Данный параметр выбора указывает, что услуга завершилась неудачно.

ErrorInfo (ИнформацияОбОшибке)

Данный параметр содержит информацию об ошибке.

Допустимые значения:
STATE_CONFLICT (в текущем состоянии услуга недоступна).

8.2.2.7 Услуга AL_PDCycle

Локальная услуга AL_PDCycle указывает на конец цикла PD. Приложение Устройства может применять эту услугу для передачи новых входных данных AL через услугу AL_SetInput. Параметры сервисных примитивов приведены в таблице 66.

Таблица 66 — Услуга AL_PDCycle

Имя параметра	.ind
Argument	
Port	O

Argument (Аргумент)

Определяемые услугой параметры передаются в данном параметре.

Port (Порт)

Данный параметр содержит номер порта полученных новых PD (только Ведущий узел).

8.2.2.8 Услуга AL_GetOutput

Услуга AL_GetOutput считывает выходные данные в PD, предоставленных DL Устройства. Параметры сервисных примитивов приведены в таблице 67.

Таблица 67 — Услуга AL_GetOutput

Имя параметра	.req	.cnf
Argument	M	
Result (+)		S
InputData		M
Result (–)		S
ErrorInfo		M

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

InputData (ВыходныеДанные)
Данный параметр содержит значения OD затребованных выходных данных.

Тип параметра: Строка октетов.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

ErrorInfor (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
NO_DATA (DL не предоставляет PD).

8.2.2.9 Услуга AL_NewOutput

Локальная услуга AL_NewOutput указывает на получение обновленных выходных данных в PD Устройства. Данная услуга не имеет параметров. Сервисные примитивы показаны в таблице 68.

Таблица 68 — Услуга AL_NewOutput

Имя параметра	.ind
<отсутствуют>	

8.2.2.10 Услуга AL_SetOutput

Локальная услуга AL_SetOutput обновляет выходные данные в PD Ведущего узла. Параметры сервисных примитивов приведены в таблице 69.

Таблица 69 — Услуга AL_SetOutput

Имя параметра	.req	.cnf
Argument	M	
Port	M	
OutputData	M	
Result (+)		S
Port		M
Result (–)		S
Port		M
ErrorInfo		M

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

Port (Порт)
Параметр содержит номер порта для записываемых PD.

OutputData (ВыходныеДанные)
Данный параметр содержит выходные данные, записываемые в указанном порте.

Тип параметра: Строка октетов.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Port (Порт)
Данный параметр содержит номер порта PD.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

Port (Порт)
Данный параметр содержит номер порта PD.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
STATE_CONFLICT (в текущем состоянии услуга недоступна).

8.2.2.11 Услуга AL_Event

Услуга AL_Event указывает до шести промежуточных состояний или сообщений об ошибке. Источник одного События может быть локальным (Ведущий узел) или удаленным (Устройство). Событие может запускаться уровнем связи или приложением. Параметры сервисных примитивов приведены в таблице 70.

Таблица 70 — Услуга AL_Event

Имя параметра		.req	.ind	.rsp	.cnf
Argument		M	M	M	M
Port			M	M	M
EventCount		M	M		
Event(1)	Instance	M	M		
	Mode	M	M		
	Type	M	M		
	Origin		M		
	EventCode	M	M		
Event(n)	Instance	M	M		
	Mode	M	M		
	Type	M	M		
	Origin		M		
	EventCode	M	M		

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

Port (Порт)
Данный параметр содержит номер порта данных События.

EventCounter (СчетчикСобытий)
Данный параметр указывает число *n* (от 1 до 6) Событий в памяти Событий.

Event(x) [Событие(x)]
В зависимости от значения счетчика событий Данный параметр существует в *n* экземплярах. Каждый экземпляр содержит следующие элементы.

Instance (Экземпляр)
Данный параметр указывает источник События.
Допустимые значения: приложение (см. таблицу А.17).

Mode (Режим)
Данный параметр указывает режим События.

Допустимые значения: SINGLESHOT, APPEARS, DISAPPEARS (см. таблицу А.20).

Type (Тип)
Данный параметр указывает категорию События.

Допустимые значения: ERROR, WARNING, NOTIFICATION (см. таблицу А.19).

Origin (Источник)
Данный параметр указывает, было ли Событие в локальной секции связи или это удаленное Событие (в Устройстве).

Допустимые значения: LOCAL, REMOTE.

EventCode (КодСобытия)
Данный параметр содержит код, идентифицирующий определенное Событие.

Допустимые значения см. приложение D.

8.2.2.12 Услуга AL_Control

Услуга AL_Control содержит информацию о состоянии определителя PD, передаваемую в приложение Устройства или из него. Параметры сервисных примитивов приведены в таблице 71.

Таблица 71 — Услуга AL_Control

Имя параметра	.req	.ind
Argument	M	M
Port	C	C
ControlCode	M	M

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

Port (Порт)
Данный параметр содержит номер соответствующего порта.

ControlCode (УправляющийКод)
Данный параметр указывает состояние описателя PD.

Допустимые значения:
VALID (входные PD допустимы);
INVALID (входные PD недопустимы);
PDOUTVALID (выходные PD допустимы, см. таблицу 53);
PDOUTVALID (выходные PD недопустимы, см. таблицу 53).

8.3 Протокол прикладного уровня

8.3.1 Обзор

На рисунке 7 показано, что AL предлагает услуги для объектов данных, которые преобразуются в специальные каналы связи DL.

AL управляет передачей данных всеми назначенными ему портами. Это означает, что вызовы услуг AL определяют конкретный порт, с которым они связаны.

8.3.2 Передача Данных запроса

8.3.2.1 Конечная машина Данных запроса прикладного уровня Ведущего узла

На рисунке 57 показана конечная машина для обработки OD на AL Ведущего узла. Термин AL_Service означает любую услугу AL из таблицы 59, связанную с OD. Обозначение Portx указывает на конкретный номер порта.

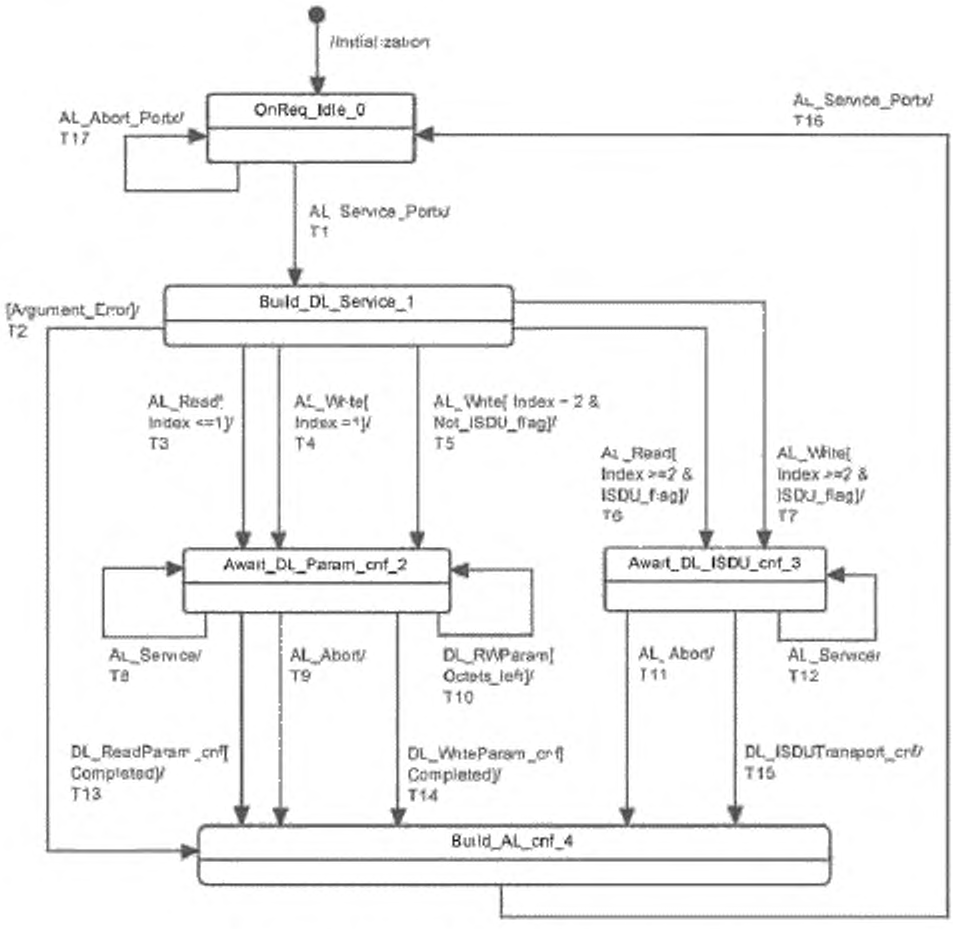


Рисунок 57 — Конечная машина OD AL Ведущего узла

В таблице 72 показаны состояния и переходы конечной машины OD AL Ведущего узла.

Таблица 72 — Состояния и переходы конечной машины OD AL Ведущего узла

Имя состояния		Описание состояния	
OnReq_Idle_0		В данном состоянии могут быть приняты вызовы услуг AL из приложений Ведущего узла или из обработчика порта x SM (см. таблицу 55)	
Build_DL_Service_1		В данном состоянии проверяются вызовы услуг AL и создаются соответствующие услуги DL в определенных состояниях. В случае ошибки в аргументах услуги AL создается и возвращается отрицательное подтверждение	
Await_DL_Param_cnf_2		В данном состоянии услуга AL преобразуется в последовательность требуемого числа вызовов услуг DL_ReadParam или DL_WriteParam (доступ к страницам Непосредственных параметров, см. канал связи страниц на рисунке 6). Все происходящие асинхронно вызовы услуг AL, за исключением услуги AL_Abort, отвергаются (см. 3.3.7)	
Await_DL_ISDU_cnf_3		В данном состоянии вызов услуги AL преобразуется в вызов услуги DL_ISDUTransport (см. канал связи ISDU на рисунке 6). Все происходящие асинхронно вызовы услуг AL, за исключением услуги AL_Abort, отвергаются (см. 3.3.7)	
Build_AL_cnf_4		В данном состоянии создается подтверждение услуги AL в зависимости от ошибки аргумента, подтверждение услуги DL или вызов услуги AL_Abort	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	Запоминание номера порта «Portx»
T2	1	4	Подготовка отрицательного подтверждения услуги AL
T3	1	2	Подготовка вызова услуги DL_ReadParam для значения Индекса 0 или 1
T4	1	2	Подготовка вызова услуги DL_WriteParam для значения Индекса 1
T5	1	2	Подготовка вызова услуги DL_WriteParam для Индекса 2, если Устройство не поддерживает ISDU
T6	1	3	Подготовка вызова услуги DL_ISDUTransport(read)
T7	1	3	Подготовка вызова услуги DL_ISDUTransport(write)
T8	2	2	Возврат отрицательного подтверждения услуги AL для асинхронного вызова данной услуги
T9	2	4	Запрещены действия всех текущих услуг DL, и подготавливаются отрицательные подтверждения услуг AL
T10	2	2	Вызов следующей услуги DL_ReadParam или DL_WriteParam, если не все OD переданы
T11	3	4	Запрещены действия всех текущих услуг DL, и подготавливаются отрицательные подтверждения услуг AL
T12	3	3	Возврат отрицательного подтверждения услуги AL для асинхронного вызова данной услуги
T13	2	4	Подготовка положительного подтверждения услуги AL
T14	2	4	Подготовка положительного подтверждения услуги AL
T15	3	4	Подготовка положительного подтверждения услуги AL
T16	4	0	Возврат положительного подтверждения услуги AL с номером порта «Portx»
T17	0	0	Возврат отрицательного подтверждения услуги AL с номером порта «Portx»
Внутренние элементы		Тип	Определение
Argument_Error		Bool	Недопустимые значения в теле услуги, например, «Номер порта или Индекс вне допустимого диапазона»
DL_RWParam		Метка	DL_RWParam: возможные значения DL_WriteParam_cnf или DL_ReadParam_cnf
Completed		Bool	Не осталось OD для передачи
Octets_left		Bool	Имеются OD для передачи
Portx		Переменная	Переменная тела услуги, указывающая номер порта

Окончание таблицы 72

Внутренние элементы	Тип	Определение
ISDU_Flag	Bool	Устройство поддерживает ISDU
AL_Service	Метка	Термин AL_Service означает любую услугу AL из таблицы 59, связанную с OD

8.3.2.2 Конечная машина Данных запроса прикладного уровня Устройства

На рисунке 58 показана конечная машина для обработки OD AL на Устройстве.

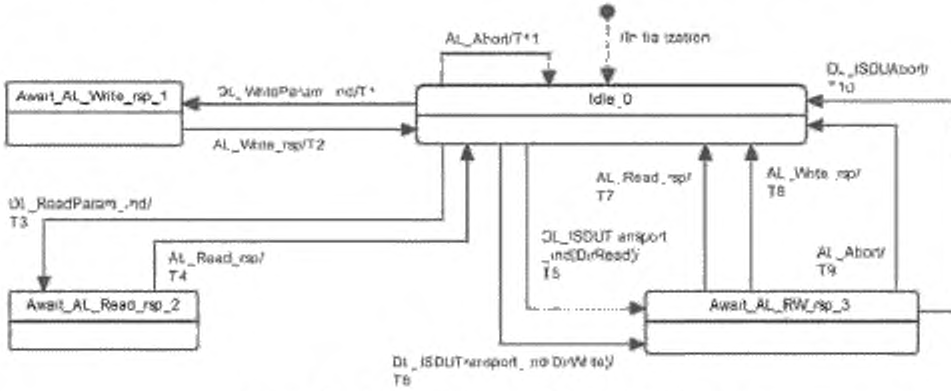


Рисунок 58 — Конечная машина OD AL на Устройстве

В таблице 73 показаны состояния и переходы конечной машины OD AL на Устройстве.

Таблица 73 — Состояния и переходы конечной машины OD AL на Устройстве

Имя состояния		Описание состояния	
Idle_0		AL Устройства ожидает подчиненные вызовы услуг DL, запущенных сообщениями Ведущего узла	
Await_AL_Write_rsp_1		AL ожидает ответа от специальных технических приложений (доступ на запись к странице Непосредственных параметров)	
Await_AL_Read_rsp_2		AL ожидает ответа от специальных технических приложений (доступ на чтение к странице Непосредственных параметров)	
Await_AL_RW_rsp_3		AL ожидает ответа от специальных технических приложений (доступ на чтение или запись через ISDU)	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	Вызов AL_Write
T2	1	0	Вызов DL_WriteParam (номер параметра от 16 до 31)
T3	0	2	Вызов AL_Read
T4	2	0	Вызов DL_ReadParam (номер параметра от 0 до 31)
T5	0	3	Вызов AL_Read
T6	0	3	Вызов AL_Write
T7	3	0	Вызов DL_ ISDUTransport(read)
T8	3	0	Вызов DL_ ISDUTransport(write)
T9	3	0	Текущие услуги AL_Read или AL_Write отменяются при этом асинхронном вызове услуги AL_Abort. Возвращает отрицательный результат услуги DL_ISDUTransport (см. 3.3.7)
T10	3	0	Текущее ожидание AL_Read или AL_Write прекращается
T11	0	0	Текущая услуга DL_ ISDUTransport прекращается. Все OD обнуляются

Окончание таблицы 73

Внутренние элементы	Тип	Определение
DirRead	Bool	Направление доступа: Вызов DL_ ISDUTransport(read) приводит к вызову AL_Read
DirWrite	Bool	Направление доступа: Вызов DL_ ISDUTransport(write) приводит к вызову AL_Write

8.3.2.3 Диаграммы последовательностей для Данных запроса

На рисунках 59—61 продемонстрированы все взаимодействия между Ведущим узлом и Устройством для определенных сценариев использования обмена OD.

На рисунке 59 показаны два примера обмена OD. Для значений Индекса больше 1 обмены выполняются с помощью ISDU и соответствующих услуг DL (канал связи ISDU в соответствии с рисунком 6). Доступ к страницам 0 и 1 Непосредственных параметров применяет различные услуги DL (канал связи страниц в соответствии с рисунком 6).

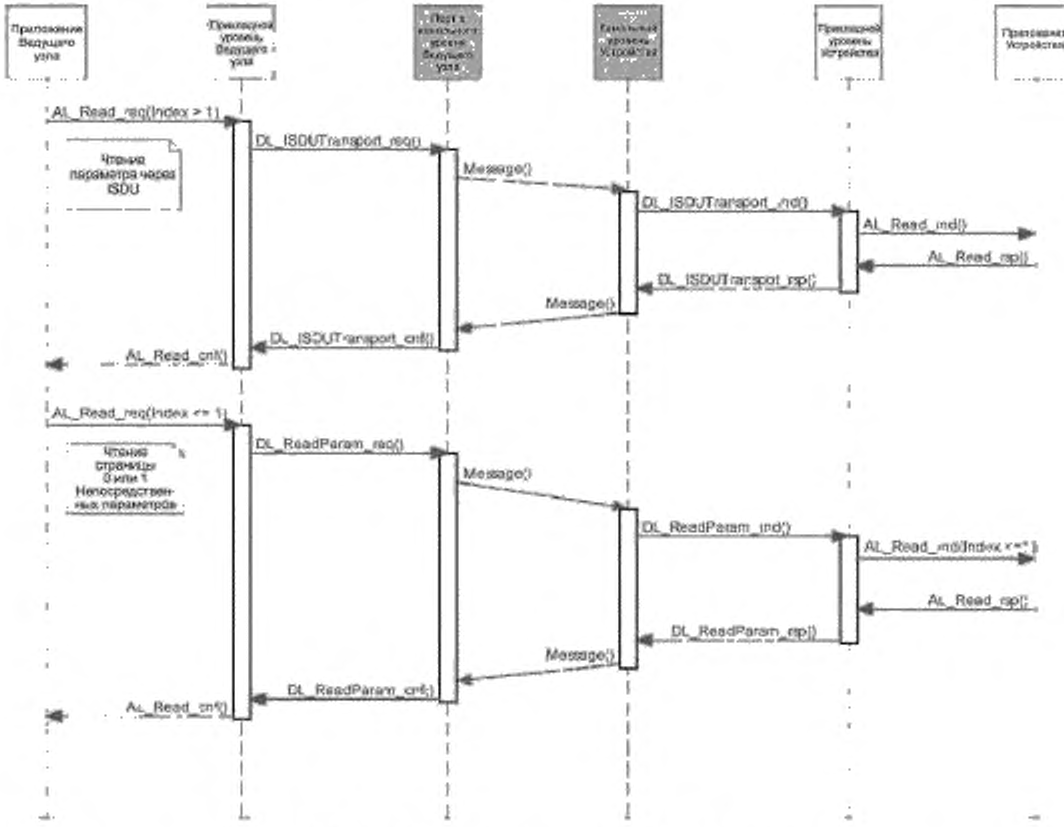


Рисунок 59 — Диаграмма последовательностей при передаче OD

На рисунке 60 показано поведение обмена OD в случае ошибки, такой как недоступность затребованного Индекса (см. таблицу С.1).

Другая возможная ошибка происходит, когда приложение Ведущего узла (шлюз) пытается считать Индекс больший, чем 1 из Устройства, которое не поддерживает ISDU. AL Ведущего узла должен немедленно отреагировать сообщением отсутствия поддержки ISDU «NO_ISDU_SUPPORTED» (ISDU не поддерживается), поскольку характеристики Устройства собираются во время запуска страницы 1 Непосредственных параметров через параметр свойств М-последовательности «M-sequence Capability» Свойства М-последовательности) (см. таблицу В.1).

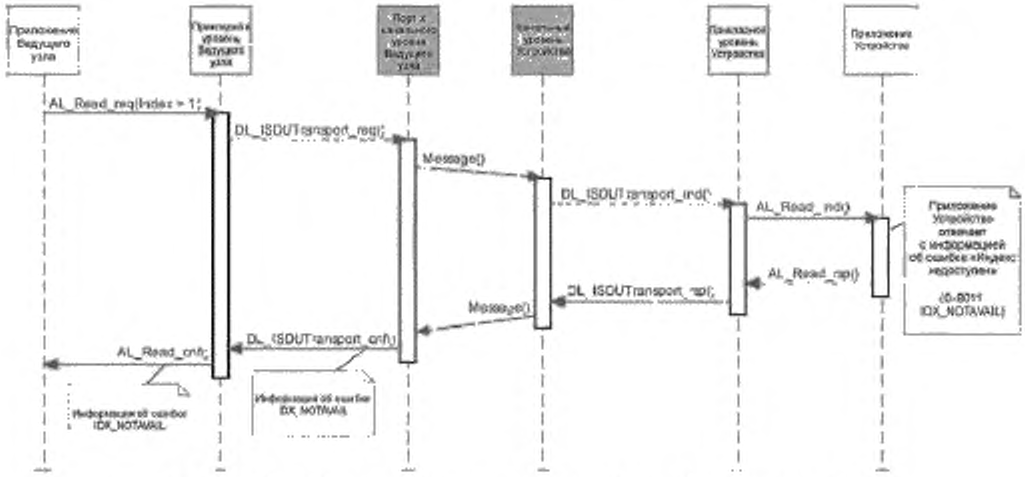


Рисунок 60 — Диаграмма последовательностей для OD в случае ошибок

На рисунке 61 показано поведение обмена OD в случае превышения лимита времени ISDU (5500 мс). Устройство будет реагировать в течение промежутка «ISDU acknowledgement time» (Время подтверждения ISDU) (см. 10.7.5).

Примечание — См. таблицу 97 с параметрами системы, такими как ISDU acknow_ledgement time (Время подтверждения ISDU).

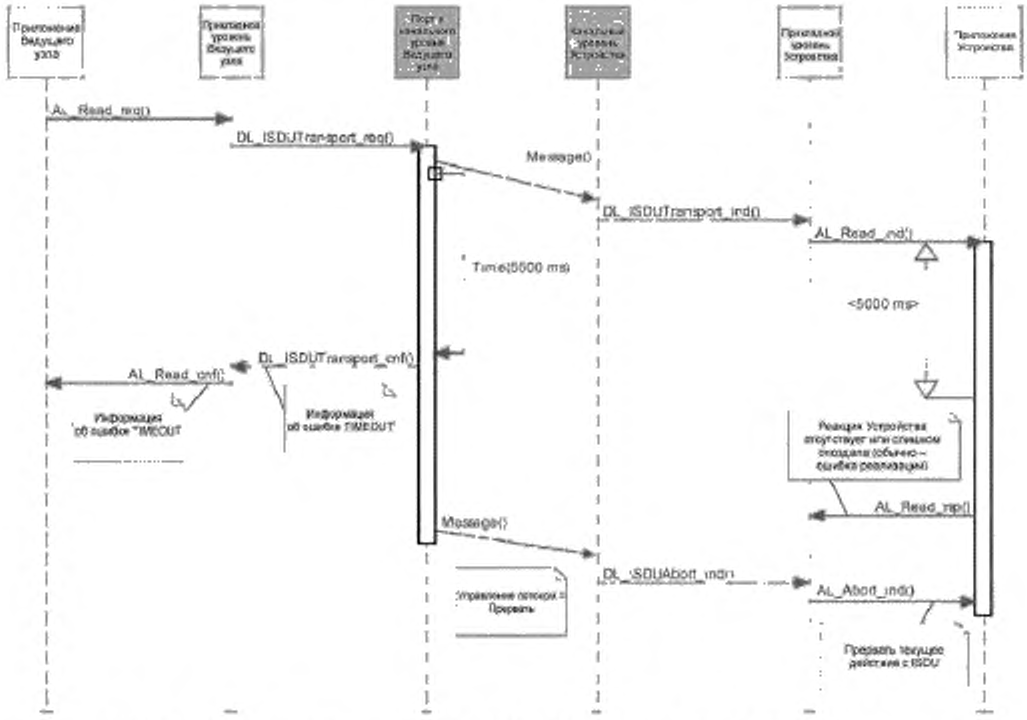


Рисунок 61 — Диаграмма последовательностей для OD в случае превышения лимита времени

8.3.3 Обработка Событий

8.3.3.1 Конечная машина Событий прикладного уровня Ведущего узла

На рисунке 62 показана конечная машина Событий AL на Ведущем узле.

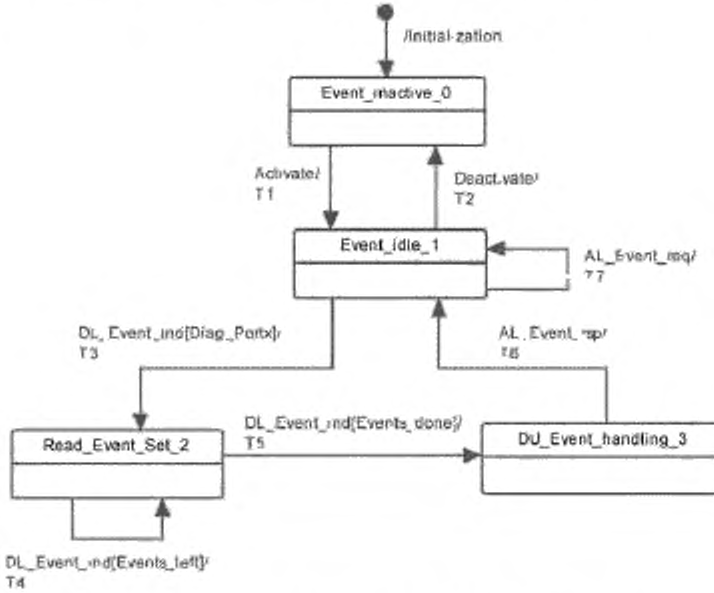


Рисунок 62 — Конечная машина Событий AL на Ведущем узле

В таблице 74 устанавливаются состояния и переходы конечной машины Событий AL Ведущего узла.

Таблица 74 — Состояния и переходы конечной машины Событий AL на Ведущем узле

Имя состояния		Описание состояния	
Event_inactive_0		Обработка Событий AL на Ведущем узле неактивна	
Event_idle_1		AL Ведущего узла готов принимать События канального уровня DL_Events (диагностическую информацию) из DL	
Read_Event_Set_2		AL Ведущего узла получил услугу DL_Event_ind с диагностической информацией. После первой услуги DL_Event.ind AL собирает полный набор (от 1 до 6) событий DL DL_Events от текущего триггера событий EventTrigger (см. 11.5)	
DU_Event_handling_3		AL Ведущего узла остается в этом положении до тех пор, пока Блок диагностики (см. 11.5) не подтвердит информацию о событиях AL AL_Event.ind	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	0	—
T3	1	2	—
T4	2	2	—
T5	2	3	AL_Event.ind
T6	3	1	DL_EventConf.req
T7	1	1	AL_Event.ind

Окончание таблицы 74

Внутренние элементы	Тип	Определение
Diag_Portx	Bool	Набор событий содержит подробную диагностическую информацию
Events_done	Bool	Набор событий обработан
Events_left	Bool	Набор событий еще не полон

8.3.3.2 Конечная машина Событий прикладного уровня на Устройстве

На рисунке 63 показана конечная машина Событий AL на Устройстве.

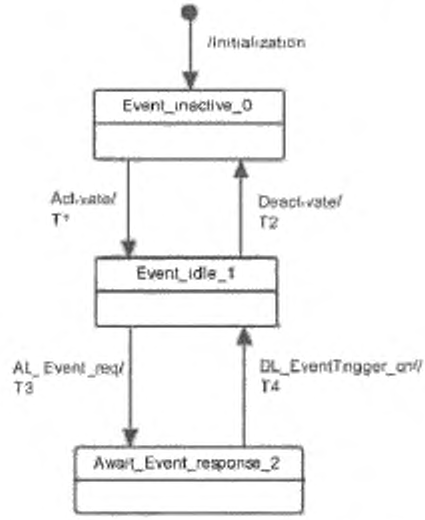


Рисунок 63 — Конечная машина Событий прикладного уровня на Устройстве

В таблице 75 устанавливаются состояния и переходы конечной машины Событий AL на Устройстве.

Таблица 75 — Состояния и переходы конечной машины Событий AL на Устройстве

Имя состояния		Описание состояния	
Event_inactive_0		Обработка Событий AL на Устройстве неактивна	
Event_idle_1		AL Устройства готов к приему Событий AL AL_Events (диагностической информации) от приложений специального технического Устройства для дальнейшей передачи на DL. В это время приложения Устройства могут создавать новые События	
Await_event_response_2		AL Устройства распространил События AL AL_Event с диагностической информацией и ожидает подтверждения триггера событий DL. AL Устройства не будет принимать никаких Событий AL AL_Event в Данный период времени	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	0	—
T3	1	2	Запрос События AL возбуждает Событие DL и вызывает соответствующую услугу DL_EventTrigger DL. Событие DL передает диагностическую информацию от AL на DL. Услуга DL_EventTrigger устанавливает флаг События во время циклического обмена данными (см. А.1.5)
T4	2	1	Подтверждение DL_EventTrigger порождает подтверждение События AL

Окончание таблицы 75

Внутренние элементы	Тип	Определение
<отсутствуют>		

8.3.3.3 График единичного События

На рисунке 64 показано, как обрабатывается единичное Событие из Устройства в соответствии с подходящими конечными машинами.

- Приложение Устройства создает запрос События (Шаг 1), который передается с AL на DL и буферизуется в памяти Событий (см. таблицу 56).
- AL Устройства активирует услугу EventTrigger для возбуждения флага События, что заставляет Ведущий узел считать Событие из памяти Событий.
- Затем Ведущий узел передает данное Событие приложению шлюза (Шаг 2) и ожидает подтверждения События.
- После получения подтверждения о Событии (Шаг 3) Устройство оповещается об этом через запись в код состояния StatusCode (Шаг 4).
- Устройство подтверждает первоначальный запрос События своему приложению (Шаг 5), которое может теперь инициировать новый запрос События.

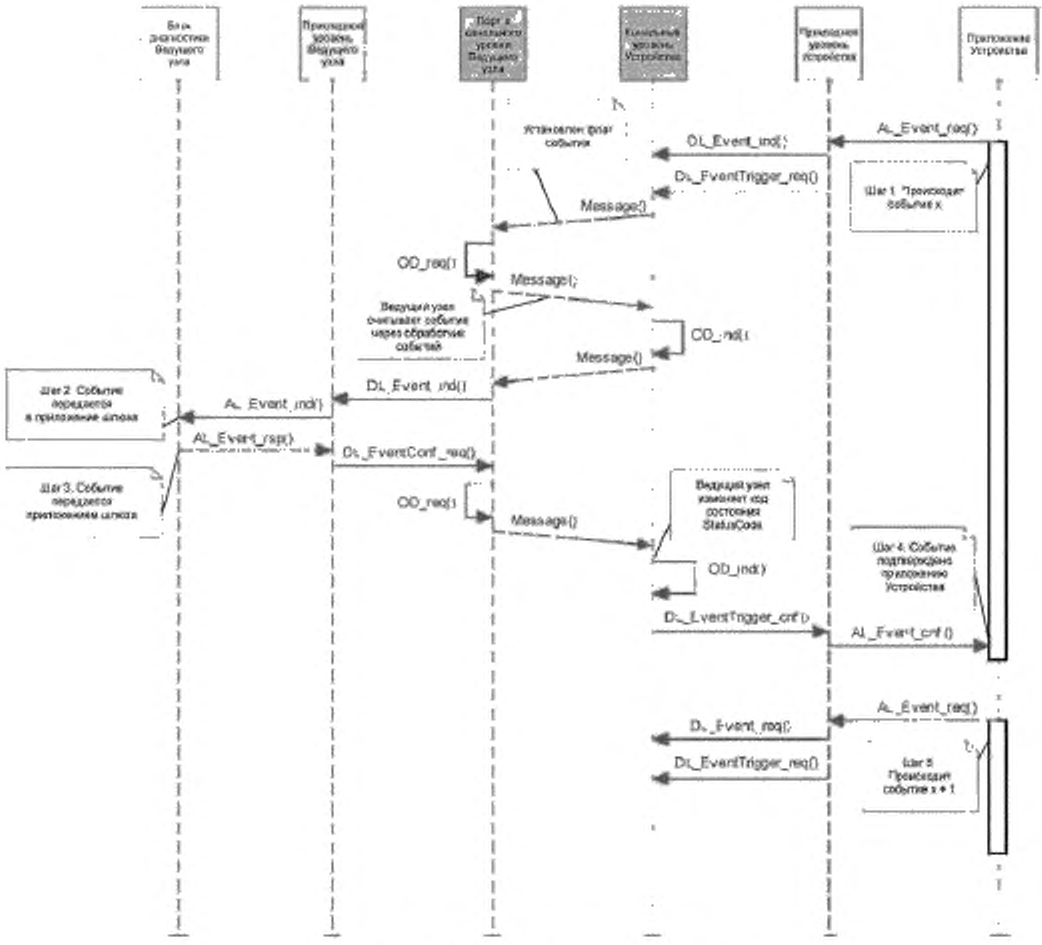


Рисунок 64 — График одиночного События

8.3.3.4 Передача множественных Событий (только устаревшие Устройства)

Кроме метода, установленного в 8.3.3.3, где одиночное Событие передается через уровни и подтверждается приложением шлюза, все Ведущие узлы будут поддерживать так называемую передачу множественных Событий, которая позволяет передавать одновременно до шести Событий. AL Ведущего узла передает набор Событий как отдельный диагностический индикатор в приложение шлюза и возвращает одиночное подтверждение для всего набора в приложение устаревшего Устройства.

Рисунок 64 можно также применять для иллюстрации передачи множественных Событий, за исключение того, что такая передача применяет один индикатор События DL для каждой записи памяти Событий и один индикатор События AL для всего набора Событий.

Один запрос AL_Event.req содержит до шести Событий, и один индикатор AL_Event.ind может содержать до шести ждущих Событий. Услуги AL_Event.rsp и AL_Event.cnf ссылаются на полный указанный набор Событий.

8.3.4 Циклы Данных процесса

На рисунках 65 и 66 показаны полные взаимодействия между Ведущим узлом и Устройством для сценариев использования выходных и входных PD.

На рисунке 65 показано, как услуги AL и DL Ведущего узла и Устройства вовлечены в циклический обмен выходными OD. Приложение Устройства может собирать текущие значения выходных PD, применяя услугу AL_GetOutput.

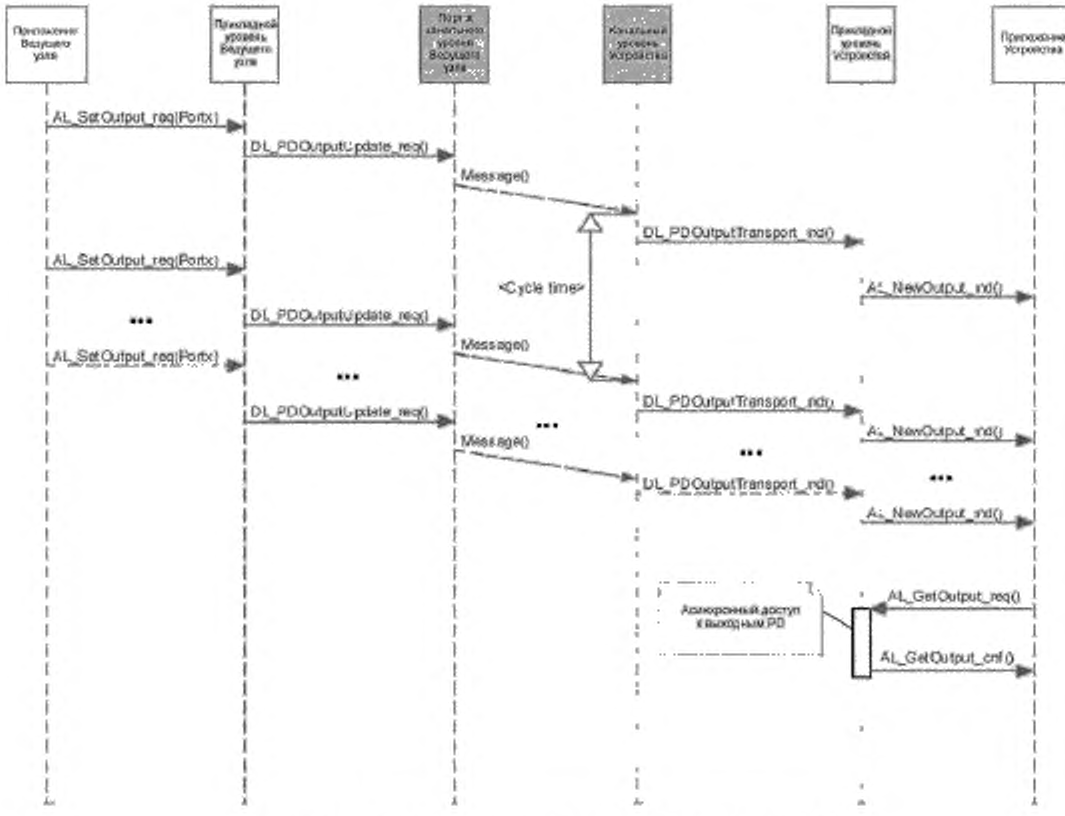


Рисунок 65 — Диаграмма последовательностей для выходных PD

На рисунке 66 показано, как услуги AL и DL Ведущего узла и Устройства вовлечены в циклический обмен входными OD. Приложение Ведущего узла может собирать текущие значения входных PD, применяя услугу AL_GetInput.

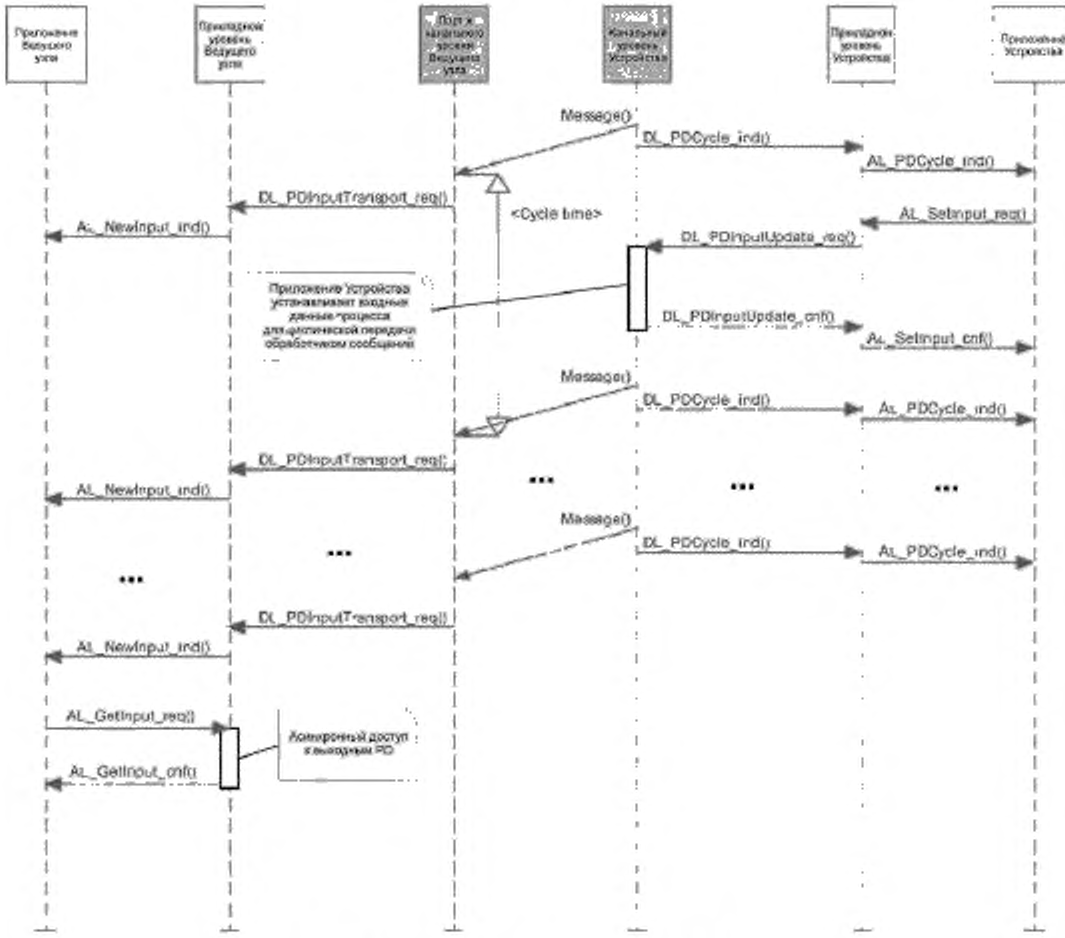


Рисунок 66 — Диаграмма последовательностей для входных PD

9 Модуль управления системой (SM)

9.1 Общие положения

SM SDCI несет ответственность за координированный запуск портов в Ведущем узле и соответствующие операции с подключенными Устройствами.

Разница между SM на Ведущем узле и на Устройстве более значительна, чем разница между другими уровнями. Поэтому структура раздела 9 различает услуги и протоколы Ведущего узла и Устройства.

9.2 Модуль управления системой на Ведущем узле

9.2.1 Обзор

Услуги SM на Ведущем узле применяются для настройки портов Ведущего узла и системы для всех возможных режимов работы.

SM на Ведущем узле регулирует порты через:

- установки требуемой версии протокола связи;
- проверку совместимости Устройства (характеристики реального Устройства должны соответствовать ожидаемым значениям);

- регулировки соответствующих типов М-последовательностей Ведущего узла и времен цикла Ведущего узла.

Для этого применяются услуги, показанные на рисунке 67:

- услуга SM_SetPortConfig передает требуемые параметры Устройства (данные конфигурации) от СМ к SM. После этого порт запускается неявным образом;

- услуга SM_PortMode сообщает положительные результаты о настройке порта обратно в СМ в случае правильной настройки и проверки порта. Она сообщает об отрицательных результатах обратно в СМ через соответствующие «ошибки» в случае несоответствия версий или несовместимых Устройств;

- услуга SM_GetPortConfig считывает фактические и эффективные параметры;

- услуга SM_Operate переключает порты в режим «OPERATE».

На рисунке 67 представлен обзор структуры и услуг управления системой на Ведущем узле.

SM на Ведущем узле требует одной услуги AL (AL_Read) для сбора данных (идентификационных параметров) из специальных Индексов для проверки.

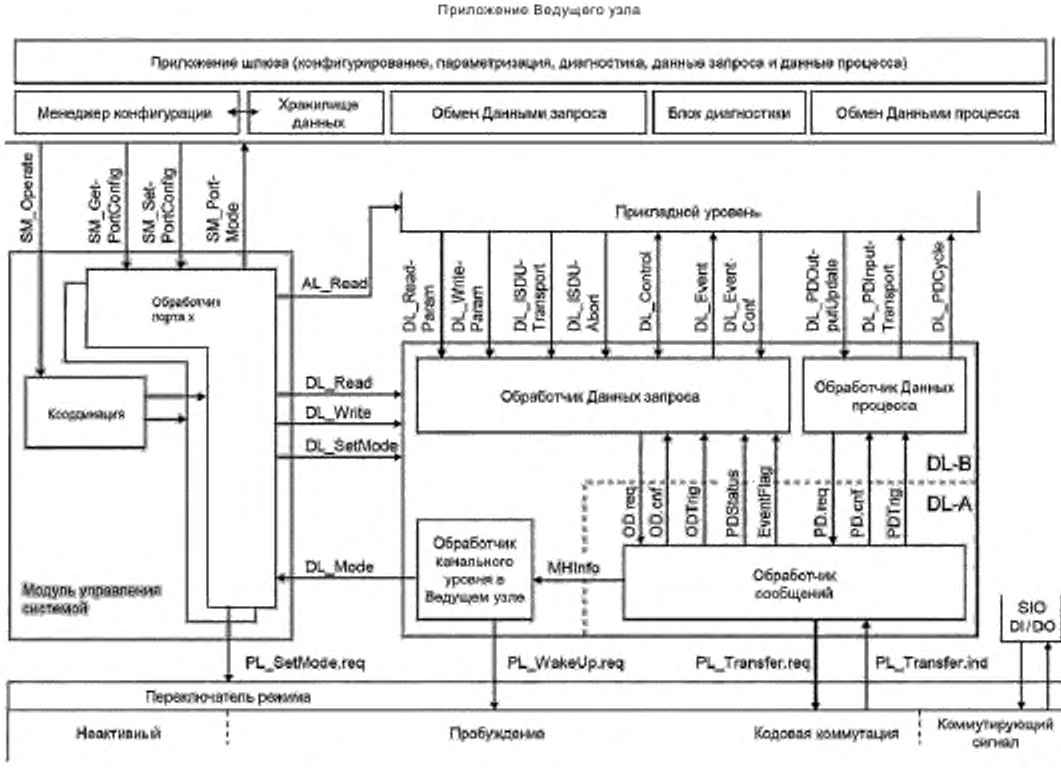


Рисунок 67 — Структура и услуги управления системой на Ведущем узле

На рисунке 68 показаны взаимодействия между уровнями приложения Ведущего узла (Master App), CM, SM, DL и AL для сценария запуска конкретного порта.

Данный конкретный сценарий использования характеризуется следующими утверждениями:

- устройство для доступной конфигурации подключено, и проверка оказалась успешной;

- устройство применяет правильную версию протокола в соответствии с настоящей спецификацией;

- сконфигурированный уровень проверки InspectionLevel — «type compatible» (совместимый по типу) (заводской номер SerialNumber считан с устройства, но не проверялся).

Пунктирные стрелки на рисунке 68 представляют ответные сигналы услуг.

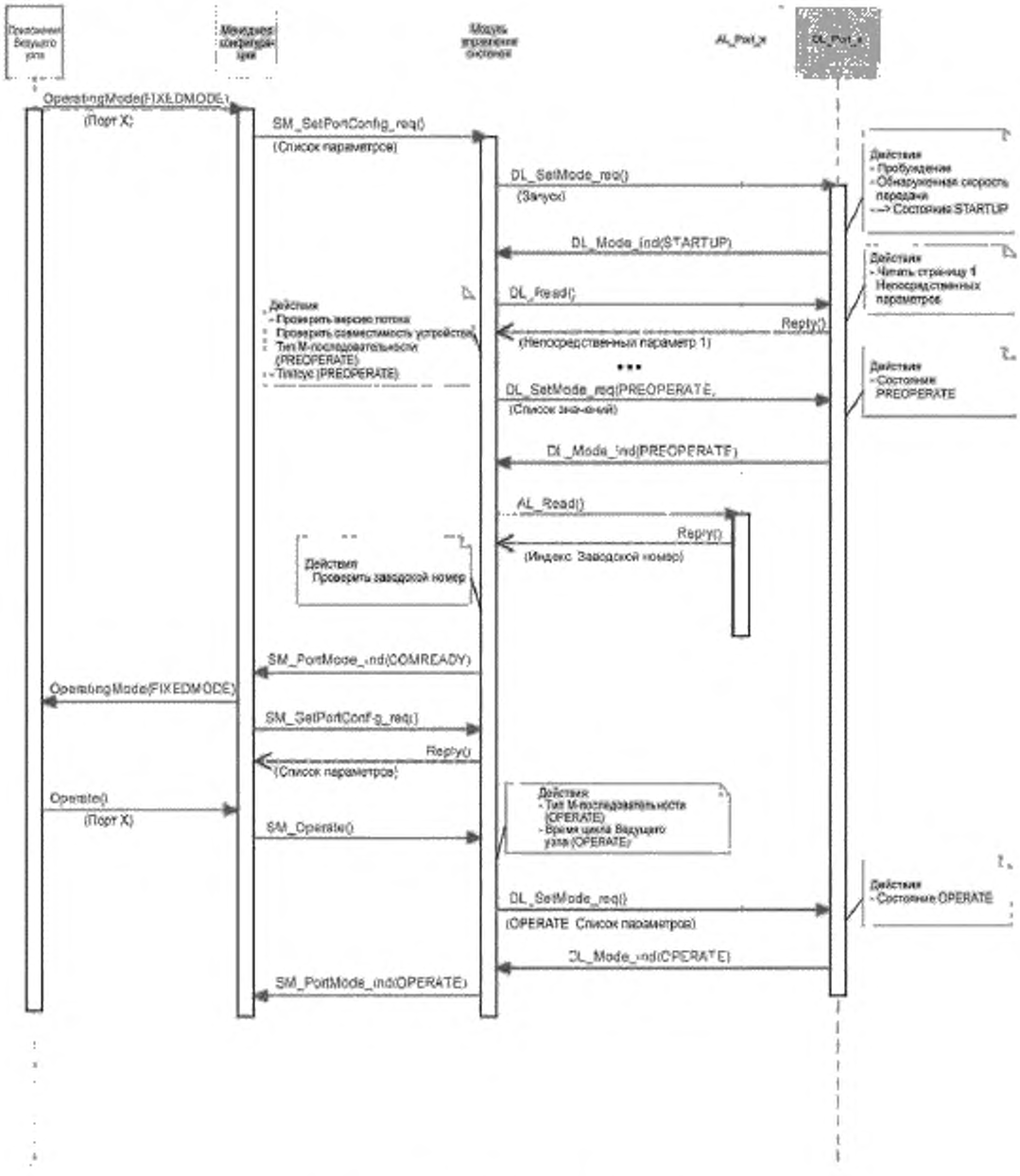


Рисунок 68 — Схема последовательности
сценария использования «Настройка порта»

9.2.2 Услуги модуля управления системой на Ведущем узле

9.2.2.1 Обзор

SM предоставляет услуги Ведущего узла пользователю через интерфейс более высокого уровня. В таблице 76 приведены средства Ведущего узла для его роли инициатора или получателя отдельных услуг управления системой.

Таблица 76 — Услуги SM на Ведущем узле

Имя услуги	Ведущий узел
SM_SetPortConfig	R
SM_GetPortConfig	R
SM_PortMode	I
SM_Operate	R
Обозначения (см. 3.3.4): I — Инициатор услуги; R — Приемник (Ответчик) услуги.	

9.2.2.2 Услуга SM_SetPortConfig

Услуга SM_SetPortConfig применяется для установки требуемой конфигурации Устройства. Параметры сервисных примитивов приведены в таблице 77.

Таблица 77 — Услуга SM_SetPortConfig

Имя параметра	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
PortNumber		M
Result (–)		S
PortNumber		M
ErrorInfo		M

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

ParameterList (СписокПараметров)
Данный параметр содержит параметры сконфигурированного порта и параметры Устройства, подключенного к порту Ведущего узла.

Тип параметра: Запись.

Элементы записи:

PortNumber (НомерПорта)
Данный параметр содержит номер порта.

ConfiguredCycleTime (СконфигурированноеВремяЦикла)
Данный параметр содержит требуемое время цикла для режима OPERATE.

Допустимые значения:
0 (свободный ход);
Время (см. таблицу B.3).

TargetMode (ЦелевойРежим)
Данный параметр указывает требуемый рабочий режим порта.

Допустимые значения: INACTIVE, DI, DO, CFGCOM, AUTOCOM (см. таблицу 79).

ConfiguredBaudrate (СконфигурированнаяСкоростьБод)
Данный параметр указывает требуемую скорость передачи данных в бодах.

Допустимые значения:
AUTO (Ведущий узел принимает скорость передачи, обнаруженную во время установки связи ESTABLISHCOM);
COM1 (скорость передачи порта COM1);
COM2 (скорость передачи порта COM2);
COM3 (скорость передачи порта COM3).

ConfiguredRevisionID (СконфигурированныйИдРевизии) (CRID)
Длина данных: один октет для версии протокола (см. B.1.5).

InspectionLevel (УровеньПроверки)
Допустимые значения: NO_CHECK, TYPE_COMP, IDENTICAL (см. таблицу 78).

ConfiguredVendorID (СконфигурированныйИдИзготовителя) (CVID)
Длина данных: два октета.

Примечание — ИдИзготовителя назначается консорциумом IO-Link.

ConfiguredDeviceID (СконфигурированныйИдУстройства) (CDID)
Длина данных: три октета.

ConfiguredFunctionID (СконфигурированныйИдФункции) (CFID)
Длина данных: два октета.

ConfiguredSerialNumber (СконфигурированныйЗаводскойНомер) (CSN)
Длина данных: до 16 октетов.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга была выполнена успешно.

PortNumber (НомерПорта)
Данный параметр содержит номер порта.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

PortNumber (НомерПорта)
Данный параметр содержит номер порта.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
PARAMETER_CONFLICT (нарушена корректность набора параметров).

В таблице 78 установлено кодирование различных уровней проверки (значения параметра УровеньПроверки) (см. 9.2.3.2 и 11.8.5).

Таблица 78 — Определение параметра УровеньПроверки (IL)

Параметр	УровеньПроверки (IL)		
	NO_CHECK (нет проверки)	TYPE_COMP (сравнение типа)	IDENTICAL (идентичный)
ID Устройства (DID) (совместимый)	—	Да (RDID=CDID)	Да (RDID=CDID)
ID Изготовителя (VID)	—	Да (RDID=CDID)	Да (RDID=CDID)
ЗаводскойНомер (SN)	—	—	Да (RSN = CSN)

В таблице 79 установлено кодирование различных целевых режимов.

Таблица 79 — Определения целевых режимов

Целевой режим	Определение
CFGCOM	Устройство, осуществляющее связь в режиме CFGCOM после успешной проверки
AUTOCOM	Устройство, осуществляющее связь в режиме AUTOCOM без проверки
INACTIVE	Связь отключена, нет цифрового ввода DI, нет цифрового вывода DO
DI	Порт в режиме цифрового ввода DI (SIO)
DO	Порт в режиме цифрового вывода DO (SIO)

CFGCOM — целевой режим, основанный на конфигурации пользователя (например, с помощью IODD) и проверках совместимости RID, VID, DID.

AUTOCOM — целевой режим без конфигурации. Это означает отсутствие проверок CVID и CDID. CRID устанавливается в наивысшую редакцию, поддерживаемую Ведущим узлом. AUTOCOM должен выбираться только вместе с уровнем проверки «NO_CHECK» (см. таблицу 78).

9.2.2.3 Услуга SM_GetPortConfig

Услуга SM_GetPortConfig применяется для получения реальной (действующей) конфигурации Устройства. Параметры сервисных примитивов приведены в таблице 80.

Таблица 80 — Услуга SM_GetPortConfig

Имя параметра	.req	.cnf
Argument	M	
PortNumber	M	
Result (+)		S(=)
ParameterList		M
Result (–)		S(=)
PortNumber		M
ErrorInfo		M

Argument (Аргумент)

Определяемые услугой параметры передаются в данном параметре.

PortNumber (НомерПорта)

Данный параметр содержит номер порта.

Result (+) (ПоложительныйРезультат)

Данный параметр выбора указывает, что запрос услуги был выполнен успешно.

ParameterList (СписокПараметров)

Данный параметр содержит сконфигурированный порт и параметры Устройства, подключенного к порту Ведущего узла.

Тип параметра: Запись.

Элементы записи:

PortNumber (НомерПорта)

Данный параметр содержит номер порта.

TargetMode (ЦелевойРежим)

Данный параметр указывает режим работы.

Допустимые значения: INACTIVE, DI, DO, CFGCOM, AUTOCOM (см. таблицу 79).

RealBaudrate (ФактСкоростьБод)
Данный параметр указывает фактическую скорость передачи данных в бодах.
Допустимые значения:
COM1 (скорость передачи порта COM1);
COM2 (скорость передачи порта COM2);
COM3 (скорость передачи порта COM3).

RealCycleTime (ФактВремяЦикла)
Данный параметр содержит фактическое (действительное) время цикла.

RealRevision (ФактРедакция) (RRID)
Длина данных: один октет для версии протокола (см. В.1.5).

RealVendorID (ФактИдИзготовителя) (RVID)
Длина данных: два октета.

Примечание — Идентификатор изготовителя назначается консорциумом IO-Link.

RealDeviceID (ФактИдУстройства) (RDID)
Длина данных: три октета.

RealFunctionID (ФактИдФункции) (RFID)
Длина данных: два октета.

RealSerialNumber (ФактЗаводскойНомер) (RSN)
Длина данных: до 16 октетов.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

PortNumber (НомерПорта)
Данный параметр содержит номер порта.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
PARAMETER_CONFLICT (нарушена корректность набора параметров).
Если информация недоступна, все параметры будут установлены в «0».

9.2.2.4 Услуга SM_PortMode

Услуга SM_PortMode применяется для указания изменений или отказов режима локальной связи. Об этом будет сообщено приложению Ведущего узла. Параметры сервисных примитивов приведены в таблице 81.

Таблица 81 — Услуга SM_PortMode

Имя параметра	.ind
Argument	M
PortNumber	M
Mode	M

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

PortNumber (НомерПорта)
Данный параметр содержит номер порта.

Mode (Режим)
Допустимые значения:
INACTIVE (связь отключена, нет цифрового ввода, нет цифрового вывода);
DI [порт в режиме цифрового ввода (SIO)];
DO [порт в режиме цифрового вывода (SIO)];
COMREADY (связь установлена, и проверка прошла успешно);
SM_OPERATE (порт готов к обмену PD);
COMLOST (сбой связи, требуется новая процедура пробуждения);
REVISION_FAULT (несовместимая версия протокола);
COMP_FAULT (несовместимое или устаревшее Устройство для уровня проверки);
SERNUM_FAULT (заводской номер не соответствует уровню проверки).

9.2.2.5 SM_Operate

Услуга SM_Operate заставляет SM вычислить времена цикла Ведущего узла, когда они положительно подтверждаются параметром Result (+). Данная услуга эффективна для всех портов. Параметры сервисных примитивов приведены в таблице 82.

Таблица 82 — SM_Operate

Имя параметра	.req	.cnf
Result (+)		S
Result (–)		S
ErrorInfo		M

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
TIMING_CONFLICT (требуемая комбинация времен цикла невозможна для активированных портов).

9.2.3 Протокол управления системой на Ведущем узле

9.2.3.1 Обзор

Вследствие всестороннего конфигурирования, параметризации и рабочих характеристик интерфейса SDCI описание поведения средствами диаграмм состояний становится достаточно сложным. Как и в случае конечных машин DL, в 9.2.3 применяются возможности функциональных блоков с основными конечными машинами.

Для функциональных блоков выполняются методы проверки полной совместимости. Данные методы указываются полями «метода действия» на графах состояний, см., например, рисунок 70.

Соответствующая логика принятия решения демонстрируется посредством диаграмм деятельности (см. рисунки 71, 72, 73 и 76).

9.2.3.2 Конечная машина модуля управления системой на Ведущем узле

На рисунке 69 показана основная конечная машина SM на Ведущем узле. Два функциональных блока для проверки совместимости и заводского номера определены в 9.2.3.3 и 9.2.3.4. В случае нарушения связи SM информируется через услугу DL_Mode (COMLOST). Изменение конфигурации порта можно осуществить только через услугу SM_SetPortConfig. Услуга SM_Operate (доступная на всех портах) не действует ни в каких состояниях, кроме состояния «wait_4».

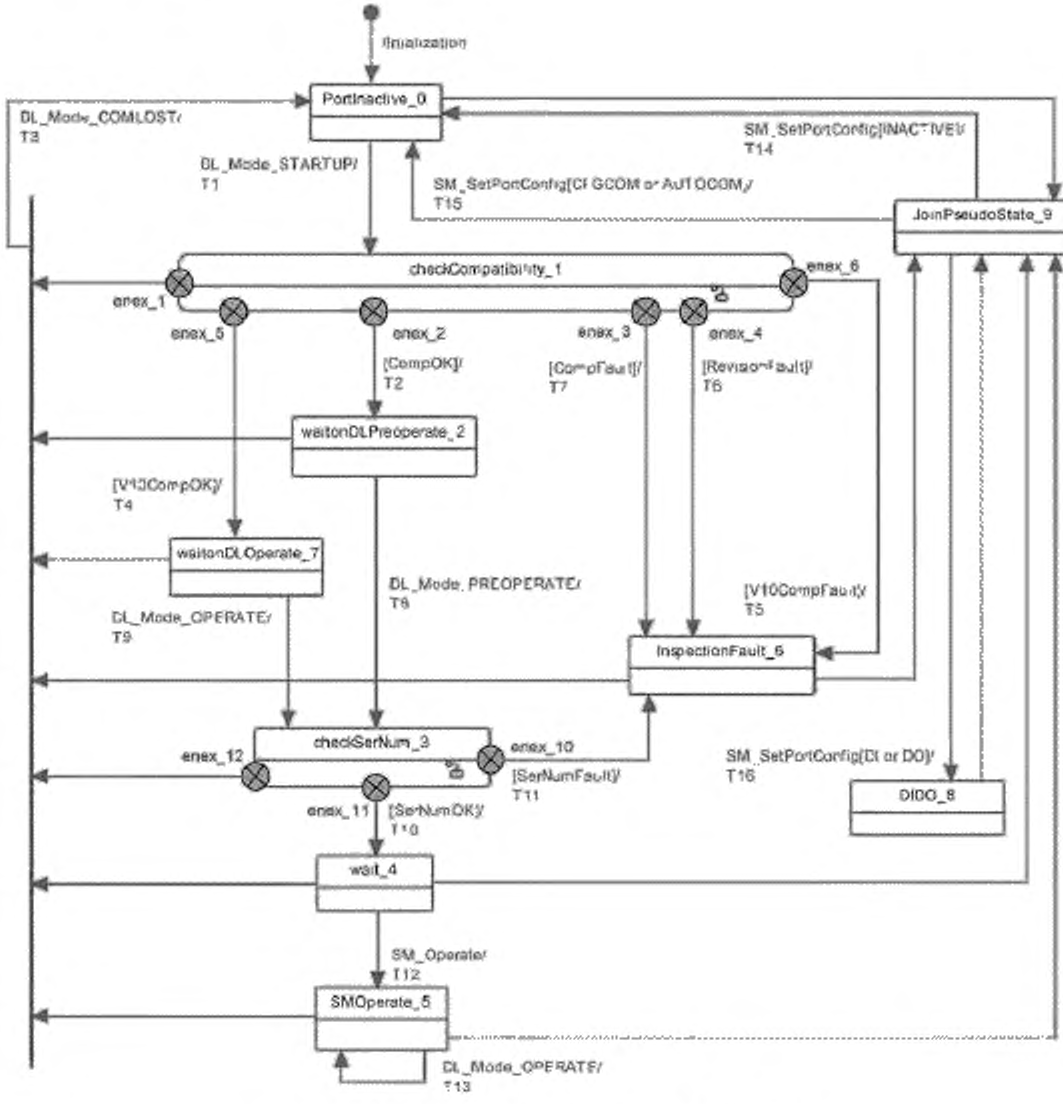


Рисунок 69 — Структура и услуги SM на Ведущем узле

В таблице 83 показаны таблицы перехода состояний обработчика SM на Ведущем узле.

Таблица 83 — Таблицы перехода состояний SM на Ведущем узле

Имя состояния	Описание состояния
PortInactive_0	Нет связи
CheckCompatibility_1	Порт запускается, и проверяются версия и совместимость Устройства. См. рисунок 70
waitonDLPreoperate_2	Ожидается установка состояния PREOPERATE, и запускаются все обработчики OD. Порт готов к связи

Продолжение таблицы 83

Имя состояния		Описание состояния	
CheckSerNum_3		Заводской номер проверяется в зависимости от уровня проверки (IL). См. рисунок 75	
wait_4		Порт готов к связи и ожидает услугу SM_Operate из СМ	
SM Operate_5		Порт находится в состоянии OPERATE и выполняет обмен циклическими PD	
InspectionFault_6		Порт готов к связи. Однако обмен циклическими PD не может выполняться из-за несовместимостей	
waitonDLOperate_7		Ожидание затребованного состояния OPERATE в случае, когда Ведущий узел подключен к устаревшему Устройству. После этого может быть считан заводской номер	
DIDO_8		Порт будет переключен в режим цифрового ввода или цифрового вывода (SIO, нет связи)	
JoinPseudoState_9		Данное псевдосостояние применяется вместо панели соединения универсального языка программирования UML. Оно разрешает выполнение индивидуальных услуг SM_SetPortConfig в зависимости от состояния системы (INACTIVE, CFGCOM, AUTOCOM, DI или DO)	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	CompRetry = 0
T2	1	2	DL_SetMode.req (PREOPERATE, ValueList)
T3	1, 2, 3, 4, 5, 6, 7	0	DL_SetMode.req (INACTIVE) и SM_Mode.ind (COMLOST) из-за отказа связи
T4	1	7	DL_SetMode.req (OPERATE, ValueList)
T5	1	6	SM_PortMode.ind (COMP_FAULT), DL_SetMode.req (OPERATE, ValueList)
T6	1	6	SM_PortMode.ind (REVISION_FAULT), DL_SetMode.req (PREOPERATE, ValueList)
T7	1	6	SM_PortMode.ind (COMP_FAULT), DL_SetMode.req (PREOPERATE, ValueList)
T8	2	3	—
T9	7	3	—
T10	3	4	SM_PortMode.ind (COMREADY)
T11	3	6	SM_PortMode.ind (SERNUM_FAULT)
T12	4	5	DL_SetMode.req (OPERATE, ValueList)
T13	5	5	—
T14	0, 4, 5, 6, 8	0	SM_PortMode.ind (INACTIVE), DL_SetMode.req (INACTIVE)
T15	0, 4, 5, 6, 8	0	DL_SetMode.req (STARTUP, ValueList), PL_SetMode.req (SDCI)
T16	0, 4, 5, 6, 8	8	PL_SetMode.req (SIO), SM_Mode.ind (DI или DO), DL_SetMode.req (INACTIVE)
Внутренние элементы		Тип	Определение
CompOK		Bool	См. рисунок 73
CompFault		Bool	См. рисунок 73, переменная ошибки COMP_FAULT
RevisionFault		Bool	См. рисунок 71, переменная ошибки REVISION_FAULT
SerNumFault		Bool	См. рисунок 76, переменная ошибки SERNUM_FAULT

Окончание таблицы 83

Внутренние элементы	Тип	Определение
SerNumOK	Bool	См. рисунок 76
V10CompFault	Bool	См. рисунок 72, переменная ошибки COMP_FAULT
V10CompOK	Bool	См. рисунок 72
INACTIVE	Переменная	Целевой режим в услуге SM_SetPortConfig
CFGCOM, AUTOCOM	Переменные	Целевые режимы в услуге SM_SetPortConfig

9.2.3.3 Функциональный блок checkCompatibility_1 модуля управления системой на Ведущем узле

На рисунке 70 показан функциональный блок checkCompatibility_1 на Ведущем узле.

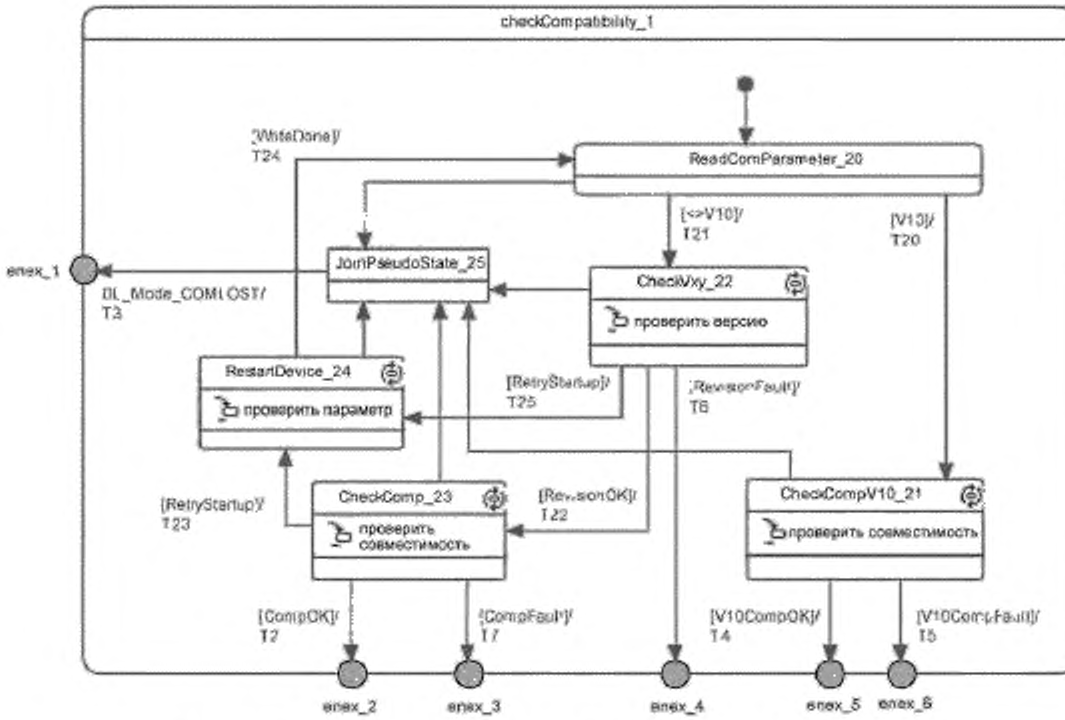


Рисунок 70 — Функциональный блок CheckCompatibility_1 на Ведущем узле

В таблице 84 показаны таблицы перехода состояний функционального блока checkCompatibility_1 на Ведущем узле.

Таблица 84 — Таблицы перехода состояний функционального блока checkCompatibility_1 на Ведущем узле

Имя состояния	Описание состояния
ReadComParameter_20	Получает параметры связи из страницы 1 Непосредственных параметров (от 0x02 до 0x06) через услугу DL_Read (см. таблицу B.1)
CheckCompV10_21	Получает параметры идентификации из страницы 1 Непосредственных параметров (от 0x07 до 0x0D) через услугу DL_Read (см. таблицу B.1). Сконфигурированный уровень проверки (IL) определяет логику принятия решения последующей проверки совместимости «CheckCompV10» с параметрами RVID, RDID и RFID в соответствии с рисунком 72

Окончание таблицы 84

Имя состояния		Описание состояния	
CheckVxy_22		Проверка устанавливает, соответствует ли сконфигурированная версия (CRID) реальной (фактической) версии (RRID) в соответствии с рисунком 71	
CheckComp_23		Получает параметры идентификации из страницы 1 Непосредственных параметров (от 0x07 до 0x0D) через услугу DL_Read (см. таблицу B.1). Сконфигурированный уровень проверки (IL) определяет логику принятия решения последующей проверки совместимости CheckComp в соответствии с рисунком 73	
Restart Device_24		Записывает параметры совместимости сконфигурированной версии протокола (CRID) и сконфигурированного идентификатора Устройства DeviceID (CDID) в Устройство в зависимости от целевого режима связи CFGCOM или AUTOCOM (см. таблицу 79) в соответствии с рисунком 74	
JoinPseudoState_25		Данное псевдосостояние применяется вместо панели соединения универсального языка программирования UML. Средства защиты не применяются	
Переход	Исходное состояние	Конечное состояние	Действие
T20	20	21	—
T21	20	22	DL_Write (0x00, MCmd_MASTERIDENT), см. таблицу B.2
T22	22	23	—
T23	23	24	—
T24	24	20	—
T25	22	24	CompRetry = CompRetry +1
Внутренние элементы		Тип	Определение
CompOK		Bool	См. рисунок 73
CompFault		Bool	См. рисунок 73, переменная ошибки COMP_FAULT
RevisionFault		Bool	См. рисунок 71, переменная ошибки REVISION_FAULT
RevisionOK		Bool	См. рисунок 71
SerNumFault		Bool	См. рисунок 76, переменная ошибки SERNUM_FAULT
SerNumOK		Bool	См. рисунок 76
V10		Bool	Фактическая версия протокола подключенного Устройства — устаревшая версия (V1.0, см. B.1.5)
<>V10		Bool	Фактическая версия протокола подключенного Устройства соответствует настоящему стандарту
V10CompFault		Bool	См. рисунок 72, переменная ошибки COMP_FAULT
V10CompOK		Bool	См. рисунок 72
RetryStartup		Bool	См. рисунок 71 и рисунок 73
CompRetry		Переменная	Внутренний счетчик
WriteDone		Bool	Полное завершение последовательности услуг повторного запуска
M C m d_XXXXXXX		Call	См. таблицу 43

Некоторые состояния содержат сложную логику для проверок совместимости и допустимости. Это демонстрируется на рисунках 71—74.

На рисунке 71 показана логика принятия решения для проверки версии протокола в состоянии CheckVxy. В случае сконфигурированных Устройств применяются следующие правила: если сконфигурированная версия (CRID) и фактическая версия (RRID) не совпадают, на Устройство будет передана версия CRID. Если устройство не соглашается, то Ведущий узел возвращает индикацию через услугу SM_Mode с параметром REV_FAULT.

В случае несконфигурированных Устройств будет применяться режим AUTOCOM. Аббревиатуры имен параметров описываются в 9.2.2.2 и 9.2.2.3.

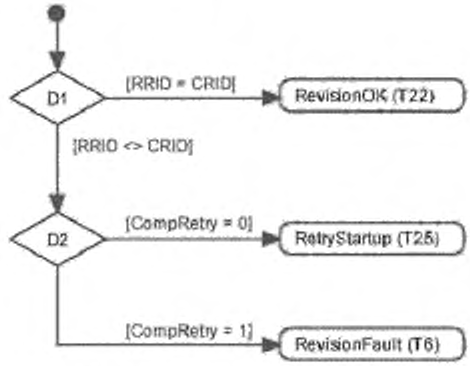


Рисунок 71 — Действия для состояния CheckVxy

На рисунке 72 показана логика принятия решения для проверки совместимости устаревшего Устройства в состоянии CheckV10.

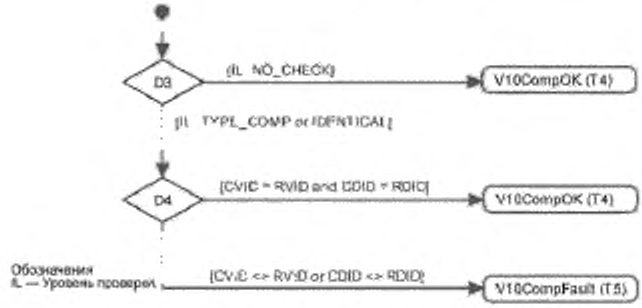


Рисунок 72 — Действия для состояния CheckV10

На рисунке 73 показана логика принятия решения для проверки совместимости в состоянии CheckComp.

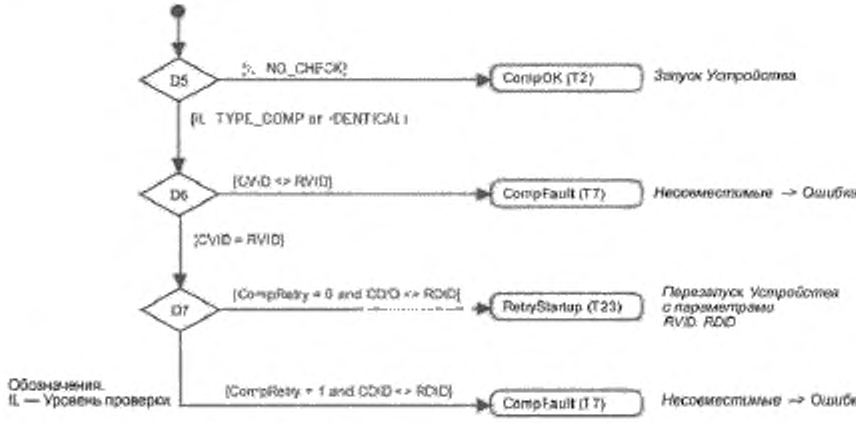


Рисунок 73 — Действия для состояния CheckComp

На рисунке 74 показаны действия (запись параметра) в состоянии RestartDevice.

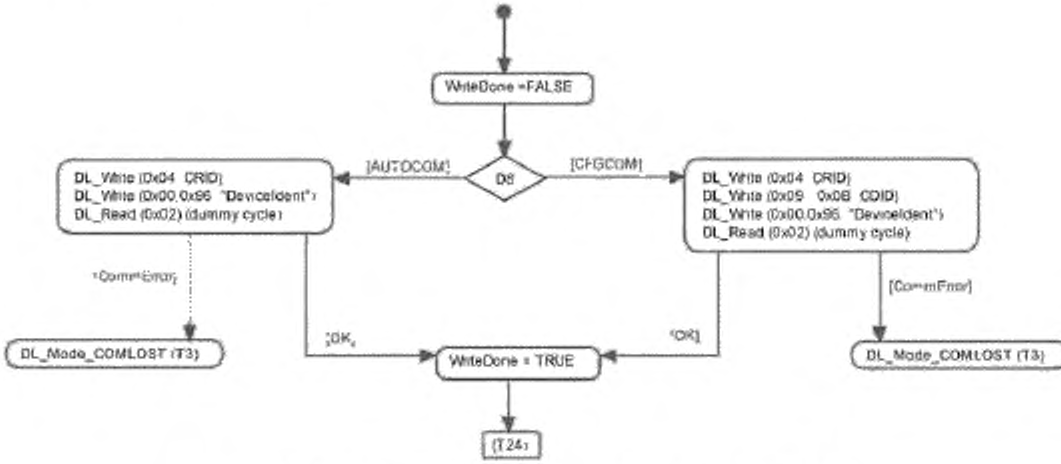


Рисунок 74 — Действия (запись параметра) в состоянии RestartDevice

9.2.3.4 Функциональный блок проверки заводского номера «Check serial number» модуля управления системой на Ведущем узле

На рисунке 75 показан функциональный блок checkSerNum_3 SM на Ведущем узле. Данная проверка является обязательной.

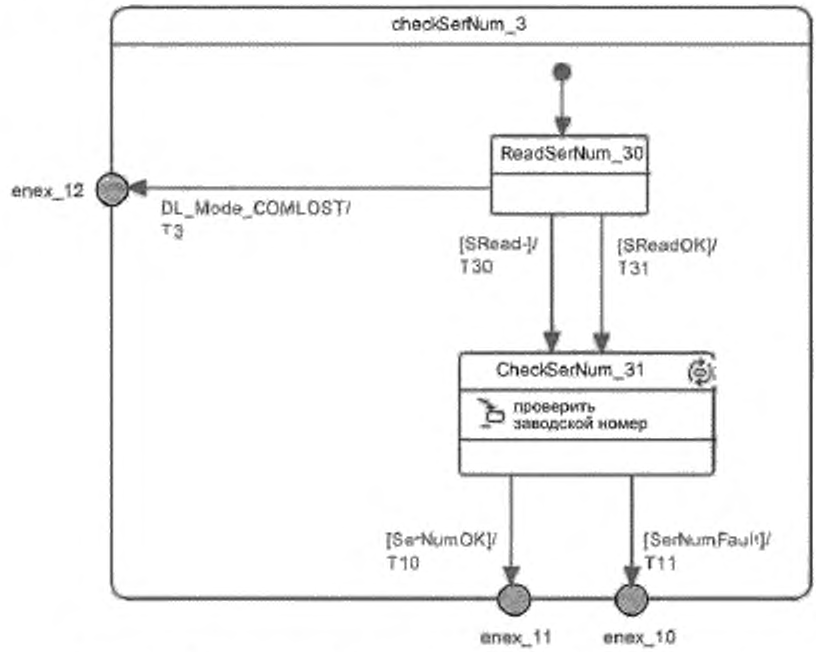


Рисунок 75 — Функциональный блок CheckSerNum_3 SM на Ведущем узле

В таблице 85 показаны таблицы перехода состояний функционального блока CheckSerNum_3 на Ведущем узле.

Таблица 85 — Таблицы перехода состояний функционального блока CheckSerNum_3 на Ведущем узле

Имя состояния		Описание состояния	
ReadSerNum_30		Получает заводской номер из Устройства через вызов услуги AL_Read.req (Index:0x0015). Положительный ответ (AL_Read(+)) приводит к SReadOK = true. Отрицательный ответ (AL_Read(+)) приводит к SRead = true	
CheckSerNum_31		Сконфигурированный (CSN) и фактический (RSN) заводские номера проверяются в зависимости от уровня проверки (IL) в соответствии с рисунком 76	
Переход	Исходное состояние	Конечное состояние	Действие
T30	40	41	
T31	40	41	
Внутренние элементы		Тип	Определение
SRead-		Bool	Отрицательный результат вызова услуги AL_Read (Index 0x0015)
SReadOK		Bool	Заводской номер считан правильно
SERNumOK		Bool	См. рисунок 76
SERNumFault		Bool	См. рисунок 76

На рисунке 76 показана логика принятия решения (действия) для состояния CheckSerNum_3.

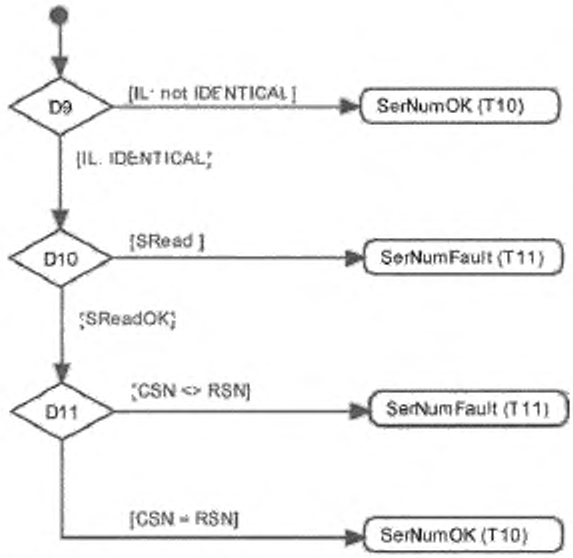


Рисунок 76 — Действия (проверка заводского номера) для состояния CheckSerNum_3

9.2.3.5 Правила использования типов М-последовательностей

SM отвечает за установку правильных типов М-последовательностей. Это происходит после действий по проверке совместимости (переход в состояние PREOPERATE) и перед переходом в состояние OPERATE.

В разных рабочих состояниях будут применяться различные типы М-последовательностей (см. А.2.6). Например, при переключении в состояние OPERATE будет применяться тип М-последовательности, соответствующий циклической операции. Тип М-последовательности, который будет применяться в рабочем состоянии OPERATE, определяется размером входных и выходных

PD. Доступные типы M-последовательностей в трех режимах STARTUP, PREOPERATE и OPERATE и соответствующее кодирование параметра M-sequence Capability устанавливаются в A.2.6. Форматы входных и выходных данных будут получаться из подключенного Устройства, чтобы подстраиваться к типу M-последовательности. Ведущий узел должен обязательно реализовывать все определенные типы M-последовательностей, установленные в A.2.6.

9.3 Модуль управления системой на Устройстве

9.3.1 Обзор

На рисунке 77 представлен обзор структуры и услуг SM на Устройстве.

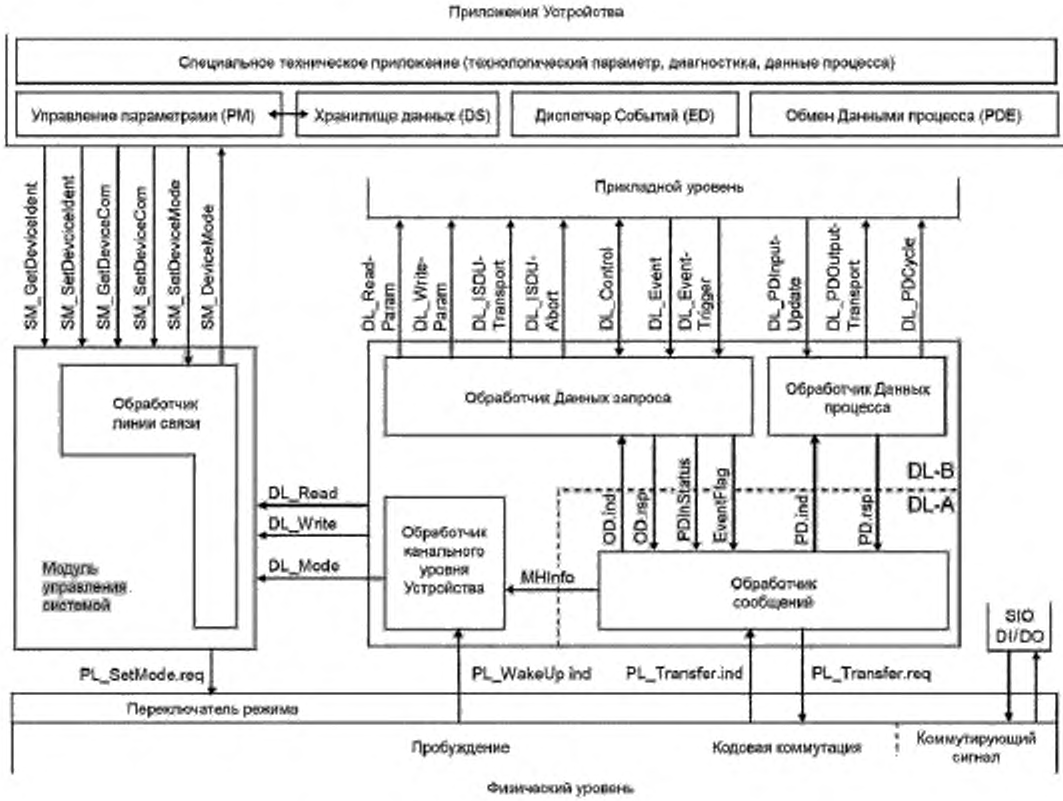


Рисунок 77 — Структура и услуги SM на Устройстве

SM Устройства предоставляет центральную контролирующую рабочую копию программы через Обработчик линии связи на всех стадиях инициализации, установки состояния по умолчанию (SIO), установки связи, передачи данных и возврата в исходный режим SIO.

SM на Устройстве взаимодействует с PL для установки требуемых регулировок драйвера линии и приемника (см. рисунок 15), с DL для получения необходимой информации из Ведущего узла (пробуждение, скорости передачи и др.) и с приложениями Устройства для обеспечения идентичности Устройства и совместимости (параметров идентификации).

Переходы между состояниями обработчика (см. рисунок 79) инициируются действиями порта Ведущего узла (пробуждение и связь) и передаются через DL Устройства, применяя индикацию DL_Mode и запросы (команды) услуги DL_Write.

SM обеспечивает параметры идентификации Устройства через интерфейс приложений Устройства.

В диаграмме последовательностей на рисунке 78 показана типовая последовательность от инициализации режима SIO по умолчанию и через запрос пробуждения до окончания передачи данных. Диаграмма последовательностей дополнена сценарием обработки ошибки связи, такой как истечение промежутка времени TDSIO, сбой связи или запрос от Ведущего узла, например, возврат в исходное состояние Fallback (вызванный Событием).

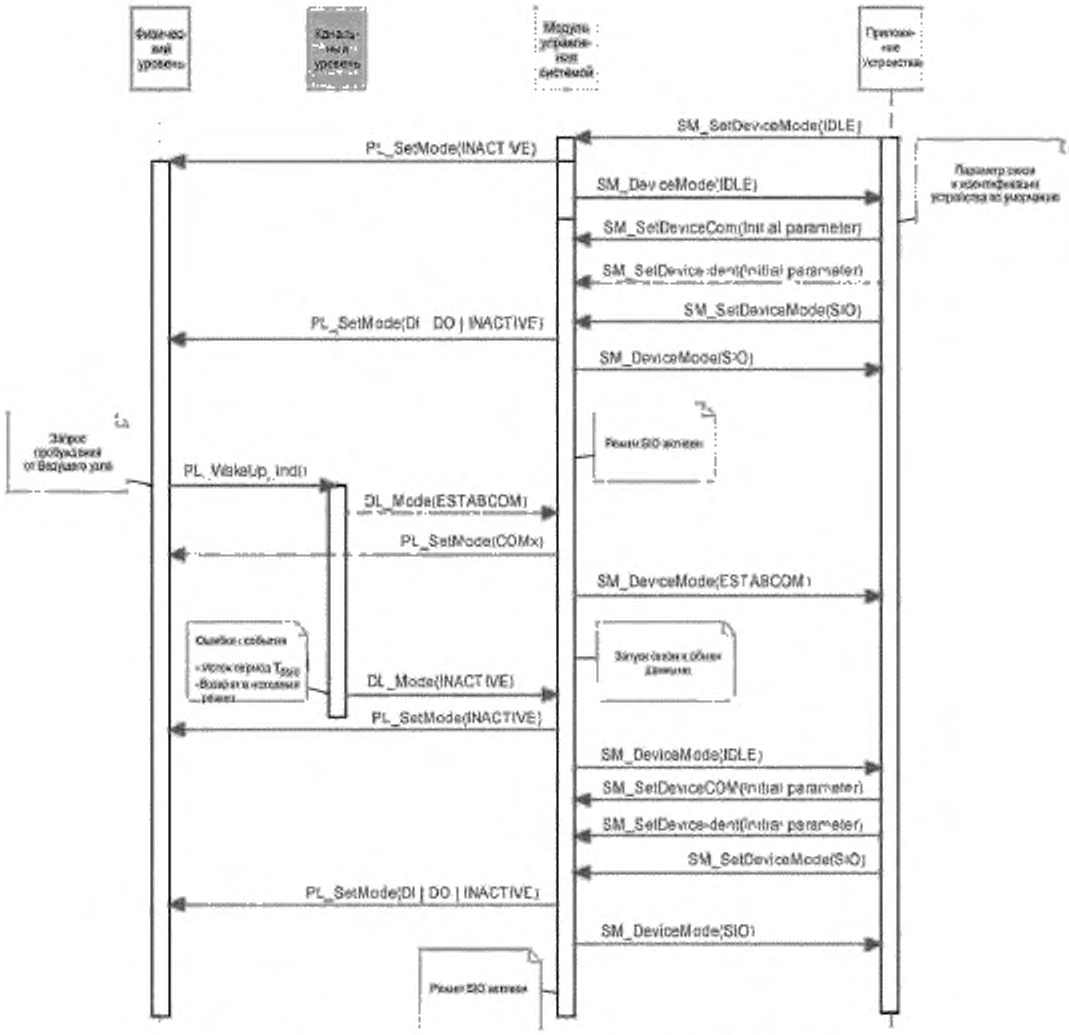


Рисунок 78 — Диаграмма последовательностей сценария использования «INACTIVE — SIO — SDCI — SIO»

Службы SM, показанные на рисунке 78, установлены в 9.3.2.

9.3.2 Услуги модуля управления системой на Устройстве

9.3.2.1 Обзор

В 9.3.2 описаны службы, которые SM на Устройстве предоставляет приложениям Устройства, как показано на рисунке 77.

В таблице 86 приведены средства Устройства в его роли инициатора или получателя для отдельных услуг SM.

Таблица 86 — Услуги SM на Устройстве

Имя услуги	Устройство
SM_SetDeviceCom	R
SM_GetDeviceCom	R
SM_SetDeviceIdent	R
SM_GetDeviceIdent	R
SM_SetDeviceMode	R
SM_DeviceMode	I
Обозначения (см. 3.3.4): I — Инициатор услуги; R — Приемник (Ответчик) услуги.	

9.3.2.2 Услуга SM_SetDeviceCom

Услуга SM_SetDeviceCom применяется для конфигурирования характеристик связи, поддерживаемых Устройством в модуле управления системой. Параметры сервисных примитивов приведены в таблице 87.

Таблица 87 — Услуга SM_SetDeviceCom

Имя параметра	.req	.cnf
Argument ParameterList	M M	
Result (+)		S
Result (–) ErrorInfo		S M

Argument (Аргумент)

Определяемые услугой параметры передаются в данном параметре.

ParameterList (СписокПараметров)

Данный параметр содержит сконфигурированные параметры связи для Устройства.

Тип параметра: Запись.

Элементы записи:

SupportedSIOMode (ПоддерживаемыйРежимSIO)

Данный параметр указывает, что режим SIO поддерживается Устройством.

Допустимые значения:
INACTIVE (высокий импеданс линии C/Q);
DI (линия C/Q находится в режиме цифрового ввода);
DO (линия C/Q находится в режиме цифрового вывода).

SupportedTransmissionrate (ПоддерживаемаяСкоростьПередачи)

Данный параметр указывает скорости передачи данных, поддерживаемые Устройством.

Допустимые значения:
COM1 (скорость передачи порта COM1);
COM2 (скорость передачи порта COM2);
COM3 (скорость передачи порта COM3).

MinCycleTime (МинВремяЦикла)
Данный параметр содержит минимальное время цикла, поддерживаемое Устройством (см. В.1.3).

M-sequenceCapability (СвойстваМ-Последовательностей)
Данный параметр указывает свойства М-последовательностей, поддерживаемые Устройством (см. В.1.4):
- поддержка ISDU;
- типы М-последовательностей в состоянии OPERATE;
- типы М-последовательностей в состоянии PREOPERATE.

RevisionID (ИдРедакции) (RID)
Данный параметр поддерживает редакцию протокола (см. В.1.5), поддерживаемую Устройством.

ProcessDataIn (ВхДанныеПроцесса)
Данный параметр содержит длину PD, подлежащих передаче в Ведущий узел (см. В.1.6).

ProcessDataOut (ВыхДанныеПроцесса)
Данный параметр содержит длину PD, подлежащих передаче из Ведущего узла (см. В.1.7).

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неуспешно.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
PARAMETER_CONFLICT (нарушена корректность набора параметров).

9.3.2.3 Услуга SM_GetDeviceCom

Услуга SM_GetDeviceCom применяется для чтения текущих характеристик связи из SM. Параметры сервисных примитивов приведены в таблице 88.

Таблица 88 — Услуга SM_SetDeviceCom

Имя параметра	.req	.cnf
Argument	M	
Result (+)		S
ParameterList		M
Result (–)		S
ErrorInfo		M

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

ParameterList (СписокПараметров)
Данный параметр содержит сконфигурированные параметры связи для Устройства.

Тип параметра: Запись.

Элементы записи:

CurrentMode (ТекущийРежим)
Данный параметр указывает режим SIO или режим связи Устройства.

Допустимые значения:
INACTIVE (линия C/Q имеет высокий импеданс);
DI (линия C/Q находится в режиме цифрового ввода);
DO (линия C/Q находится в режиме цифрового вывода);
COM1 (скорость передачи порта COM1);
COM2 (скорость передачи порта COM2);
COM3 (скорость передачи порта COM3).

MasterCycleTime (ВремяциклаВедущегоузла)
Данный параметр содержит время цикла Ведущего узла, которое будет установлено SM Ведущего узла (см. B.1.3). Данный параметр действует только в состоянии SM_Operate.

M-sequenceCapability (СвойстваМ-Последовательности)
Данный параметр указывает на текущие свойства, сконфигурированные в модуле управления системой на Устройстве (см. B.1.4):
- поддержка ISDU;
- типы M-последовательностей в состоянии OPERATE;
- типы M-последовательностей в состоянии PREOPERATE.

RevisionID (ИдРедакции) (RID)
Данный параметр содержит текущую редакцию протокола (см. B.1.5) в модуле управления системой на Устройстве.

ProcessDataIn (ВхДанныеПроцесса)
Данный параметр содержит длину PD, подлежащих передаче в Ведущий узел (см. B.1.6).

ProcessDataOut (ВыхДанныеПроцесса)
Данный параметр содержит длину PD, подлежащих передаче из Ведущего узла (см. B.1.7).

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
STATE_CONFLICT (в текущем состоянии услуга недоступна).

9.3.2.4 Услуга SM_SetDeviceIdent

Услуга SM_SetDeviceIdent применяется для конфигурирования данных идентификации Устройства в модуле управления системой. Параметры сервисных примитивов приведены в таблице 89.

Таблица 89 — Услуга SM_SetDeviceIdent

Имя параметра	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
Result (–)		S
ErrorInfo		M

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

ParameterList (СписокПараметров)
Данный параметр содержит сконфигурированные параметры связи для Устройства.

Тип параметра: Запись.

Элементы записи.

VendorID (ИдИзготовителя) (VID)
Данный параметр содержит идентификатор изготовителя VendorID, назначенный Устройству (см. В.1.8).

Длина данных: два октета.

DeviceID (ИдУстройства) (DID)
Данный параметр содержит один из назначенных идентификаторов Устройства. DeviceID (см. В.1.9).

Длина данных: три октета.

FunctionID (ИдФункции) (FID)
Данный параметр содержит один из назначенных идентификаторов функции FunctionID (см. В.1.10).

Длина данных: два октета.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
STATE_CONFLICT (в текущем состоянии услуга недоступна);
PARAMETER_CONFLICT (нарушена корректность набора параметров).

9.3.2.5 Услуга SM_GetDeviceIdent

Услуга SM_GetDeviceIdent применяется для чтения параметра идентификации Устройства из SM. Параметры сервисных примитивов приведены в таблице 90.

Таблица 90 — Услуга SM_GetDeviceIdent

Имя параметра	.req	.cnf
Argument	M	
Result (+)		S
ParameterList		M
Result (–)		S
ErrorInfo		M

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

ParameterList (СписокПараметров)
Данный параметр содержит сконфигурированные параметры связи для Устройства.

Тип параметра: Запись.

Элементы записи:

VendorID (ИдИзготовителя) (VID)
Данный параметр содержит фактический идентификатор изготовителя VendorID, назначенный Устройству (см. В.1.8).

Длина данных: два октета.

DeviceID (ИдУстройства) (DID)
Данный параметр содержит фактический идентификатор Устройства DeviceID (см. В.1.9).

Длина данных: три октета.

FunctionID (ИдФункции) (FID)
Данный параметр содержит фактический идентификатор функции FunctionID (см. В.1.10).

Длина данных: два октета.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что слуга завершилась неуспешно.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
STATE_CONFLICT (в текущем состоянии услуга недоступна).

9.3.2.6 Услуга SM_SetDeviceMode

Услуга SM_SetDeviceMode применяется для установки Устройства в определенное рабочее состояние при инициализации. Параметры сервисных примитивов приведены в таблице 91.

Таблица 91 — Услуга SM_SetDeviceMode

Имя параметра	.req	.cnf
Argument	M	
Mode	M	
Result (+)		S
Result (–)		S
ErrorInfo		M

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

Mode (Режим)
Допустимые значения:
IDLE (устройство переходит в состояние ожидания конфигурирования);
SIO (устройство переходит в режим, определенный в услуге SM_SetDeviceCom).

Result (+) (ПоложительныйРезультат)
Данный параметр выбора указывает, что услуга выполнена успешно.

Result (–) (ОтрицательныйРезультат)
Данный параметр выбора указывает, что услуга завершилась неудачно.

ErrorInfo (ИнформацияОбОшибке)
Данный параметр содержит информацию об ошибке.

Допустимые значения:
STATE_CONFLICT (в текущем состоянии услуга недоступна).

9.3.2.7 Услуга SM_DeviceMode

Услуга SM_DeviceMode применяется для индикации изменений в состояниях связи приложения Устройства. Параметры сервисных примитивов приведены в таблице 92.

Таблица 92 — Услуга SM_DeviceMode

Имя параметра	.ind
Argument	M
Mode	M

Argument (Аргумент)
Определяемые услугой параметры передаются в данном параметре.

Mode (Режим)
Допустимые значения:
IDLE (устройство перешло в состояние ожидания конфигурирования);
SIO (устройство перешло в режим, определенный в услуге «SM_SetDeviceCom»);
ESTABCOM (устройство перешло в режим управления системой «SM_ComEstablish»);
COM1 (устройство перешло в режим COM1);
COM2 (устройство перешло в режим COM2);
COM3 (устройство перешло в режим COM3);
STARTUP (устройство перешло в режим STARTUP);
IDENT_STARTUP (устройство перешло в режим управления системой «SM_IdentStartup»);
IDENT_CHANGE (устройство перешло в режим управления системой «SM_IdentCheck»);
PREOPERATE (устройство перешло в режим PREOPERATE);
OPERATE (устройство перешло в режим OPERATE).

9.3.3 Протокол модуля управления системой на Устройстве

9.3.3.1 Обзор

Поведение Устройства в основном управляется сообщениями Ведущего узла.

9.3.3.2 Конечная машина модуля управления системой на Устройстве

На рисунке 79 показана конечная машина SM на Устройстве. Конечная машина запускается обработчиком канального режима и приложением Устройства. Конечная машина анализирует различные стадии связи во время запуска и управляет состоянием линии связи Устройства.

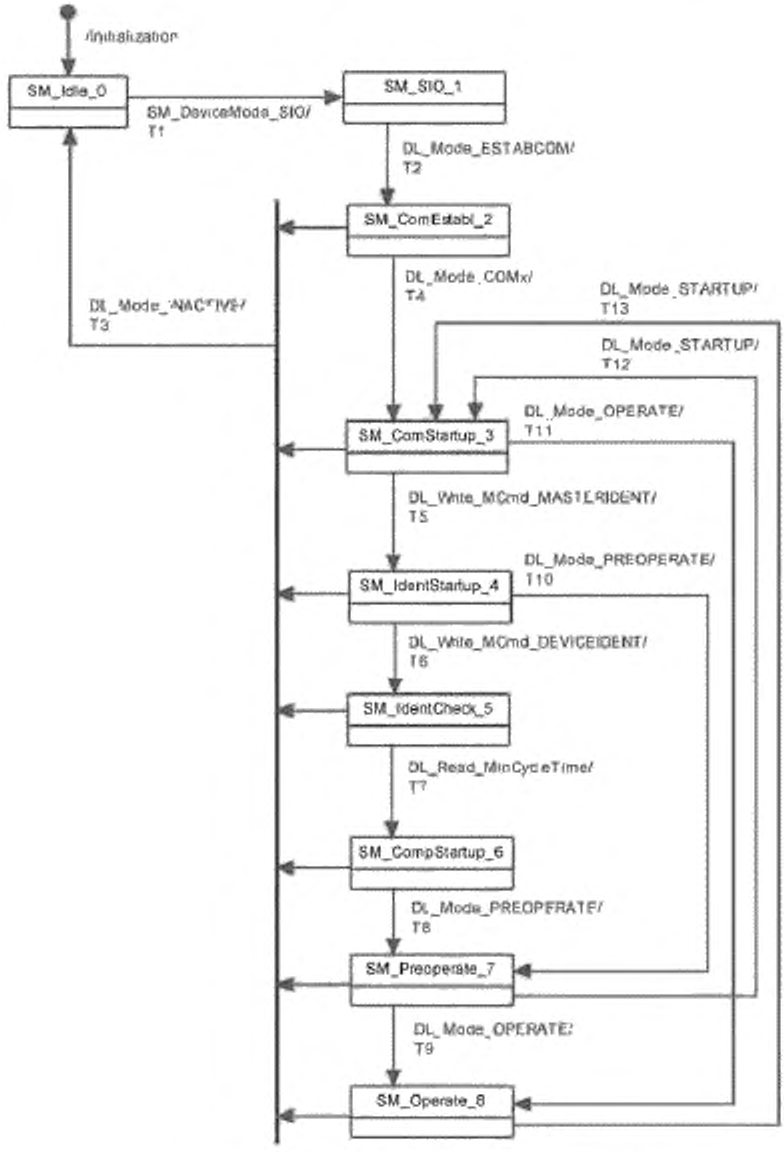


Рисунок 79 — Конечная машина SM на Устройстве

В таблице 93 установлены отдельные состояния и действия во время переходов.

Таблица 93 — Таблицы перехода состояний SM на Устройстве

Имя состояния	Описание состояния
SM_Idle_0	В состоянии SM_Idle SM ожидает конфигурирования приложением Устройства и установки в режим SIO. Выход из данного состояния происходит при получении запроса SM_SetDeviceMode(SIO) из приложения Устройства. Между приложением Устройства и SM выполняется следующая последовательность услуг: вызов услуги SM_SetDeviceCom (список начальных параметров); вызов услуги SM_SetDeviceIdent (VID, начальные DID, FID)

Продолжение таблицы 93

Имя состояния	Описание состояния
SM_SIO_1	В состоянии SM_SIO обработчик линии связи управления системой остается в режиме по умолчанию SIO. PL устанавливается с характеристиками режима SIO, определенными приложением Устройства через услугу SetDeviceMode. Выход из данного состояния происходит при получении индикации DL_Mode(ESTABCOM)
SM_ComEstablish_2	В состоянии SM_ComEstablish SM ожидает установления связи на DL. Выход из данного состояния происходит при получении индикации DL_Mode(INACTIVE) или DL_Mode(COMx), где COMx может быть любым из портов COM1, COM2 или COM3
SM_Com Start up_3	В состоянии SM_ComStartup параметр связи (страница 1 Непосредственных параметров, адреса от 0x02 по 0x06) считывается SM на Ведущем узле через запросы DL_Read. Выход из данного состояния происходит при получении одной из индикаций DL_Mode(INACTIVE), DL_Mode(OPERATE) (только для устаревшего Ведущего узла) или DL_Write(MCmd_MASTERIDENT)
SM_IdentStartup_4	В состоянии SM_IdentStartup данные идентификации (VID, DID, FID) считываются и проверяются Ведущим узлом. В случае несовместимостей SM на Ведущем узле записывает поддерживаемую редакцию SDCI (RID) и сконфигурированный идентификатор Устройства (DID) в Устройство. Выход из данного состояния происходит при получении индикаций DL_Mode(INACTIVE), DL_Mode(PREOPERATE) (проверка совместимости прошла успешно) или запроса DL_Write(MCmd_DEVICEIDENT) (запрос новой совместимости)
SM_IdentCheck_5	В состоянии SM_IdentCheck SM ожидает новой инициализации параметров связи или идентификации. Выход из данного состояния происходит при получении идентификации DL_Mode(INACTIVE) или запроса DL_Read (страница 1 Непосредственных параметров, адрес 0x02 = MinCycleTime). В данном состоянии приложение Устройства будет проверять параметры RID и DID из SM и устанавливать эти данные в поддерживаемые значения. Это означает, что между приложением Устройства и SM выполняется следующая последовательность услуг: вызов услуги SM_GetDeviceCom (сконфигурированный RID, список параметров); вызов услуги SM_GetDeviceIdent (сконфигурированный DID, список параметров); вызов проверок из приложения Устройства и обеспечение совместимости функции и параметров; вызов услуги SM_SetDeviceCom (новый поддерживаемый RID, новый список параметров); вызов услуги SM_SetDeviceIdent (новый поддерживаемый DID, список параметров)
SM_CompStartup_6	В состоянии SM_CompatStartup данные связи и идентификации считываются повторно и подтверждаются SM на Ведущем узле. Выход из данного состояния происходит при получении индикации DL_Mode(INACTIVE) или DL_Mode(PREOPERATE)
SM_Preoperate_7	Во время пребывания в состоянии SM_Preoperate SM может считываться и проверяться заводской номер SerialNumber, а также может производиться параметризация DS и Устройства. Выход из данного состояния происходит при получении одного из сообщений DL_Mode(INACTIVE), DL_Mode(STARTUP) или DL_Mode(OPERATE)
SM_Operate_8	Во время пребывания в состоянии SM_Operate активны обмен циклическими PD и передача ациклических OD. Выход из данного состояния происходит при получении индикаций DL_Mode(INACTIVE) или DL_Mode(STARTUP)

Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	Устройство переключается в сконфигурированный режим SIO при получении пускового сигнала SM_SetDeviceMode.req(SIO). Вызов услуги PL_SetMode(DI|DO|INACTIVE). Вызов услуги SM_DeviceMode(SIO)

Окончание таблицы 93

Переход	Исходное состояние	Конечное состояние	Действие
T2	1	2	Устройство переключается в режим связи при получении пускового сигнала DL_Mode.ind(ESTABCOM). Вызов услуги PL_SetMode(COMx). Вызов услуги SM_DeviceMode(ESTABCOM)
T3	2, 3, 4, 5, 6, 7, 8	0	Устройство переключается в режим SM_Idle при получении пускового сигнала DL_Mode.ind(INACTIVE). Вызов услуги PL_SetMode(INACTIVE). Вызов услуги SM_DeviceMode(IDLE)
T4	2	3	Приложение Устройства получает индикацию о скорости передачи в бодах, которая была установлена услугой DL_Mode.ind(COMx), запущенной на DL. Вызов услуги SM_DeviceMode(COMx)
T5	3	4	Происходит вход в фазу идентификации Устройства при получении пускового сигнала DL_Write.ind(MCmd_MASTERIDENT). Вызов услуги SM_DeviceMode(IDENTSTARTUP)
T6	4	5	Происходит вход в фазу проверки идентичности Устройства при получении пускового сигнала DL_Write.ind(MCmd_DEVICEIDENT). Вызов услуги SM_DeviceMode(IDENTCHANGE)
T7	5	6	Вход в фазу проверки совместимости Устройства при получении пускового сигнала DL_Read.ind (страница 1 Непосредственных параметров, адрес 0x02 = MinCycleTime)
T8	6	7	Происходит вход в фазу предварительных операций Устройства при получении пускового сигнала DL_Mode.ind(PREOPERATE). Вызов услуги SM_DeviceMode(PREOPERATE)
T9	7	8	Происходит вход в фазу операций Устройства при получении пускового сигнала DL_Mode.ind(OPERATE). Вызов услуги SM_DeviceMode(OPERATE)
T10	4	7	Происходит вход в фазу предварительных операций Устройства при получении пускового сигнала DL_Mode.ind(PREOPERATE). Вызов услуги SM_DeviceMode(PREOPERATE)
T11	3	8	Происходит вход в фазу операций Устройства при получении пускового сигнала DL_Mode.ind(OPERATE). Вызов услуги SM_DeviceMode(OPERATE)
T12	7	3	Происходит переход в фазу запуска связи Устройства при получении пускового сигнала DL_Mode.ind(STARTUP). Вызов услуги SM_DeviceMode(STARTUP)
T13	8	3	Происходит переход в фазу запуска связи Устройства при получении пускового сигнала DL_Mode.ind(STARTUP). Вызов услуги SM_DeviceMode(STARTUP)
Внутренние элементы		Тип	Определение
COMx		Переменная	Скорость передачи любого из портов COM1, COM2 или COM3
DL_Write_MCmd_xxx		Услуга	Услуга DL записывает команды Ведущего узла (xxx = значения из таблицы B.2)

На рисунке 80 показана типовая диаграмма последовательностей при запуске связи SM для установки соответствия конфигурации порта на Ведущем узле (нормальный запуск).

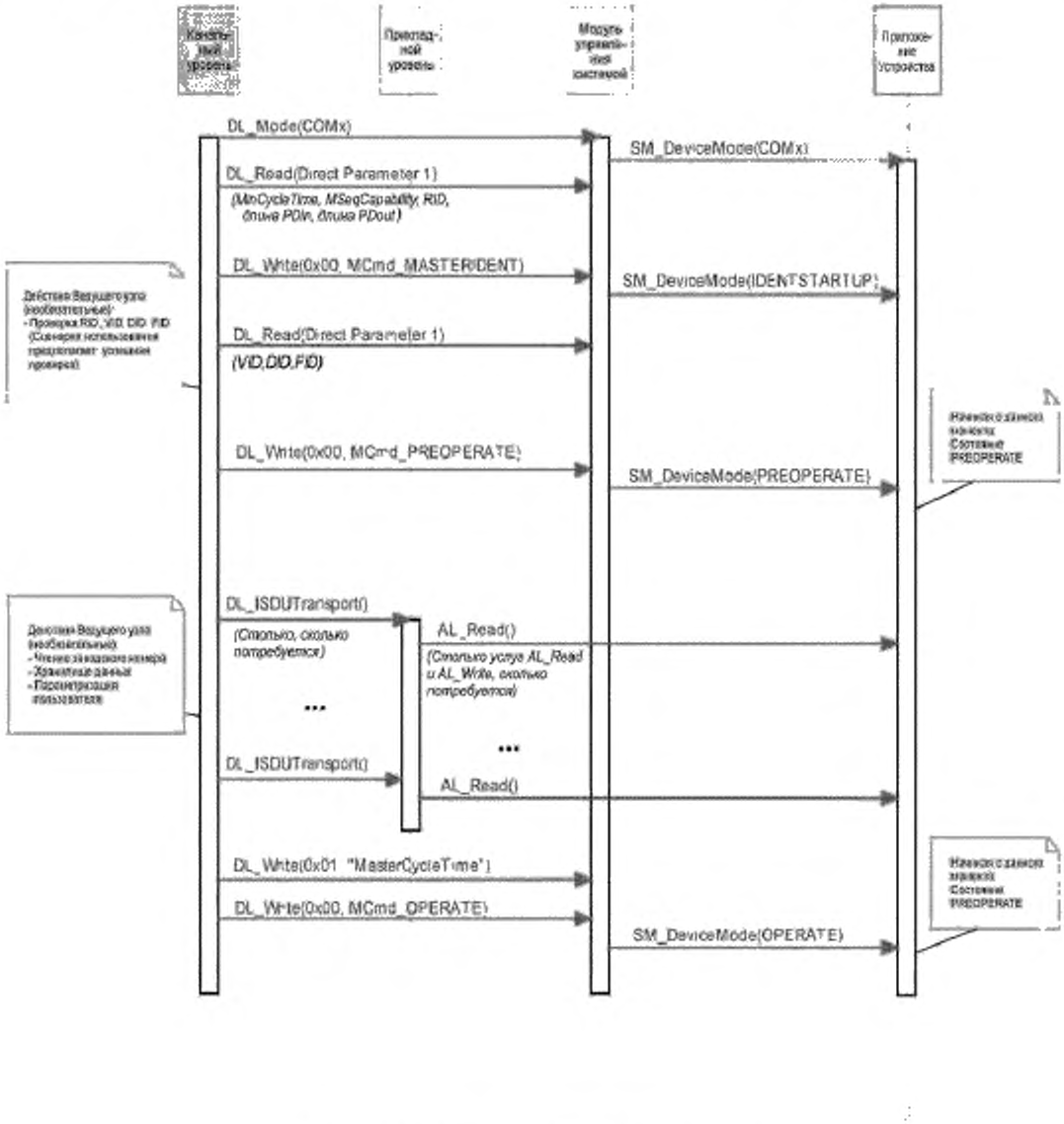


Рисунок 80 — Диаграмма последовательностей нормального запуска Устройства

На рисунке 81 показана типовая диаграмма последовательностей при запуске связи SM для установки соответствия конфигурации порта на Ведущем узле (режим совместимости). В этом режиме Ведущий узел пытается перезаписать параметры идентификации Устройства для достижения совместимого и работоспособного режима.

Диаграмма последовательностей на рисунке 81 показывает только действия, предшествующие переходу в состояние PREOPERATE. Последующие действия до перехода в состояние OPERATE показаны на рисунке 80.

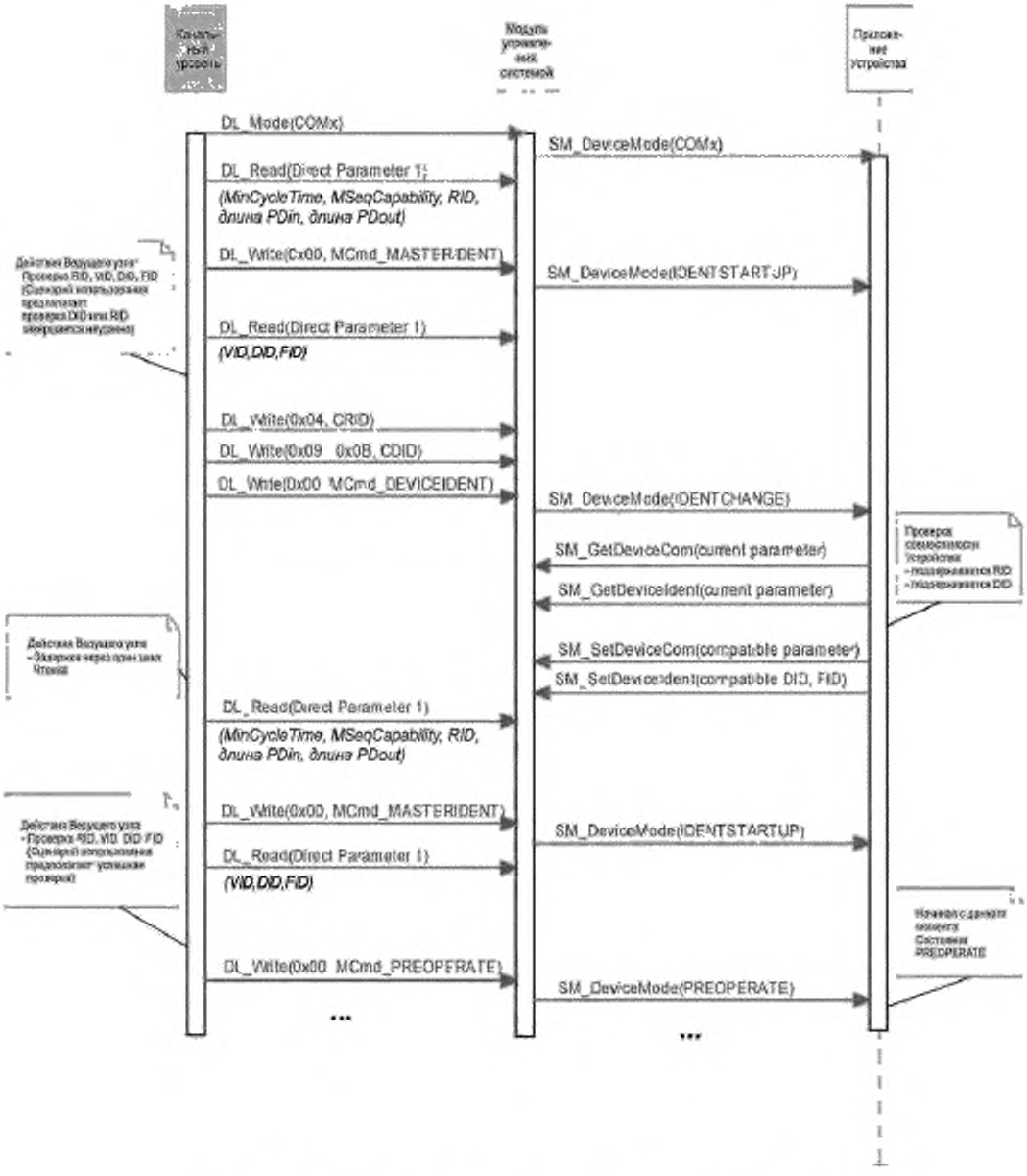


Рисунок 81 — Диаграмма последовательностей запуска Устройства в режиме совместимости

На рисунке 82 показана типовая диаграмма последовательностей при запуске связи SM для установки соответствия конфигурации порта на Ведущем узле. SM на Ведущем узле пытается переконфигурировать Устройства с альтернативными параметрами идентификации Устройства (режим совместимости). В данном сценарии использования считается, что альтернативные параметры являются несовместимыми.

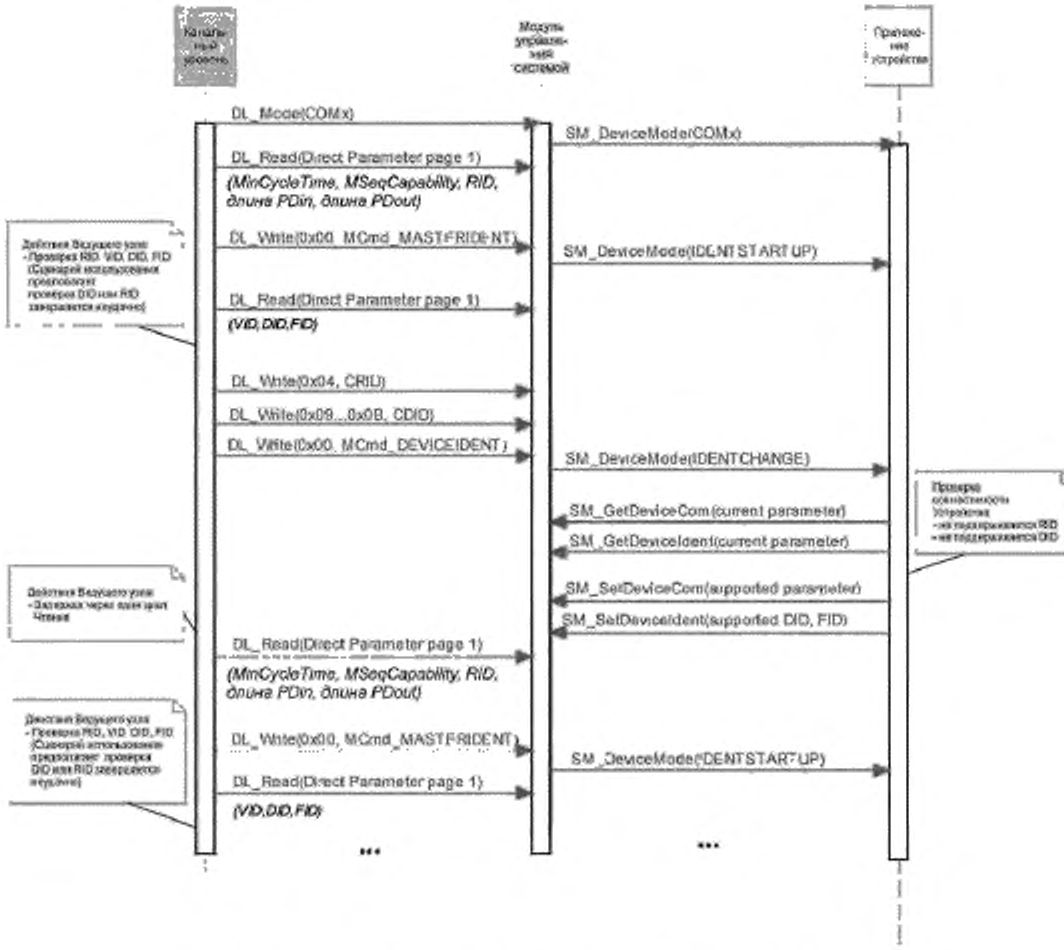


Рисунок 82 — Диаграмма последовательностей запуска Устройства, когда не удается достичь совместимости

10 Устройство

10.1 Обзор

На рисунке 83 предоставлен обзор полной структуры и услуг Устройства.

Приложения Устройства включают в себя прежде всего специальное техническое приложение, состоящее из преобразователя с его технологическими параметрами, диагностической информацией и PD. Общие приложения Устройства включают в себя:

- PM, имеющий дело с проверкой совместимости и правильности полного набора специальных технических (зависящих от изготовителя) и общих системных параметров (см. 10.3);
- DS, который по запросу отправляет или скачивает параметры на Ведущий узел (см. 10.4);
- ED, наблюдающий за состояниями и передающий диагностическую информацию, такую как уведомления, предупреждения, ошибки и запросы Устройства, как инициативы периферийных устройств (см. 10.5);
- PDE, модифицирующее структуры данных для передачи в случае датчика или подготавливающее полученные структуры данных для генерации сигнала. Устройство PDE также контролирует рабочие состояния для обеспечения достоверности PD (см. 10.2).

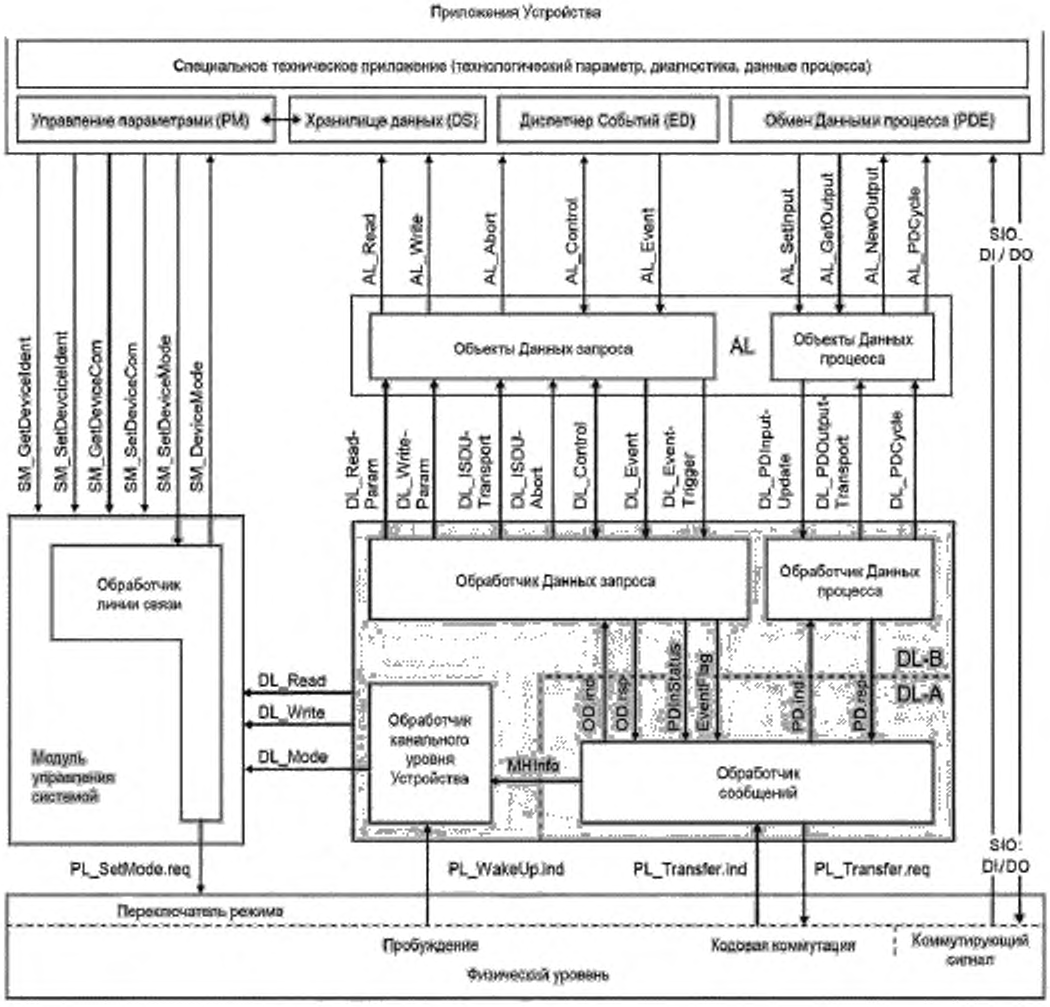


Рисунок 83 — Структура и услуги устройства

Данные приложения Устройства предоставляют стандартные методы, функции и параметры, общие для всех Устройств, специальные функции и параметры Устройства, установленные в разделе 10.

10.2 Обмен Данными процесса (PDE)

Устройство PDE циклически передает и получает PD без вмешательства OD (параметров, команд и Событий).

Исполнительный механизм (выходные PD) наблюдает за циклическими передачами и переходит в подходящие состояния по умолчанию, например, сохраняет последнее значение, осуществляет остановку и отключение питания при любых прерываниях передачи данных (см. 7.3.3.5 и 10.7.3). Исполнительный механизм ожидает команды обработки выходных PD ProcessDataOutputOperate Ведущего узла (см. таблицу B.2, «допустимые» выходные PD) до возобновления нормального функционирования после рестарта в случае прерывания.

Во время циклического обмена данными исполнительный механизм (выходные PD) получает команду Ведущего узла DeviceOperate (ОбработатьУстройство) всякий раз, когда выходные PD становятся недопустимыми, и команду обработки выходных PD ProcessDataOutputOperate Ведущего узла всякий раз, когда они снова становятся допустимыми (см. таблицу В.2).

Устройствам датчика (входным PD) нет необходимости отслеживать циклический обмен данными. Тем не менее, если Устройство не в состоянии гарантировать допустимые PD, состояние PL Process-DataInvalid (Недопустимые PD, см. А.1.5) будет уведомлять приложение Ведущего узла.

10.3 Менеджер параметров (PM)

10.3.1 Общие положения

Устройство может быть параметризовано двумя основными методами: применяя Непосредственные параметры или применяя пространство памяти Индекса, доступное с помощью ISDU (см. рисунок 5).

Обязательными для всех Устройств являются так называемые Непосредственные параметры на странице 1. Данная страница содержит общие параметры связи и идентификации (см. В.1).

Страница 2 Непосредственных параметров дополнительно предлагает пространство для максимального числа в 16 октетов специальных технических (определяемых изготовителем) параметров для Устройств, требующих не более этого ограниченного числа параметров. Данная возможность применима для небольших систем, характеризующихся следующими свойствами: связь ISDU не реализована, возможна более легкая обработка полевой шины, но с меньшими удобствами. Доступ к странице 2 Непосредственных параметров осуществляется через услуги AL_Read и AL_Write (см. 10.7.5).

Передача параметров в память Индекса и из нее может выполняться двумя способами: последовательно, параметр за параметром, или применяя блок параметров. Последовательная передача параметров, как установлено в 10.3.4, защищается несколькими проверками и подтверждением переданного параметра. Отрицательное подтверждение содержит соответствующее описание ошибки, и параметр не активируется. Передача блочного параметра, как установлено в 10.3.5, откладывает проверку непротиворечивости параметров и их активацию до завершения передачи. Устройство выполняет проверки после получения специальной команды и возвращает подтверждение или отрицательное подтверждение с соответствующим описанием ошибки. В данном случае переданные параметры будут отвергнуты и будет выполнен возврат к предыдущему набору параметров для обеспечения правильного функционирования Устройства.

10.3.2 Конечная машина PM

Устройство может быть параметризовано, применяя механизмы ISDU, если PM активен. Основными функциями PM является передача параметров Ведущему узлу [Upload (Отправить)], Устройству [Download (Скачать)] и проверка непротиворечивости и допустимости на Устройстве [ValidityCheck (ПроверитьДопустимость)], как показано на рисунке 84.

PM управляется сообщениями команд Ведущего узла (см. таблицу В.9). Например, сторожевое условие [Upload Start] (НачатьПодкачку) соответствует получению команды системы ParamUploadStart (НачатьПодкачкуПараметров), сторожевое условие [UploadEnd] (ЗавершитьПодкачку) — получению команды системы ParamUploadEnd (ЗавершитьПодкачкуПараметров).

Примечание 1 — После прерывания связи SM на Ведущем узле применяет услугу SM_DeviceMode с переменной «INACTIVE» для прекращения процесса отправки и возврата в состояние «IDLE».

Любая новая команда ParamUploadStart (НачатьПодкачкуПараметров) или ParamDownloadStart (НачатьСкачиваниеПараметров), поступающая при другой ожидающей последовательности, например из-за неожиданного закрытия инструмента параметризации от изготовителя, будет приводить к отмене ожидающей последовательности. Соответствующие изменения параметра отбрасываются.

Примечание 2 — Пользовательская программа PLC и инструмент параметризации могут конфликтовать (множественный доступ), например, если во время ввода в эксплуатацию пользователь не блокировал доступы от программы PLC и изменяет параметры через инструмент.

Механизм PM на устройстве всегда активен, и услуга DS_ParUpload.req в состоянии Т4 применяется для запуска механизма DS в 10.4.2.

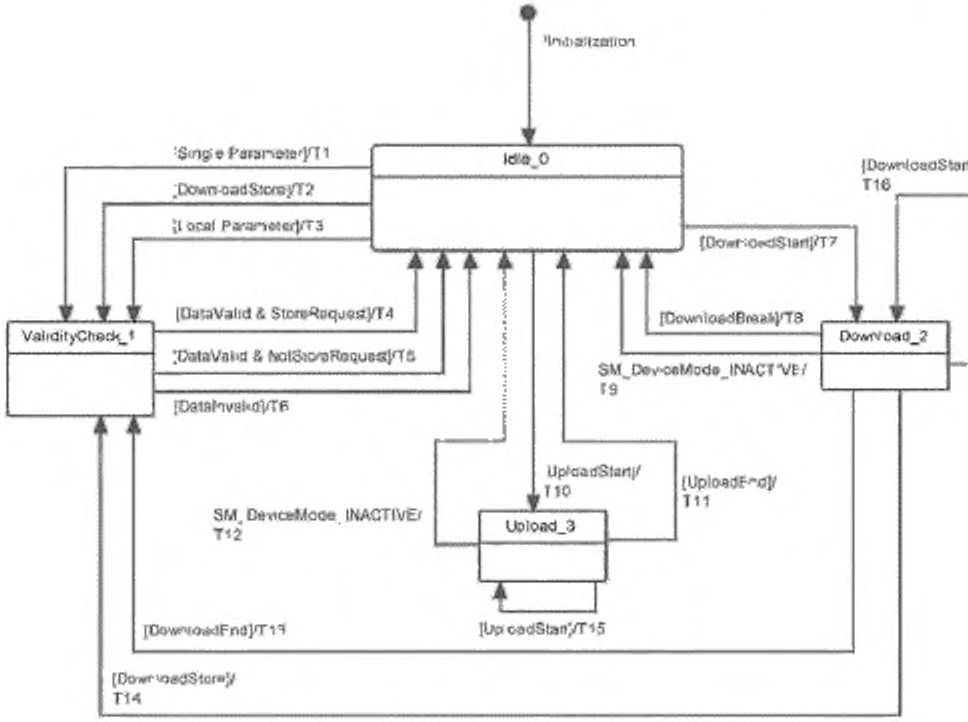


Рисунок 84 — Конечная машина PM

В таблице 94 показаны таблицы перехода состояний конечной машины PM.

Таблица 94 — Таблицы перехода состояний конечной машины PM

Имя состояния		Описание состояния	
Idle_0		Ожидание передачи параметров	
ValidityCheck_1		Проверка непротиворечивости и допустимости текущего набора параметров	
Download_2		Активное скачивание параметров, локальная параметризация блокирована (например, настройка в режиме обучения)	
Upload_3		Активная отправка параметров; параметризация глобально блокирована. Все доступы на запись для изменений параметров через инструменты будут отвергаться [Код ошибки ISDU «Service temporarily not available — Device control» (Услуга временно недоступна — Управление Устройством)]	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	0	1	Устанавливает StoreRequest (= TRUE)
T3	0	1	Устанавливает StoreRequest (= TRUE)
T4	1	0	Отметить набор параметров как допустимый; вызвать DS_ParUpload.req для DS; разрешить положительное подтверждение передачи, сбросить флаг StoreRequest (= FALSE)
T5	1	0	Отметить набор параметров как допустимый; разрешить положительное подтверждение передачи

Окончание таблицы 94

Переход	Исходное состояние	Конечное состояние	Действие
T6	1	0	Отметить набор параметров как недопустимый; разрешить отрицательное подтверждение передачи; сбросить флаг StoreRequest (= FALSE); сбросить буфер параметров
T7	0	2	Блокировать доступ к локальным параметрам
T8	2	0	Разблокировать доступ к локальным параметрам; сбросить буфер параметров
T9	2	0	Разблокировать доступ к локальным параметрам; сбросить буфер параметров
T10	0	3	Блокировать доступ к локальным параметрам
T11	3	0	Разблокировать доступ к локальным параметрам
T12	3	0	Разблокировать доступ к локальным параметрам
T13	2	1	Разблокировать доступ к локальным параметрам
T14	2	1	Разблокировать доступ к локальным параметрам, установить флаг StoreRequest (= TRUE)
T15	3	3	Блокировать доступ к локальным параметрам
T16	2	2	Сбросить буфер параметров таким образом, чтобы возможный повторный запуск не был блокирован
Внутренние элементы		Тип	Определение
DownloadStore		Bool	Получена команда системы ParamDownloadStore (СохранитьСкачанные Параметры), см. таблицу B.9
DataValid		Bool	Положительный результат проверки на непротиворечивость и допустимость
DataInvalid		Bool	Отрицательный результат проверки на непротиворечивость и допустимость
DownloadStart		Bool	Получена команда системы ParamDownloadStart (НачатьСкачивание-Параметров), см. таблицу B.9
DownloadBreak		Bool	Получена команда системы ParamBreak (ПрерватьПередачуПараметров) или ParamUploadStart (НачатьПодкачкуПараметров)
DownloadEnd		Bool	Получена команда системы ParamDownloadEnd (ЗавершитьСкачивание Параметров), см. таблицу B.9
StoreRequest		Bool	Флаг для затребованной последовательности DS, т. е. получена команда системы ParamDownloadStore (СохранитьСкачанныеПараметры) (= TRUE)
NotStoreRequest		Bool	Инвертированное значение запроса на сохранение StoreRequest
UploadStart		Bool	Получена команда системы ParamUploadStart (НачатьПодкачкуПараметров), см. таблицу B.9
UploadEnd		Bool	Получена команда системы ParamUploadEnd (ЗавершитьПодкачкуПараметров), см. таблицу B.9
SingleParameter		Bool	В случае «одиночного параметра» — как определено в 10.3.4
LocalParameter		Bool	В случае «локального параметра» — как определено в 10.3.3

PM поддерживает обработку передачи «одиночного параметра» (Индекс и Субиндекс), а также передачу блочного параметра (полного набора параметров).

10.3.3 Динамический параметр

Параметры, доступные через услуги чтения или записи SDCI, могут также быть изменены через встроенные элементы управления (например, кнопку обучения) или человеко-машинный интерфейс

Устройства. Данные изменения проходят такие же проверки допустимости, как и при доступе к одиночному параметру. Таким образом, в случае положительного результата DataValid (ДопустимыеДанные) на рисунке 84 флаг StoreRequest (ЗапросСохранения) будет применен, чтобы добиться непротиворечивости DS. В случае отрицательного результата InvalidData (НедопустимыеДанные) будут восстановлены соответствующие параметры («откат»). Кроме того, рекомендуется применять специфическую для Устройства индикацию в человеко-машинном интерфейсе для положительной или отрицательной обратной связи с пользователем.

Рекомендуется избегать одновременного доступа к параметру через локальные элементы управления и услуги записи SDCI.

10.3.4 Одиночный параметр

Типовые диаграммы последовательностей для допустимого и недопустимого изменений одиночного параметра показаны на рисунке 85.

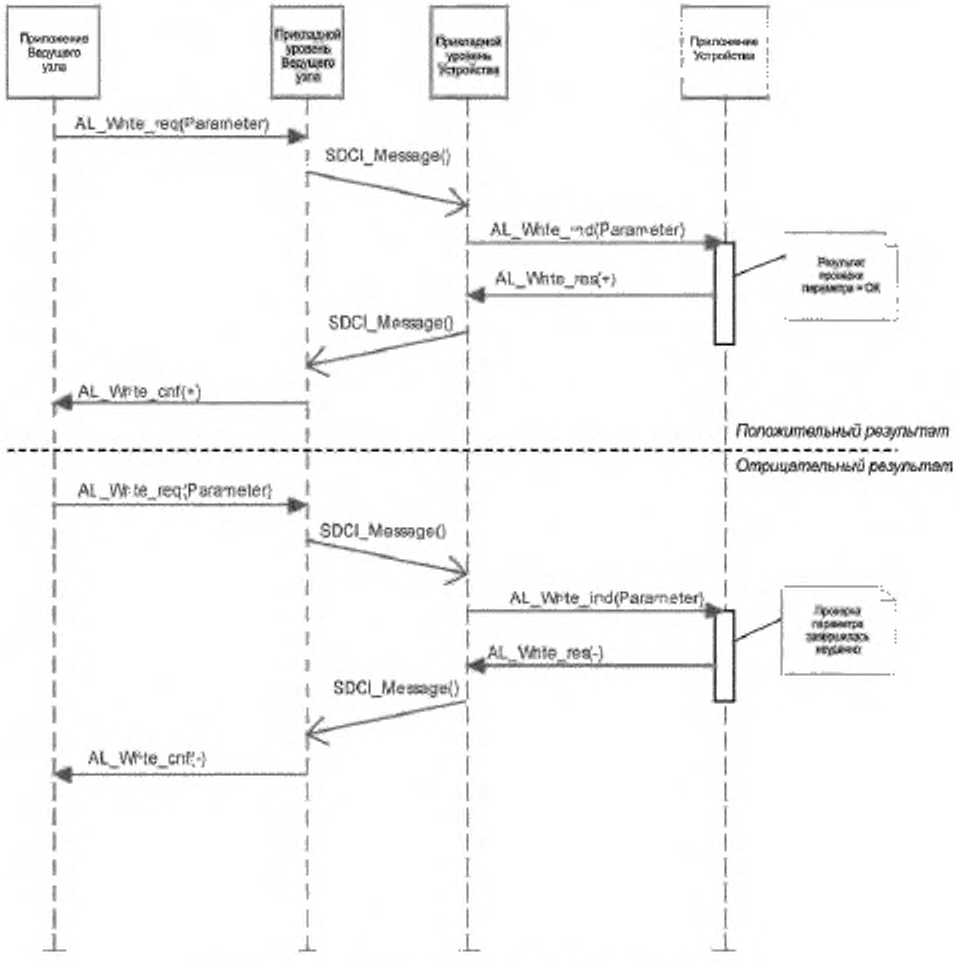


Рисунок 85 — Положительный и отрицательный результаты проверки параметра

Если выполняется параметризация с одиночным параметром через объекты ISDU, Устройство проверяет доступ, структуру, непротиворечивость и допустимость (см. таблицу 95) переданных данных вместе с контекстом полного набора параметров и возвращает результат в подтверждении. Отрицательное подтверждение переносит индикации ошибки из таблицы С.2.

Таблица 95 — Определение проверки параметров

Проверка параметра	Определение	Индикация ошибки
Доступ	Проверяет законность прав доступа для данного Индекса/Субиндекса, независимо от содержания данных (Индекс/Субиндекс постоянно или временно недоступен, доступ на запись или Индекс только на чтение)	См. С.2.3—С.2.8
Непротиворечивость	Проверяет допустимость содержания данных всего набора параметров, исследуя взаимное влияние и корреляцию между параметрами	См. С.20.16 и С.20.17
Структура	Проверяет допустимость структуры данных (например, размер данных). Могут записываться только полные структуры данных, например два октета в тип данных UInteger16	См. С.2.12 и С.2.13
Допустимые	Проверяет допустимость содержания данных отдельных параметров, проводя тесты на пределы данных	См. С.2.9—С.2.11, С.2.14, С.2.15

10.3.5 Блочный параметр

Приложения пользователей, такие как функциональные блоки в PLC и программное обеспечение инструментов параметризации, могут применять команду начала и завершения для указания начала и окончания передачи блочного параметра. Во время передачи блочного параметра приложение Устройства запрещает любые изменения параметров, исходящих из других источников, например, локальную оптимизацию, настройку в режиме обучения и т. п.

Типовая диаграмма последовательностей допустимых изменений блочного параметра с необязательным запросом сохранения в DS показана на рисунке 86.

Типовая диаграмма последовательностей недопустимых изменений блочного параметра показана на рисунке 87.

Команда ParamDownloadStart (НачатьСкачиваниеПараметров, см. таблицу В.9) указывает на начало передачи блочного параметра в направлении скачивания от приложения пользователя к Устройству. Команда системы ParamDownloadEnd (ЗавершитьСкачиваниеПараметров) или ParamDownloadStore (СохранитьСкачанныеПараметры) завершает данную последовательность. Обе функции похожи. Однако команда системы ParamDownloadStore (СохранитьСкачанныеПараметры) дополнительно заставляет механизм DS отправить набор параметров через Событие DS_UPLOAD_REQ (см. 10.4.2).

Во время скачивания блочного параметра проверка непротиворечивости для одиночного передаваемого параметра запрещается и параметры не активируются. С командой ParamDownloadEnd (ЗавершитьСкачиваниеПараметров) Устройство проверяет полный набор параметров и показывает результат инициатору передачи блочного параметра в подтверждение ISDU в ответ на команду.

Во время скачивания блочного параметра всегда выполняются проверки доступа и структуры (см. таблицу 95). По запросу может выполняться также проверка допустимости. PM не выходит из режима передачи блока в случае незаконного доступа или нарушений структуры.

В случае недопустимого набора параметров измененные параметры будут отброшены и будет выполнен откат к предыдущему набору параметров. Отрицательное подтверждение содержит одну из индикаций ошибок, приведенных в таблице С.2. При отрицательном подтверждении команды системы ParamDownloadStore (СохранитьСкачанныеПараметры) запрос на сохранение параметров в DS пропускается.

Команда ParamUploadStart (НачатьПодкачкуПараметров, см. таблицу В.9) указывает на начало передачи блочного параметра в направлении от Устройства к приложению пользователя. Команда системы ParamUploadEnd (ЗавершитьПодкачкуПараметров) завершает данную последовательность и указывает на конец передачи.

Передача блочного параметра прекращается, если PM получает команду системы ParamBreak (ПрекратитьПередачуПараметров). В данном случае передача блока завершается без каких-либо изменений в установках параметров.

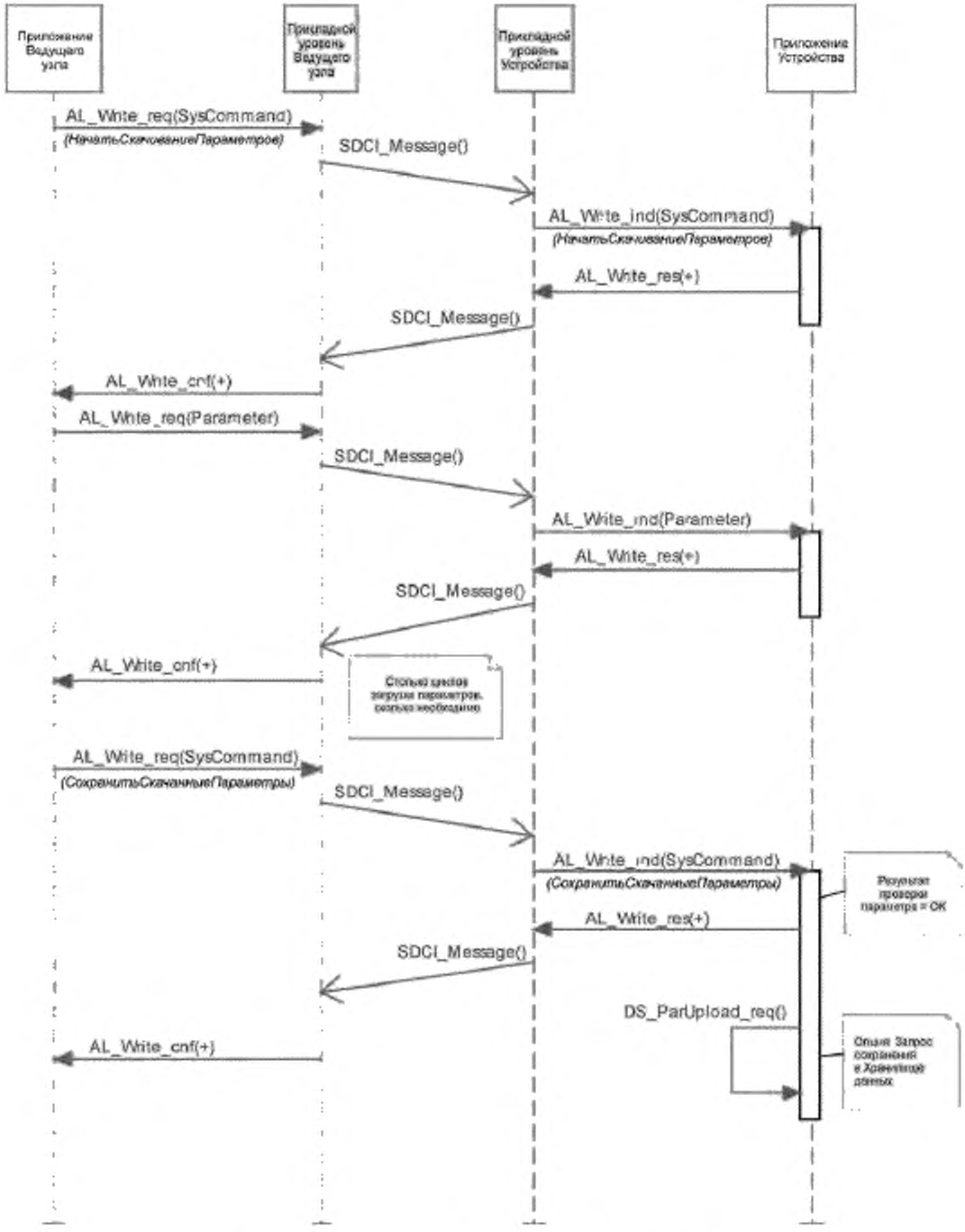


Рисунок 86 — Успешное скачивание блочного параметра с запросом сохранения в DS

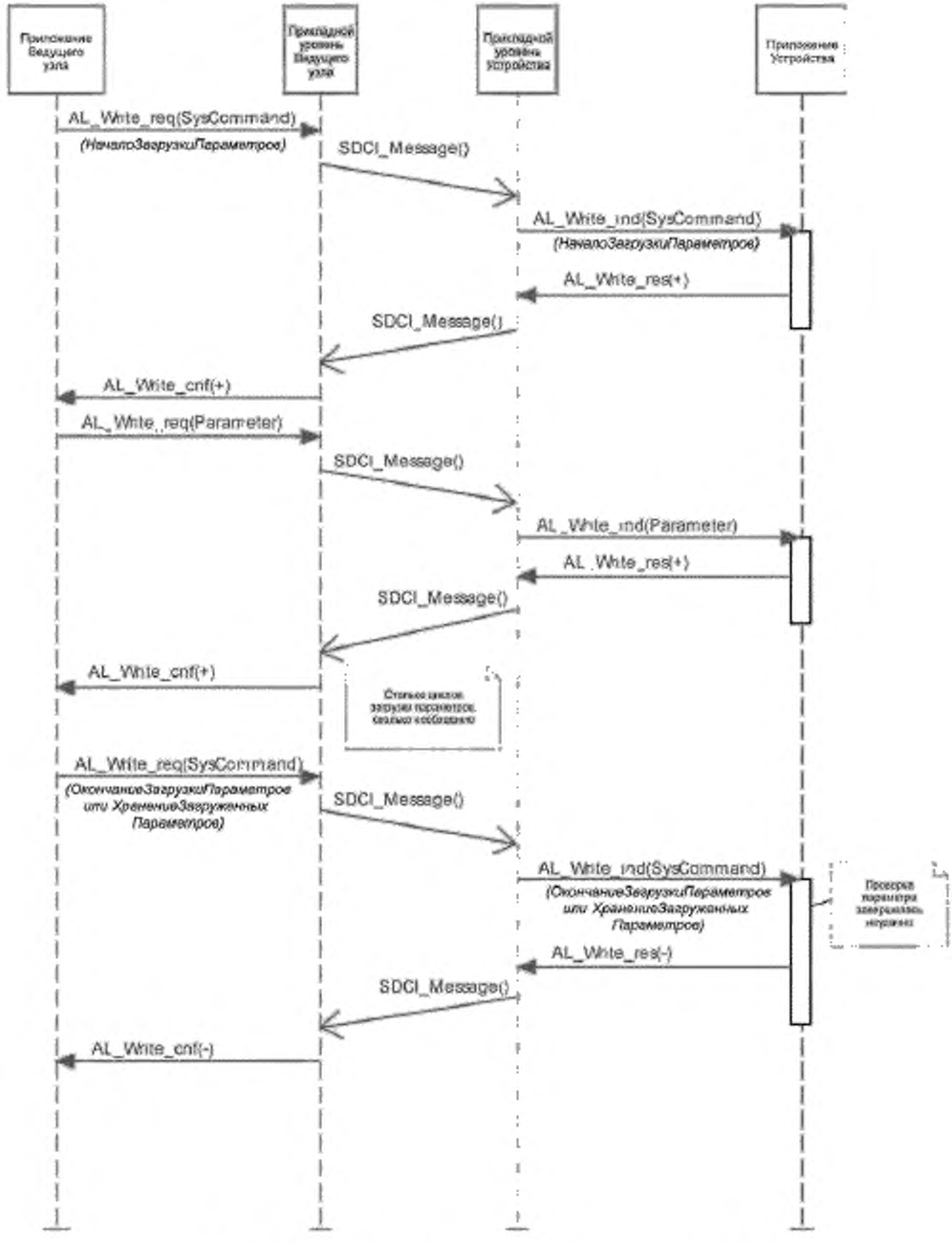


Рисунок 87 — Неудачное скачивание блочного параметра

10.3.6 Параллельный доступ к параметрам

В случае параллельного доступа из различных приложений пользователя выше уровня Ведущего узла нет механизма, обеспечивающего непротиворечивость параметров в пределах Устройства. Данные ситуации должны поддерживаться или блокироваться на пользовательском уровне.

10.3.7 Обработка команд

Команды приложения, такие как настройка в режиме обучения или восстановление заводских установок, передаются в виде параметров. Команда приложения подтверждается положительным ответом услуги — AL_Write.res(+). Отрицательный ответ услуги — AL_Write.res(–) — указывает на неуспешное выполнение команды приложения. В обоих случаях должны учитываться ограничения на превышение лимита времени ISDU (см. таблицу 97).

10.4 Хранилище данных (DS)

10.4.1 Общие положения

Механизм DS обеспечивает согласованную и актуальную буферизацию параметров Устройства на верхних уровнях программ PLC или сервера параметров полевых шин. DS между Ведущими узлами и Устройствами устанавливается в настоящем стандарте, в то время как механизмы хранилищ памяти на соответствующих верхних уровнях зависят от конкретных полевых шин или систем. Устройство поддерживает стандартизованный набор объектов, предоставляющий информацию о параметрах для DS, таких как требования к объему памяти, управляющую информацию и информацию о состояниях механизма DS (см. таблицу B.10). Проверки параметров DS применяют Контрольную сумму параметров.

Реализация механизма DS, устанавливаемая в настоящем стандарте, настоятельно рекомендуется для Устройств. Если такой механизм не поддерживается, то поставщик Устройства обязан описать, каким образом должна обеспечиваться параметризация Устройства в системе после его замены при отсутствии соответствующих инструментов.

10.4.2 Конечная машина Хранилища данных

Любой измененный набор допустимых параметров должен отправляться в DS. Отправка инициализируется Устройством возбуждением События «DS_UPLOAD_REQ» (см. таблицу D.2). Устройство сохраняет внутреннее состояние «Data Storage Upload» (Подкачка в DS в энергонезависимой памяти (см. таблицу B.10, Свойство состояния) до тех пор, пока оно не получит команду DS DS_UploadEnd (ЗавершитьПодкачкуВХранилищаДанных) или DS_DownloadEnd (ЗавершитьСкачиваниеИзХранилищаДанных).

Устройство генерирует Событие запроса подкачки в DS «DS_UPLOAD_REQ» (см. таблицу D.2), только если набор параметров допустимый и:

- параметры, назначенные для DS, были изменены локально на Устройстве (например, настройка в режиме обучения, человеко-машинный интерфейс и т. п.) или
- устройство получает команду системы ParamDownloadStore (СохранитьСкачанныеПараметры).

С данной информацией о Событии механизм DS на Ведущем узле запускается и инициирует последовательность подкачки в DS.

Механизм DS на Устройстве определяется в конечной машине, приведенной на рисунке 88.

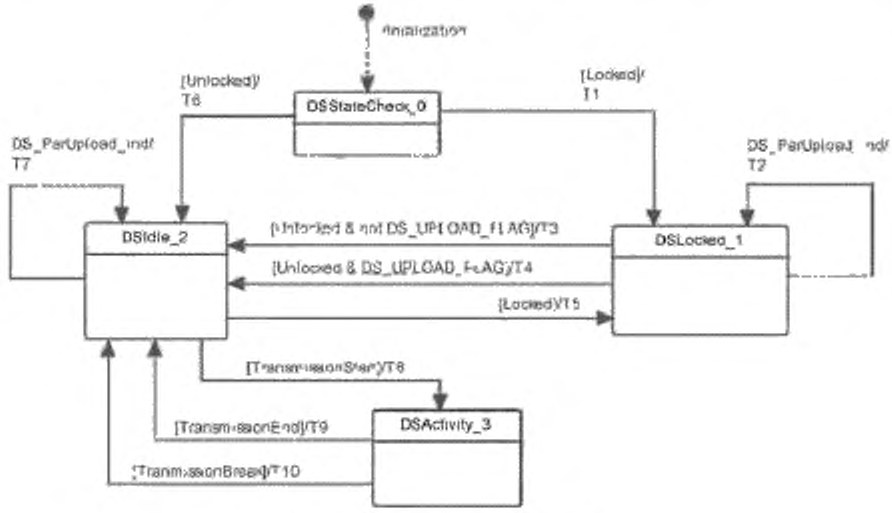


Рисунок 88 — Конечная машина DS

В таблице 96 показаны таблицы перехода состояний конечной машины DS. Назначения Индекса DS подробно показаны в таблице В.10.

Таблица 96 — Таблицы перехода состояний конечной машины DS

Имя состояния		Описание состояния	
DSStateCheck_0		Проверка состояния активации после инициализации	
DSLocked_1		Ожидание разблокировки конечной машины DS	
DSIdle_2		Ожидание действий DS	
DSActivity_3		Предоставить набор параметров; локальная параметризация заблокирована	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	Установить свойство состояния State_Property = «Data Storage access locked» (Доступ к DS заблокирован)
T2	1	1	Установить DS_UPLOAD_FLAG = TRUE
T3	1	2	Установить State_Property = «Inactive»
T4	1	2	Вызвать услугу AL_EVENT.req (EventCode: DS_UPLOAD_REQ); Установить State_Property = «Inactive»
T5	2	1	Установить State_Property = «Data Storage access locked» (Доступ к DS заблокирован)
T6	0	2	Установить State_Property = «Inactive»
T7	2	2	Установить DS_UPLOAD_FLAG = TRUE Вызвать услугу AL_EVENT.req (EventCode: DS_UPLOAD_REQ)
T8	2	3	Блокировать доступ к локальным параметрам. Установить State_Property = «Upload» или «Download»
T9	3	2	Установить DS_UPLOAD_FLAG = FALSE, разблокировать доступ к локальным параметрам. Установить State_Property = «Inactive»
T10	3	2	Разблокировать доступ к локальным параметрам. Установить State_Property = «Inactive»
Внутренние элементы		Тип	Определение
Unlocked		Bool	DS разблокировано, см. В.2.4
Locked		Bool	DS заблокировано, см. В.2.4
DS_ParUpload.ind		Услуга	Внутренняя услуга Устройства между PM и DS (см. рисунок 84)
XTransmissionStart		Bool	Была вызвана команда DS DS_UploadStart или DS_DownloadStart
TransmissionEnd		Bool	Была вызвана команда DS DS_UploadEnd или DS_DownloadEnd
TransmissionBreak		Bool	SM_MODE_INACTIVE или получена команда DS DS_Break

На усеченной диаграмме последовательностей на рисунке 89 показаны важные последовательности связи после параметризации.

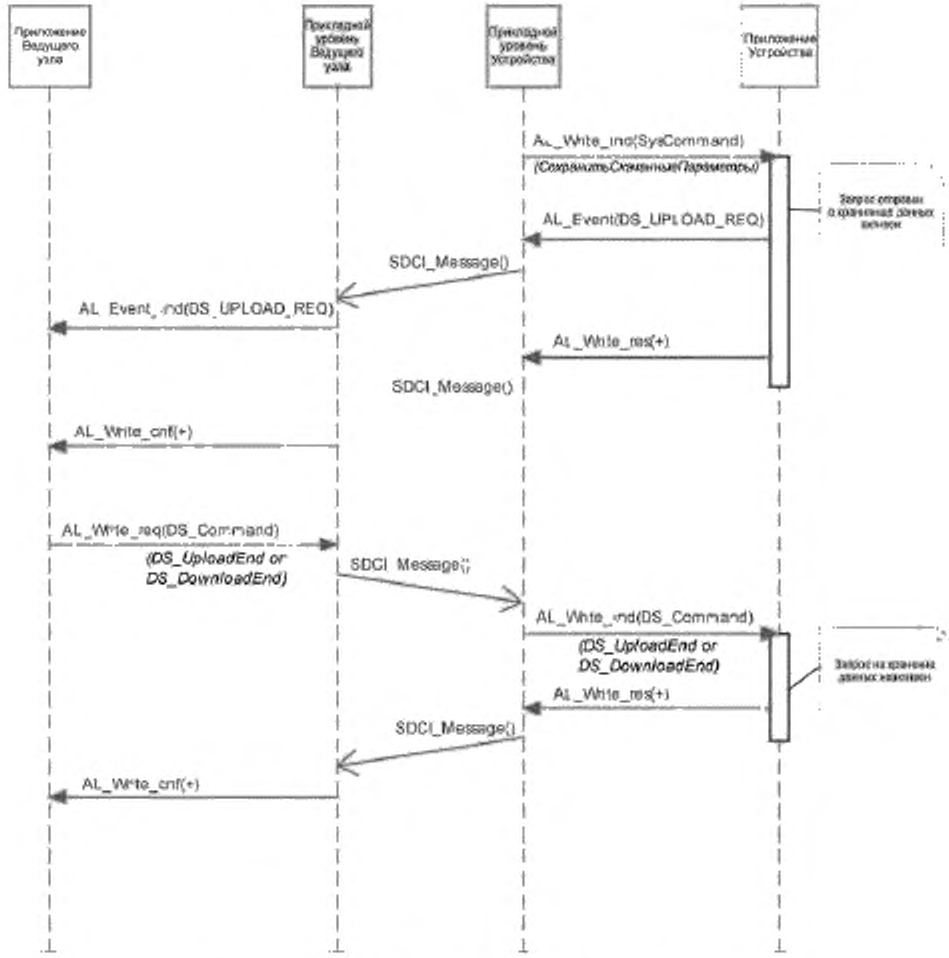


Рисунок 89 — Последовательность сообщений запроса DS

10.4.3 Конфигурирование Хранилища данных

Механизм DS внутри Устройства может быть блокирован через Ведущий узел, например, инструментом или программой PLC. Подробная информация приведена в B.2.4.

Это рекомендуется во время ввода в эксплуатацию или испытаний системы для предотвращения интенсивного обмена данными.

10.4.4 Пространство памяти Хранилища данных

Для обработки затребованного объема данных для DS при любых обстоятельствах количество сохраняемых индексов и требуемый общий объем памяти даются в параметре Size (Размер, см. таблицу B.10). Общая требуемая память (включая структурную информацию) не должна превышать 2048 октетов (см. приложение F). Механизм DS на Ведущем узле в состоянии поддерживать такой объем памяти на каждый порт.

10.4.5 Список индексов Хранилища данных

Устройство является «владельцем» Списка индексов DS (см. таблицу B.10). Назначением Списка индексов является предоставление всей необходимой информации при замене Устройства. Список

индексов DS фиксируется для каждого конкретного идентификатора устройства DeviceID. В противном случае нельзя гарантировать целостности данных между Ведущим узлом и Устройством. Список индексов содержит маркер окончания (см. таблицу В.10), если Устройство не поддерживает DS (см. 10.4.1). В данном случае размер требуемой памяти равен 0.

10.4.6 Доступность параметров Хранилища данных

Все индексы, перечисленные в Списке индексов, доступны для чтения и записи командам системы DS_UploadStart, DS_DownloadStart, DS_UploadEnd и DS_DownloadEnd (см. таблицу В.10). Если хотя бы один из индексов отвергается Устройством, DS на Ведущем узле данных прекращает подкачку или скачивание данных командой системы DS_Break. В данном случае не будет выполняться никаких повторных попыток последовательности DS.

10.4.7 Хранилище данных без ISDU

Поддержка передачи ISDU в Устройстве является непременным условием для DS-параметров. Параметры страницы 2 Непосредственных параметров не могут сохраняться и восстанавливаться механизмом DS.

10.4.8 Индикация изменения параметров Хранилища данных

Контрольная сумма параметров, определенная в таблице В.10, применяется как индикатор изменений в наборе параметров. Настоящий стандарт не требует специальных механизмов для определения изменений параметров. Набор рекомендованных методов представлен в справочном приложении J.

10.5 Диспетчер Событий (ED)

Любое приложение Устройства может генерировать предопределенную информацию о состоянии системы в случае отказа операций SDCI или специальную техническую информацию (диагностику) как результат применения специальных технических методов диагностики. ED преобразует данную информацию в Событие в соответствии с определениями из А.6. Событие состоит из описателя события EventQualifier, указывающего свойства особой ситуации, и идентификатора кода события EventCode ID, представляющего описание данной особой ситуации вместе с возможными восстановительными мерами. В таблице D.1 приводится список предопределенных идентификаторов и описаний для особых ситуаций на уровне приложения. Диапазоны идентификаторов зарезервированы для особых ситуаций, специфических для профиля и специфических для изготовителя. В таблице D.2 содержится список предопределенных идентификаторов для особых ситуаций, специфических для SDCI.

События разделены на Errors (Ошибки), Warnings (Предупреждения) и Notifications (Уведомления). Данная классификация описана в 10.9.2, а в 11.5 описывается, как Ведущий узел контролирует и обрабатывает данные События.

События происходят в конкретный момент времени и подтверждаются единственной командой. Поэтому подтверждение События может быть отложено до самого медленного подтверждения от верхних уровней системы.

10.6 Характеристики Устройства

10.6.1 Общие положения

Следующие характеристики Устройства определяются с некоторой степенью детализации, чтобы достичь общего поведения. Они доступны через стандартизированные специфические для Устройства методы или параметры. Доступность данных характеристик определена в описании устройства I/O (IODD) для Устройства.

10.6.2 Обратная совместимость Устройств

Данная характеристика позволяет Устройству выступать в роли предыдущей редакции Устройства. На стадии запуска SM на Ведущем узле перезаписывает идентификатор Устройства DeviceID (DID), затребованный прежним идентификатором устройства. Техническое приложение Устройства переключается к предыдущему набору функций или подмножеству, назначенному для данного DID. Обратная совместимость Устройства применяется по запросу.

Так как Устройство может обеспечить обратную совместимость к предыдущим DID, такие совместимые Устройства поддерживают все параметры и возможности связи предыдущего идентификатора Устройства. Таким образом, в данном случае Устройству разрешается изменять любые параметры связи и идентификации.

10.6.3 Совместимость редакций протокола

Данная характеристика позволяет Устройству регулировать уровни его протокола к предыдущей версии протокола SDCI, как, например, для версии устаревшего протокола Ведущего узла или в будущем от версии V(x) к версии V(x-n). На стадии запуска SM на Ведущем узле может перезаписать собственную редакцию протокола Устройства RevisionID (RID) в случае несоответствия с редакцией протокола, поддерживаемой Ведущим узлом. Устаревшее устройство не записывает команду Ведущего узла MasterIdent (см. таблицу В.2), и, таким образом, Устройство может приспособиться к устаревшему протоколу (V1.0). Совместимость редакций протокола Устройства применяется по запросу.

10.6.4 Заводские установки

Данная характеристика позволяет Устройству восстанавливать первоначальное состояние поставки. Флаг DS и другие динамические параметры, такие как «Error Count» (Счетчик ошибок, см. В.2.17), «Device Status» (Состояние Устройства, см. В.2.18) и «Detailed Device Status» (Подробное состояние Устройства, см. В.2.19), возвращаются в исходное состояние, когда применяется эта возможность. Данная характеристика не включает параметры, специфические для изготовителя, например, счетчик рабочих часов.

Примечание — В данном случае существующий сохраненный набор параметров на Ведущем узле будет автоматически считан в Устройство после его запуска.

Изготовитель обязан гарантировать правильное функционирование при любых обстоятельствах. Сброс в исходное состояние запускается принятием команды системы «Restore factory settings» (Восстановить заводские установки, см. таблицу В.9). Возврат к заводским установкам для Устройства выполняется по запросу.

10.6.5 Возврат приложения в исходное состояние

Данная характеристика позволяет Устройству восстанавливать специальное техническое приложение. Это особенно полезно, когда специальное техническое приложение должно быть установлено в предопределенное рабочее состояние без прерывания связи и цикла закрытия. Сброс в исходное состояние запускается принятием команды системы «Application reset» (Возврат приложения в исходное состояние, см. таблицу В.9). Возврат специального технического приложения в исходное состояние выполняется для Устройства по запросу.

10.6.6 Сброс Устройства в исходное состояние

Данная характеристика позволяет Устройству выполнять горячий запуск. Данная возможность особенно полезна, когда необходимо сбросить Устройство в начальное состояние, например при включении питания. В данном случае связь будет прервана. Горячий запуск инициируется принятием команды системы «Device reset» (Сброс Устройства в исходное состояние, см. таблицу В.9). Горячий запуск выполняется для Устройства по запросу.

10.6.7 Визуальная индикация в интерфейсе SDCI

Данная характеристика показывает рабочее состояние Устройства в интерфейсе SDCI. Индикация в режиме SDCI устанавливается в 10.9.3. Индикация режима SIO определяется изготовителем и не охватывается этим определением. Функция запускается индикацией SM (во всех состояниях, исключая SM_Idle и SM_SIO на рисунке 79). Индикация SDCI запускается для Устройства по запросу.

10.6.8 Блокировка доступа к параметрам

Данная возможность позволяет глобально блокировать или разблокировать доступ на запись ко всем записываемым параметрам Устройства, доступным через интерфейс SDCI (см. В.2.4). Блокировка запускается принятием команды системы «Device Access Locks» (Блокировки доступа к Устройству, см. таблицу В.8). Поддержка данных функций для Устройства производится по запросу.

10.6.9 Блокировка Хранилища данных

Установка такой блокировки вызывает переключение свойства состояния «State_Property» в таблице В.10 в состояние «Data Storage locked» (DS блокировано), и Устройство не может посылать Событие DS_UPLOAD_REQ запроса подкачки в DS. Поддержка этой функции является обязательной для Устройства, если реализован механизм DS.

10.6.10 Блокировка параметров Устройства

Установка такой блокировки запретит перезапись параметров Устройства через встроенные элементы управления или регулировку таких элементов, как кнопки настройки в режиме обучения (см. В.2.4). Поддержка данных функций для Устройства производится по запросу.

10.6.11 Блокировка интерфейса пользователя Устройства

Установка такой блокировки запретит работу встроенных дисплеев человеко-машинного интерфейса (HMI) и элементов регулировки, таких как кнопки настройки в режиме обучения на Устройстве (см. В.2.4). Поддержка данной функции для Устройства производится по запросу.

10.6.12 Время сдвига

Время сдвига t_{offset} — это параметр, конфигурируемый пользователем (см. B.2.22). Он определяет начало обработки технологических данных Устройства относительно начала цикла M-последовательности, что означает начало сообщения (порта) Ведущего узла.

Данный сдвиг позволяет:

- синхронизировать обработку данных Устройства с циклом (порта) Ведущего узла в пределах определенных ограничений;
- синхронизировать между собой обработку данных различных Устройств на различных портах Ведущего узла;
- выполнять обработку данных различных Устройств на различных портах Ведущего узла с определенными сдвигами.

На рисунке 90 показано распределение сообщений во времени относительно обработки данных в Устройствах.

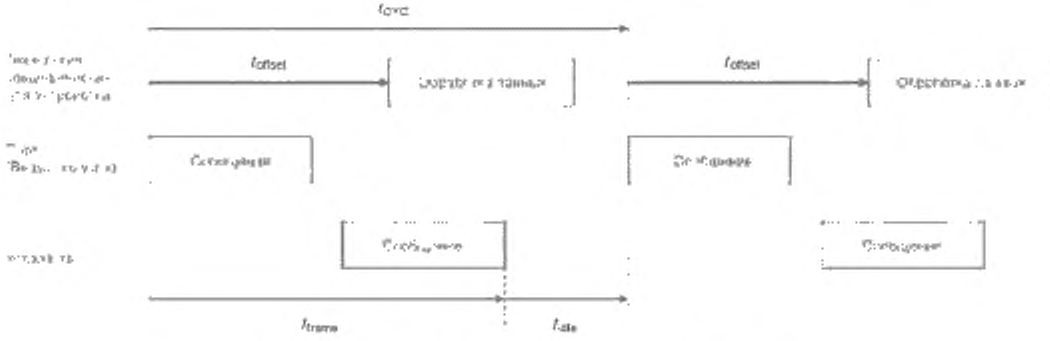


Рисунок 90 — Временная диаграмма цикла

Время сдвига определяет задержку относительно начала цикла M-последовательности. Поддержка этой функции для Устройства производится по запросу.

10.6.13 Концепция Хранилища данных

Механизм DS в Устройстве позволяет автоматически сохранять параметры на сервере DS Ведущего узла и восстанавливать их по уведомлению о Событии. Непротиворечивость данных проверяется в любом направлении на Ведущем узле и Устройстве. DS в основном фокусируется на параметрах конфигурирования настройки Устройства во время ввода в эксплуатацию (см. 10.4 и 11.3). Поддержка данных функций для Устройства производится по запросу.

10.6.14 Блочный параметр

Возможность передачи блочного параметра в Устройство позволяет передавать наборы параметров из программы PLC без проверки целостности отдельных объектов данных. Проверка допустимости и непротиворечивости выполняется в конце передачи блочного параметра для всего набора параметров. Данная функция в основном сосредотачивается на том, чтобы обмен параметрами Устройства производился во время выполнения (см. 10.3). Поддержка данных функций для Устройства производится по запросу.

10.7 Проектные нормы и ограничения Устройства

10.7.1 Общие положения

В дополнение к определению протокола в форме состояний, последовательностей, действий и временных диаграмм требуются дополнительные правила и ограничения для определения поведения Устройств. Обзор основных переменных протокола, разбросанных по всему стандарту, собран в таблице 97 с соответствующими ссылками.

10.7.2 Данные процесса

Канал связи процесса передает циклические PD без всяких помех со стороны каналов связи OD. PDE начинается автоматически после любого переключения Устройства в состояние OPERATE через сообщение из Ведущего узла.

Формат передаваемых данных зависит от Устройства и изменяется от отсутствия октетов данных до 32 октетов в каждом направлении связи.

Рекомендации:

- структуры данных должны подходить для использования приложениями PLC;
- настоятельно рекомендуется следовать правилам, изложенным в Е.3.3 и в [6].

Подробная информация об индикации допустимых и недопустимых PD через флаг PDValid при циклическом обмене данными приводится в А.1.5.

10.7.3 Потеря связи

Проектировщик Устройства обязан определить подходящее поведение Устройства в случае, когда связь с Ведущим узлом потеряна (переход Т10 на рисунке 42 обрабатывает обнаружение потери связи, а подраздел 10.2 определяет последующие действия Устройства).

Примечание — Это особенно важно для исполнительных устройств, таких как клапаны или реле управления двигателем.

10.7.4 Непосредственные параметры

Связь с использованием страниц Непосредственных параметров предоставляет механизм без подтверждения связи для обеспечения правильного получения или допустимости передаваемых параметров. К странице Непосредственных параметров можно получить доступ только октет за октетом (Субиндекс) или к целому объекту (16 октетов). Поэтому целостность параметров длиной более одного октета не может быть гарантирована в случае доступа октет за октетом.

Параметры из страницы Непосредственных параметров не могут сохраняться и восстанавливаться через механизм DS.

10.7.5 Канал связи ISDU

Канал связи ISDU предоставляет мощные средства для передачи параметров и команд (см. раздел В.2).

При использовании данного канала следует учитывать следующие правила (см. рисунок 6):

- индекс 0 недоступен при связи через канал ISDU. Доступ перенаправляется Ведущим звеном к странице 1 Непосредственных параметров, применяя канал связи страниц;
- индекс 1 недоступен при связи через канал ISDU. Доступ перенаправляется Ведущим звеном к странице 2 Непосредственных параметров, применяя канал связи страниц;
- индекс 3 недоступен для прикладной программы PLC. Доступ ограничен только для приложений Ведущего узла (DS);
- после получения запроса ISDU из Ведущего узла Устройство отвечает в течение 5000 мс (см. таблицу 97). Любое нарушение заставляет Ведущий блок снять текущее задание.

10.7.6 Правила для идентификатора Устройства DeviceID, связанные с модификациями Устройства

Устройства с определенными идентификаторами устройства DeviceID и изготовителя VendorID не будут отвергаться при передаче данных и функциональной деятельности. Это применимо к датчикам и исполнительным устройствам. Данные Устройства могут, например, отличаться по:

- длине кабеля;
- материалам корпуса;
- монтажным механизмам;
- другим характеристикам и рабочим условиям.

10.7.7 Константы протокола

В таблице 97 приводится обзор основных констант протокола для Устройств.

Таблица 97 — Обзор констант протокола для Устройств

Переменная системы	Ссылки	Значения	Определение
Время подтверждения ISDU, например, после команды системы	В.2.2	5000 мс	Время от приема ISDU (например, команды системы) до начала ответного сообщения Устройства (см. рисунок 61)
Максимальное число входов в списке индексов IndexList	В.2.3	70	Каждый вход включает Индекс и Субиндекс. Все 70 входов занимают в общем 210 октетов памяти

Окончание таблицы 97

Переменная системы	Ссылки	Значения	Определение
Предустановленные значения неиспользуемых или зарезервированных параметров, например, для идентификаторов функции	Приложение В	0 (для чисел) 0x00 (для символов)	Технические средства устанавливают все неиспользуемые параметры в предустановленные значения
Процедура пробуждения	7.3.2.2	См. таблицы 40 и 41	Минимальная и максимальная длительность и число повторных попыток
Максимальное число повторений MaxRetry	7.3.3.3	2, см. таблицу 44	Максимальное число повторных попыток после ошибок связи
Минимальное время цикла MinCycleTime	А.3.7 и В.1.3	См. таблицы А.11 и В.3	Устройство определяет свое минимальное время цикла для сбора входных или выходных PD
Используемый диапазон Индекса	Раздел В.2	См. таблицу В.8	Данная версия стандарта резервирует некоторые области общим размером 65 535 Индексов
Ошибки и предупреждения	10.9.2	50 мс	Событие с режимом Event appears (Появилось Событие) сохраняется по меньшей мере в течение данного промежутка времени
Счетчик Событий EventCount	8.2.2.11	1	Ограничение для AL_Event.req

10.8 Описание устройства ввода/вывода (IODD)

Файл IODD — это файл, который обеспечивает все необходимые свойства для установки связи и необходимые параметры и их границы для достижения требуемой функции датчика или исполнительного устройства.

Файл IODD является файлом, который формально описывает Устройство.

Файл IODD предоставляется для каждого Устройства и включает всю информацию, необходимую для поддержки настоящего стандарта.

Файл IODD может применяться инженерными средствами для PLC и/или Ведущих узлов в целях идентификации, конфигурирования, описания структур данных для PDE, параметризации и расшировки диагностики конкретного Устройства.

Примечание — Подробное описание языка IODD для описания Устройства приводится в [6].

10.9 Диагностика Устройства

10.9.1 Концепции

В настоящем стандарте в подразделе D.2 представлены только наиболее общие коды событий EventCodes. Целью этой общей диагностической информации является предоставление оператору или специалисту по техническому обслуживанию возможности принятия быстрых мер по устранению отказа без глубокого знания технологии Устройства. Таким образом, текст, связанный с конкретным кодом События EventCode, вместе с диагностической информацией всегда содержит инструкции по исправлению.

Полевая шина, Ведущий узел, шлюзы стремятся отображать только небольшое количество кодов События для верхних уровней системы. Обычно специфический для изготовителя код События, определенный через описание устройства IODD, может быть расшифрован в читабельные инструкции только через PDCT или специфический для изготовителя инструмент использования IODD.

Сжатая информация о «состоянии здоровья» Устройства может быть извлечена из параметра состояния устройства «Device Status» (см. В.2.18). В таблице 98 представлен обзор различных возможностей Устройства и показаны примеры потребителей данной информации.

Если данные возможности реализованы, можно также подсчитывать число отказов после включения питания или восстановления исходного состояния через параметр счетчика ошибок ErrorCount (см. В.2.17) и дополнительную информацию в случае профиля Устройства через параметр подробного состояния Устройства DetailedDeviceStatus (см. В.2.19).

Примечание — Специфические для профиля значения параметра DetailedDeviceStatus приведены в [7].

Если требуется, настоятельно рекомендуется предоставлять «глубокую» специальную техническую диагностическую информацию в форме параметров, зависящих от устройства (см. таблицу В.8), которая может извлекаться через инструменты конфигурирования порта и Устройства для Ведущего узла или через специфические для изготовителя инструменты. Обычно только эксперты или обслуживающий персонал изготовителя в состоянии делать выводы из этой информации.

Таблица 98 — Анализ особых ситуаций при диагностике Устройства

Диагностическая особая ситуация	Появляется/ исчезает	Однократная	Параметр	Получатель	Потребитель
Ошибка (быстрое исправление, стандартные коды Событий)	Да	—	—	PLC или HMI (отображение полевой шины)	Обслуживающий и ремонтный персонал
Ошибка (IODD: коды Событий, зависящие от изготовителя, см. таблицу D.1)	Да	—	—	PDCT или инструмент изготовителя	Обслуживающий персонал изготовителя
Ошибка (через параметры, специфические для Устройства)	—	—	См. таблицу В.8	PDCT или инструмент изготовителя	Обслуживающий персонал изготовителя
Предупреждение (быстрое исправление, стандартные коды Событий)	Да	—	—	PLC или HMI	Обслуживающий и ремонтный персонал
Предупреждение (IODD: коды Событий, зависящие от изготовителя, см. таблицу D.1)	Да	—		PDCT или инструмент изготовителя	Обслуживающий персонал изготовителя
Предупреждение (через параметры, специфические для Устройства)	—	—	См. таблицу В.8		
Уведомление (Стандартные коды Событий)	—	Да		PDCT	Персонал по вводу в эксплуатацию
Подробное состояние Устройства	—	—		PDCT или инструмент изготовителя	Персонал по вводу в эксплуатацию и обслуживающий персонал изготовителя
Число отказов через параметр счетчика ошибок «Error Count»	—	—	См. В.2.17		
«Здоровье» Устройства через параметр состояния устройства «Device Status»	—	—	См. В.2.18, таблица В.13	HMI, инструменты, такие как «Управление активами»	Оператор

10.9.2 События

Значения РЕЖИМА для События назначаются следующим образом (см. А.6.4):

- События ТИПА «Ошибка» применяют РЕЖИМ «Событие появляется/исчезает»;
- События ТИПА «Предупреждение» применяют РЕЖИМ «Событие появляется/исчезает»;
- События ТИПА «Уведомление» применяют РЕЖИМ «Однократное событие».

Применяются следующие требования:

- все События, уже помещенные в очередь Событий, отбрасываются Диспетчером Событий, когда связь прерывается или отменяется.

Примечание — После возобновления связи специальное техническое приложение отвечает за правильную отчетность о причинах текущего События;

- ED ответственен за контроль потока «Событие появляется» и «Событие исчезает». После того как ED послал Событие с РЕЖИМОМ «Событие появляется» для данного кода События, оно не должно посылаться заново для данного кода События, прежде чем было послано Событие с РЕЖИМОМ «Событие исчезает» для данного кода События;
- каждое событие применяет статические атрибуты режима, типа и экземпляра;
- каждый определяемый изготовителем код События уникально назначается одному из ТИПОВ (Ошибка, Предупреждение или Уведомление).

Чтобы предотвратить диагностический канал связи (см. рисунок 6) от «наплыва» сообщений, применяются следующие требования:

- одинаковая диагностическая информация не сообщается менее чем 60 с, т. е. ED не будет вызывать услугу AL_Event с тем же кодом События чаще, чем через 60 с;
- ED не будет выдавать диагностическую информацию «Событие исчезает» раньше, чем через 50 мс после соответствующей информации «Событие появляется»;
- последовательные особые случаи ошибок или предупреждений с одинаковыми коренными причинами будут игнорироваться, это означает, что одна коренная причина приводит к единственной ошибке или сообщению;
- ED не будет вызывать услугу AL_Event со счетчиком событий EventCount, большим единицы;
- ошибки имеют больший приоритет, чем Предупреждения.

На рисунке 91 показано, как обрабатываются две последовательные ошибки и соответствующий поток Событий «Событие появляется»/«Событие исчезает» для каждой ошибки.

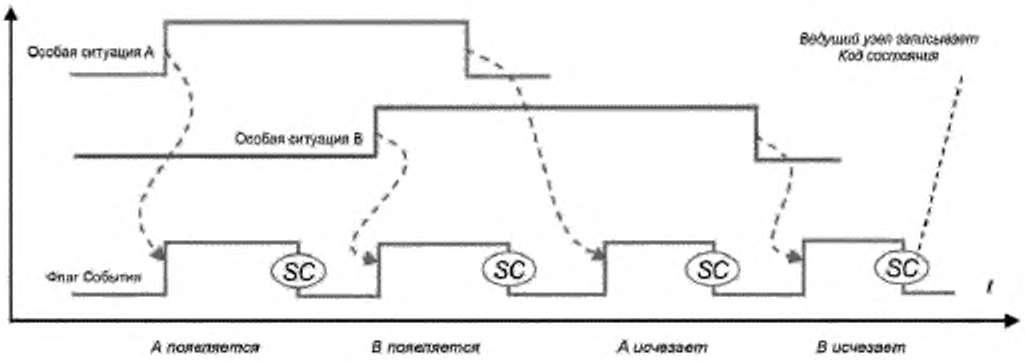


Рисунок 91 — Поток ошибок в случае последовательных ошибок

10.9.3 Визуальные индикаторы

Индикация связи SDCI на Устройстве является необязательной. Индикация SDCI применяет зеленые индикаторы. Индикация следует интервалам времени и спецификации, приведенным на рисунке 92.

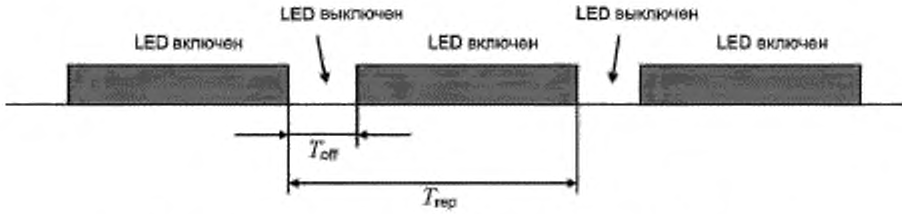


Рисунок 92 — Временные интервалы LED-индикатора Устройства

В таблице 99 определены временные интервалы для LED-индикаторов Устройств.

Таблица 99 — Временные интервалы для LED-индикаторов

Временной интервал	Минимальный	Типовой	Максимальный	Единица измерения
T_{rrep}	750	1000	1250	мс
T_{off}	75	100	150	мс
T_{off}/T_{rrep}	7,5	10	12,5	%

Примечание — Временные интервалы, показанные выше, определены таким образом, что общим ощущением является «питание включено».

Короткие периоды прерывания указывают, что Устройство находится в состоянии связи COMx. Чтобы избежать мерцания, цикл индикации должен начинаться в состоянии «LED off» (LED выключен) и всегда заканчиваться (см. таблицу 99).

10.10 Возможности подключения Устройства

Различные возможности подключения Устройств к портам Ведущего узла, а также соответствующие типа кабеля и цветовое кодирование описаны в 5.5.

Примечание — По причинам совместимости настоящий стандарт не запрещает устройствам SDCI предоставлять дополнительные провода для подключения к функциям за пределами охвата настоящего стандарта (например, для передачи аналоговых выходных сигналов).

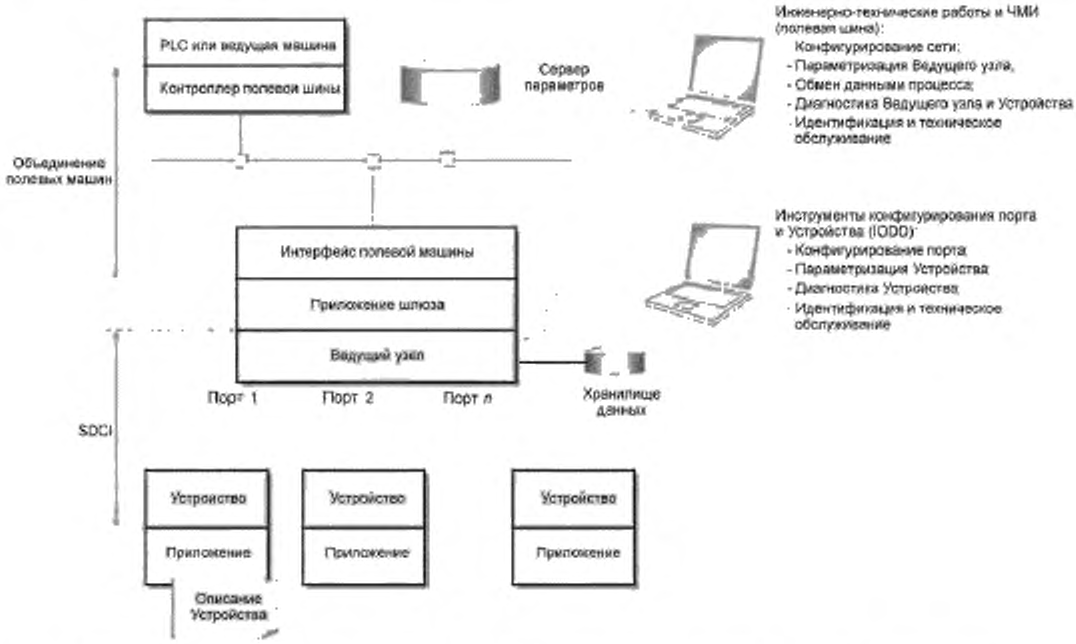
11 Ведущий узел

11.1 Обзор

11.1.1 Обобщенная модель для интеграции системы Ведущего узла

Сфера действия технологии SDCI в иерархии автоматизации уже была рассмотрена в подразделе 4.2.

На рисунке 93 показаны рекомендуемые отношения между технологией SDCI и технологией полевых шин. И хотя данный момент может быть важным для практического применения, это не предполагает автоматически, что технология SDCI зависит от интеграции в системы связи на полевых шинах. Она может также прямо интегрироваться в системы PLC, промышленных PC и других систем управления без связи на полевых шинах между компонентами.



Примечание — Голубые затененные области указывают на характеристики, устанавливаемые в настоящем стандарте.

Рисунок 93 — Обобщенные отношения технологии SDCI и технологии связи на полевых шинах

11.1.2 Структура и услуги Ведущего узла

На рисунке 94 представлен обзор полной структуры и услуг Ведущего узла.

Приложения Ведущего узла включают прежде всего шлюз, определяемый полевой шиной, или прямое соединение с PLC (ведущей машиной) для целей конфигурирования запуска и параметризации,

а также обмена PDE, изменения параметров, контролируемых пользовательской программой во время работы и распространения диагностической информации. Для целей конфигурирования, параметризации и диагностики во время ввода в эксплуатацию так называемый Инструмент конфигурирования порта и Устройства (PDCT, программное обеспечение) подключается непосредственно к Ведущему узлу или через связь на полевых шинах. Данные два инструмента применяют следующие общие приложения Ведущего узла:

- CM, который преобразует назначения конфигурации пользователя в настройки порта;
- ODE, который предоставляет, например, доступ к ациклическим параметрам;
- Механизм DS, который может применяться для сохранения и восстановления параметров Устройства;
- Блок диагностики (DU), который направляет События из AL к устройству DS или к приложению шлюза;
- PDE, устанавливающий связи с инструментами автоматизации более высокого уровня. Данные приложения Ведущего узла предоставляют стандартные методы и функции, общие для всех Ведущих узлов.

CM и механизм DS требуют специальной координации в отношении OD, см. рисунок 95 и рисунок 105.

Применение шлюза отображает данные функции в характеристики конкретной полевой шины, PLC или непосредственно в систему ведущей машины. Определение любого из данных приложений шлюза не входит в сферу действия настоящего стандарта.

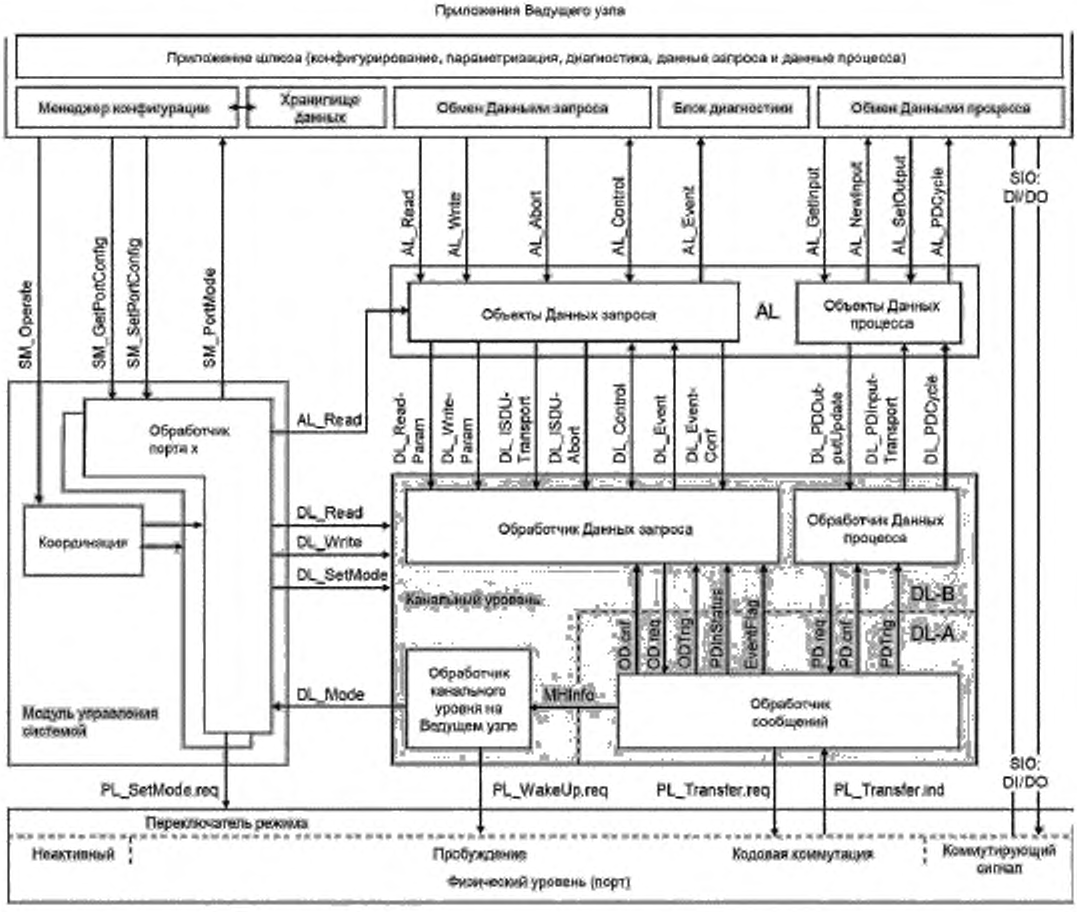


Рисунок 94 — Структура и услуги Ведущего узла

На рисунке 95 показаны взаимоотношения общих приложений Ведущего узла.

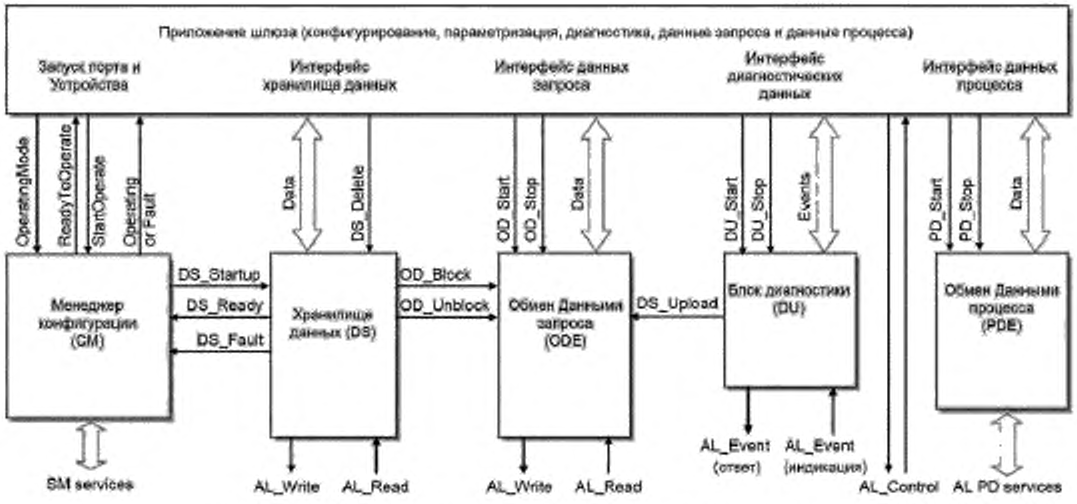


Рисунок 95 — Взаимоотношения общих приложений Ведущего узла

Внутренние переменные между общими приложениями Ведущего узла определены в таблице 100. Главная ответственность возложена на CM, как показано на рисунке 95 и объяснено в 11.2.

Таблица 100 — Внутренние переменные и События для управления общими приложениями Ведущего узла

Внутренняя переменная	Определение
OperatingMode	Данная переменная активирует порт и обеспечивает конфигурирование параметров
ReadyToOperate	Данная переменная указывает на правильную конфигурацию порта
StartOperate	Данная переменная разрешает явный перевод всех портов в режим OPERATE
Operating	Данная переменная указывает, что все порты находятся в режиме циклического обмена PD
Fault	Данная переменная указывает на прекращенную связь COMx в любом из портов (см. рисунок 98 и таблицу 101)
DS_Startup	Данная переменная запускает конечную машину DS, которая, если требуется, передает команды Upload (Отправить) или Download (Скачать) параметрам Устройства (см. 11.3)
DS_Ready	Данная переменная указывает, что DS успешно завершило операцию и находится в рабочем режиме CFGCOM или AUTOCOM (см. 9.2.2.2)
DS_Fault	Данная переменная показывает, что операция DS была прекращена из-за отказа
DS_Delete	Любое подтвержденное изменение конфигурации Устройства ведет к стиранию сохраненного набора данных в DS
DS_Upload	Данная переменная запускает конечную машину DS на Ведущем узле из-за специального События DS_UPLOAD_REQ на Устройстве
OD_Start	Данная переменная разрешает доступ к OD через услуги AL_Read и AL_Write
OD_Stop	Данная переменная указывает, что доступ к OD через услуги AL_Read и AL_Write подтвержден с отрицательным ответом в приложении шлюза
OD_Block	Действия DS по подкачке и скачиванию блокируют доступ к OD через услуги AL_Read или AL_Write. Доступ отклоняется приложением шлюза
OD_Unblock	Данная переменная разрешает доступ к OD через услуги AL_Read или AL_Write
DU_Start	Данная переменная позволяет Блоку диагностики передавать удаленные (на Устройстве) и местные (на Ведущем узле) События приложению шлюза

Окончание таблицы 100

Внутренняя переменная	Определение
DU_Stop	Данная переменная указывает, что События Устройства не передались приложению шлюза и не были подтверждены. Доступные События остаются блокированными до тех пор, пока Блок диагностики не станет снова доступным
PD_Start	Данная переменная разрешает PDE с приложением шлюза
PD_Stop	Данная переменная запрещает PDE с приложением шлюза

11.2 Менеджер конфигурации (CM)

11.2.1 Общие положения

На рисунках 95 и 96 демонстрируется координирующая роль CM среди общих приложений Ведущего узла. После установки порта в назначенные режимы (см. 11.2.2.1—11.2.2.3) CM запускает механизм DS и возвращает переменную «Operating» или «Fault» приложению шлюза.

В случае переменной «Operating» конкретного порта приложение шлюза активирует конечные машины соответствующих Блока диагностики, ODE и PDE.

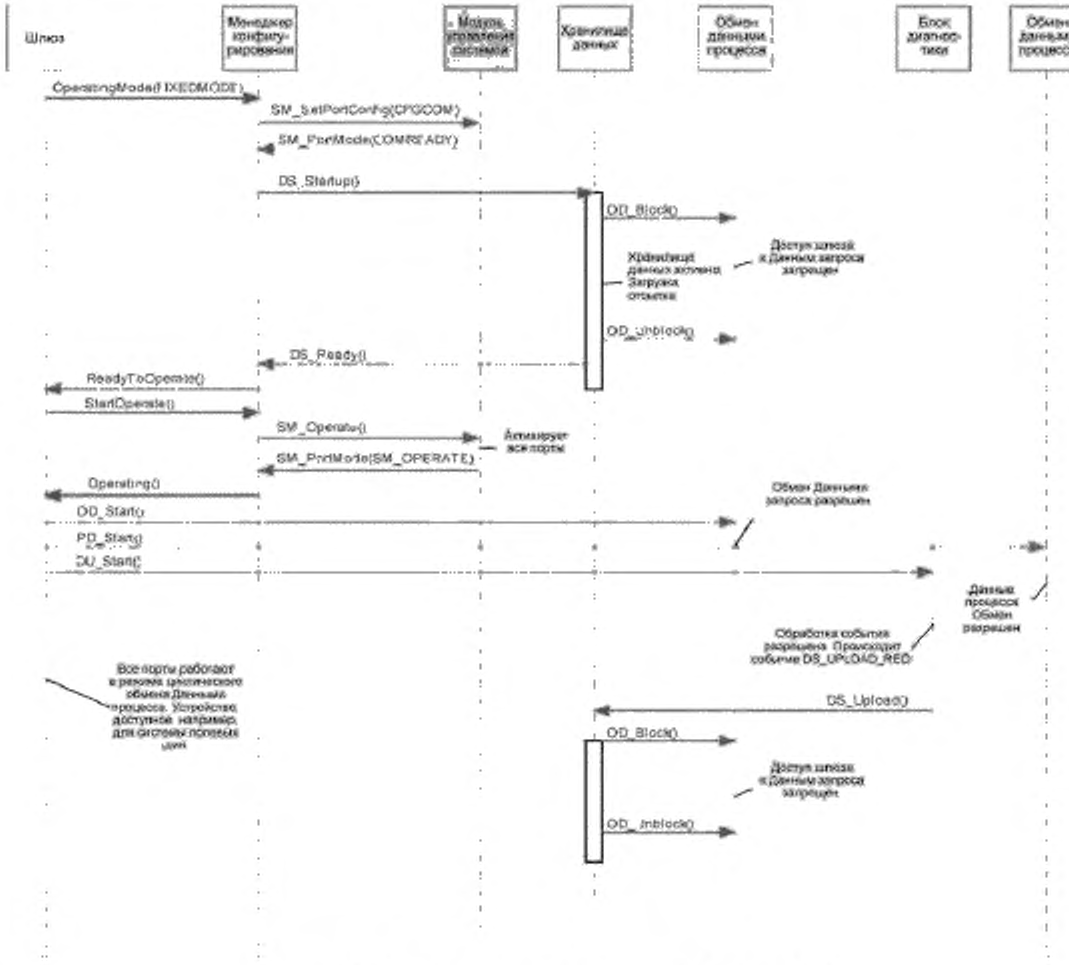


Рисунок 96 — Диаграмма последовательностей действий CM

После того как все порты SDCI готовы к работе (состояние ReadyToOperate, см. рисунок 96), приложение шлюза активирует все порты (StartOperate), обеспечивая синхронизацию циклов портов. Устройства обмениваются PD (Operating). В случае отказов приложение шлюза получает сообщение о прекращении связи «Communication abandoned» (Связь прекращена) (INACTIVE или COMLOST).

В случае вызова услуги SM_PortMode (COMP_FAULT, REVISION_FAULT или SERNUM_FAULT) в соответствии с 9.2.3 будет активирована только машина Обмена OD ODE для разрешения параметризации.

При каждом новом запуске порта приложение шлюза вначале деактивирует (например, командой OD_Stop) связанные машины Блока диагностики DU, ODE и PDE.

Некоторые параметры доступны МС для достижения специального поведения.

11.2.2 Параметр конфигурирования

11.2.2.1 Рабочий режим OperatingMode

Может быть выбран один из следующих рабочих режимов. Все режимы являются обязательными.

Режим INACTIVE

Порт SDCI деактивируется, и соответствующая длина PD для ввода и вывода обнуляется. Ведущий узел не предпринимает никаких действий на данном порту.

Режим DO

Порт SDCI конфигурируется как цифровой вывод (ограничения показаны в таблице 2). Длина выходных PD равна одному биту. Ведущий узел не предпринимает попыток пробуждения каких-либо Устройств на этом порту.

Режим DI

Порт SDCI сконфигурирован как цифровой ввод. Длина входных PD равна одному биту. Ведущий узел не предпринимает попыток пробуждения каких-либо Устройств на этом порту.

Режим FIXEDMODE

Порт SDCI сконфигурирован для непрерывной связи. Проверяется определенная идентификация. Будет или не будет разница в идентификации Устройства приводить к отклонению Устройства, зависит от конфигурации порта (уровня проверки InspectionLevel, см. таблицу 78).

Режим SCANMODE

Порт SDCI сконфигурирован для непрерывной связи. Идентификация повторно считывается из Устройства и может предоставляться как заново определенная идентификация. В остальных случаях операционный режим подобен режиму «FIXEDMODE».

11.2.2.2 Цикл порта PortCycle

Может быть выбран один из следующих режимов цикла порта. Ни один из режимов не является обязательным.

Режим свободного хода FreeRunning

Время цикла порта не ограничено.

Режим фиксированного значения FixedValue

Длительность цикла порта ограничена конкретным значением. Если Устройство не в состоянии достичь данного значения интервала, например если временной интервал меньше, чем MinCycleTime Устройства, то генерируется ошибка. Фиксированное значение может быть записано в параметр CycleTime, как показано в 11.2.2.3.

Режим синхронизации сообщений MessageSync

Время цикла порта ограничено синхронным началом всех сообщений портов SDCI данного Ведущего узла. В данном случае время цикла определяется наибольшим минимальным временем цикла MinCycleTime подключенных Устройств. Все порты Ведущего узла, установленные в данный режим, работают с этим поведением, как показано на рисунке 97. Значения смещения и флуктуаций должны быть указаны в руководстве пользователя.

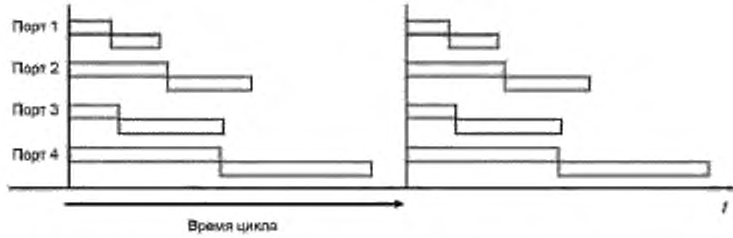


Рисунок 97 — Порты в режиме синхронизации сообщений MessageSync

11.2.2.3 Параметр CycleTime

Данный параметр содержит затребованное или фактическое время цикла для конкретных портов. Он передается как значение с разрешением 100 мкс.

11.2.2.4 Набор параметров PDConfig

Данный набор параметров содержит правила для отображения PD между потоком PD Устройства и потоком PD шлюза (см. пример определений на рисунке 107).

Параметр LenIn

Данный параметр содержит требуемую длину входных PD Устройства в битах.

Параметр PosIn

Данный параметр содержит сдвиг в потоке входных PD шлюза в битах.

Параметр SrcOffsetIn

Данный параметр содержит сдвиг в потоке входных PD Устройства в битах.

Параметр LenOut

Данный параметр содержит требуемую длину выходных PD Устройства в битах.

Параметр PosOut

Данный параметр содержит сдвиг в потоке выходных PD шлюза в битах.

Режим SrcOffsetOut

Данный параметр содержит сдвиг в потоке выходных PD Устройства в битах.

11.2.2.5 Набор параметров идентификации Устройства DeviceIdentification

Данный набор параметров содержит фактическую сконфигурированную идентификацию Устройства.

Параметр VendorID

Данный параметр содержит затребованный или считанный специфический для изготовителя идентификатор, как установлено в B.1.8.

Параметр DeviceID

Данный параметр содержит затребованный или считанный специфический для Устройства идентификатор, как установлено в B.1.9.

Параметр SerialNumber

Данный параметр содержит затребованный или считанный заводской номер, как установлено в B.2.13.

Параметр InspectionLevel

Данный параметр содержит требуемый уровень проверки, как определено в таблице 78.

11.2.2.6 Набор параметров конфигурации хранилища данных DataStorageConfig

Данный набор единичных параметров содержит установки механизма DS.

Параметр состояния активации ActivationState

Данный параметр содержит требуемые состояния механизма DS для данного порта. Поддерживаются следующие режимы:

Режим DS_Enabled

Механизм DS активен и предоставляет полную функциональность, как определено в 11.3.2.

Режим DS_Disabled
Механизм DS неактивен, и полный набор параметров данного порта остается сохраненным.

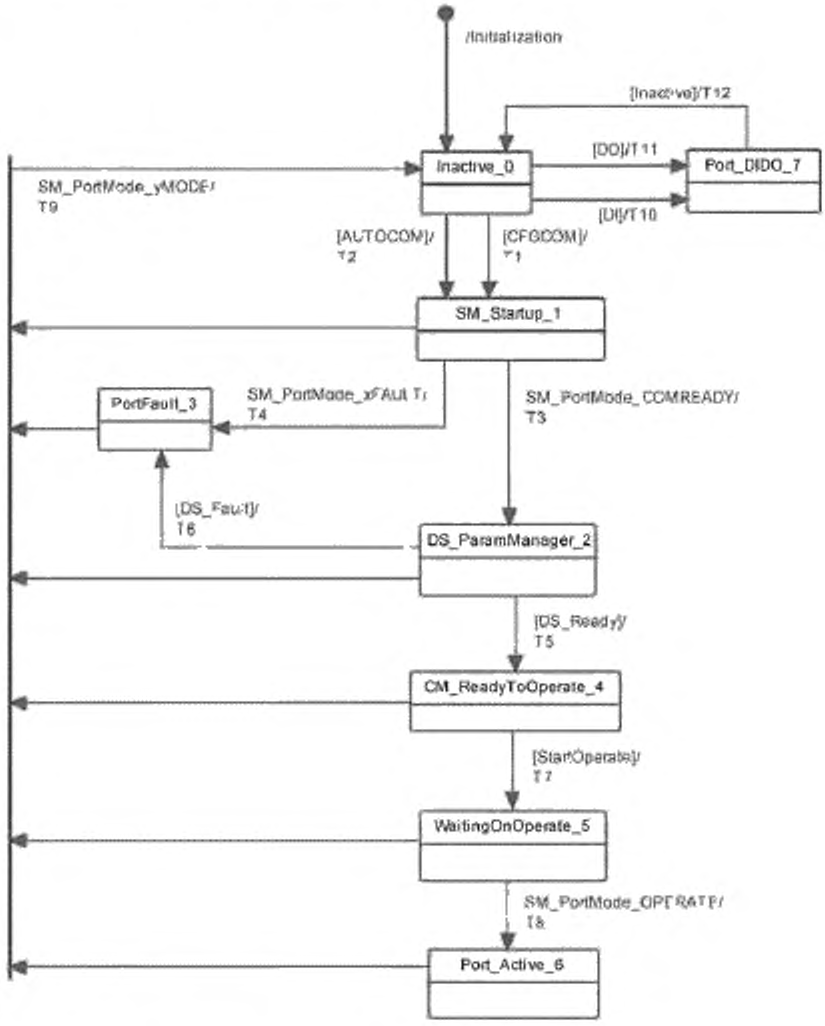
Режим DS_Cleared
Механизм DS отключен, и сохраненный набор параметров данного порта очищается.

DownloadEnable
Механизму DS разрешается скачивать данные на подключенное Устройство.

UploadEnable
Механизму DS разрешается подкачивать данные с подключенного Устройства.

11.2.3 Конечная машина Менеджера конфигурации

На рисунке 98 показана конечная машина CM Ведущего узла.



Обозначения:
xFAULT: REV_FAULT или COMP_FAULT или SERNUM_FAULT
yMODE: INACTIVE или COMLOST.

Рисунок 98 — Конечная машина CM

Различные состояния показывают шаги команд, необходимых для установки или поддержания связи и состояний DI и DO.

Любое изменение конфигурации порта может быть активировано изменением переменной рабочего режима OperatingMode (см. 11.2.2.1).

В таблице 101 показаны таблицы перехода состояний конечной машины СМ.

Таблица 101 — Таблицы перехода состояний конечной машины СМ

Имя состояния		Описание состояния	
Inactive_0		Ожидание любой из переменных рабочего режима из приложения шлюза: DO, DI, AUTOCOM или CFGCOM	
SM_Startup_1		Ожидание установления связи или потери связи или любого из отказов REVISION_FAULT, COMP_FAULT или SERNUM_FAULT (см. таблицу 83)	
DS_Param Manager_2		Ожидание завершения запуска DS. Параметры скачиваются в Устройство или подкачиваются из Устройства	
PortFault_3		Устройство в состоянии PREOPERATE (передача данных). Однако произошел один из трех отказов REVISION_FAULT, COMP_FAULT, SERNUM_FAULT или DS_Fault	
CM_ReadytoOperate_4		Порт ожидает, пока приложение шлюза покажет StartOperate (Начало работы)	
WaitingOnOperate_5		Ожидание переключения SM в состояние OPERATE	
PortActive_6		Порт в состоянии OPERATE. Приложение шлюза обменивается PD и готово послать или принять OD	
PortDIDO_7		Порт в режиме DI или DO. Приложение шлюза обменивается PD (DI или DO)	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	SM_SetPortConfig_CFGCOM
T2	0	1	SM_SetPortConfig_AUTOCOM
T3	1	2	DS_Startup: Запущена конечная машина DS
T4	1	3	Индикация «Fault» (Отказ) приложению шлюза (REVISION_FAULT, COMP_FAULT или SERNUM_FAULT), см. рисунок 95
T5	2	4	Индикация приложению шлюза: ReadyToOperate (Готов к работе)
T6	2	3	Отказ DS: Откат к предыдущему набору параметров
T7	4	5	SM_Operate
T8	5	6	Индикация приложению шлюза: «Operating» (В работе, см. рисунок 96)
T9	1, 2, 3, 4, 5, 6	0	SM_SetPortConfig_INACTIVE. Индикация отказа (Отказ) приложению шлюза: COMLOST или INACTIVE
T10	0	7	SM_SetPortConfig_DI. Индикация приложению шлюза: DI
T11	0	7	SM_SetPortConfig_DO. Индикация приложению шлюза: DO
T12	7	0	SM_SetPortConfig_INACTIVE
Внутренние элементы		Тип	Определение
DS_Ready		Bool	Последовательность (подкачка, скачивание) DS завершена. Порт в рабочем режиме FIXEDMODE или SCANMODE. См. таблицу 100
DS_Fault		Bool	См. таблицу 100
StartOperate		Bool	Приложение шлюза заставляет порт переключиться в режим OPERATE
FIXEDMODE		Bool	Один из режимов работы (см. 11.2.2.1)
SCANMODE		Bool	Один из режимов работы (см. 11.2.2.1)
DI		Bool	Один из режимов работы (см. 11.2.2.1)
DO		Bool	Один из режимов работы (см. 11.2.2.1)

11.3 Хранилище данных (DS)

11.3.1 Обзор

В настоящем стандарте устанавливается DS между Ведущим узлом и Устройством, в то время как механизмы DS на соответствующих верхних уровнях зависят от конкретной полевой шины или системы. Устройство поддерживает стандартизованный набор объектов, предоставляющий информацию о параметрах DS (таких как требования к объему памяти), управляющую информацию и информацию о состояниях механизма DS. Изменения наборов параметров DS обнаруживаются через контрольную сумму параметров «Parameter Checksum» (см. пункт 10.4.8).

11.3.2 Объекты данных DS

Структура объектов данных DS устанавливается в таблице F.1.

Ведущий узел всегда сохраняет информацию заголовков [контрольная сумма параметров Parameter Checksum, идентификатор изготовителя VendorID (ИдИзготовителя) и идентификатор Устройства DeviceID] для целей проверки и контроля. Информация об объектах (объекты 1 … *n*) сохраняется в энергонезависимой памяти Ведущего узла (см. приложение F). До скачивания объектов данных (блока параметров) DS Ведущий узел проверяет целостность информации заголовков для конкретного Устройства.

Максимально допустимый размер объектов данных DS равен 2×2^{10} октетов. Если реализован механизм DS, Ведущий узел обязан предоставлять по меньшей мере такой объем памяти для каждого порта.

11.3.3 Конечная машина DS

Механизм DS вызывается сразу после установления связи COMx, перед входом в режим OPERATE. В это время любая другая связь с Устройством отвергается шлюзом.

На рисунке 99 показана конечная машина механизма DS.

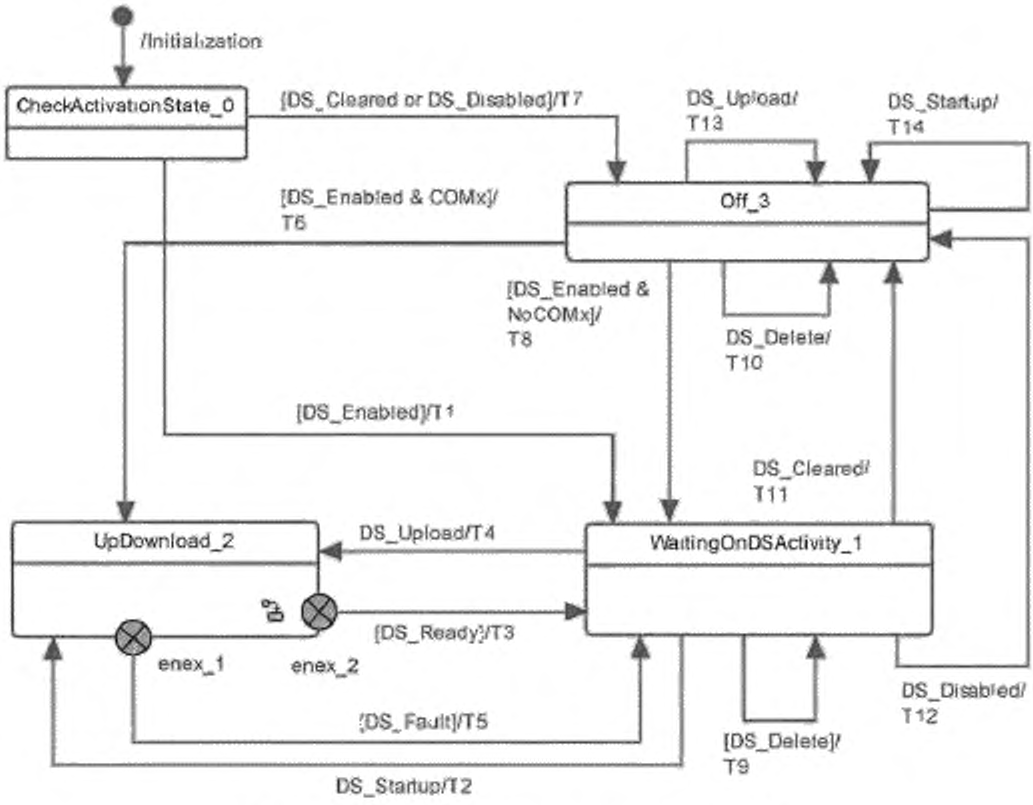


Рисунок 99 — Основная конечная машина механизма DS

На рисунке 100 показан функциональный блок состояния UpDownload_2.

Данный функциональный блок может быть вызван механизмом DS или во время выполнения запущен Событием «DS_UPLOAD_REQ» Event.

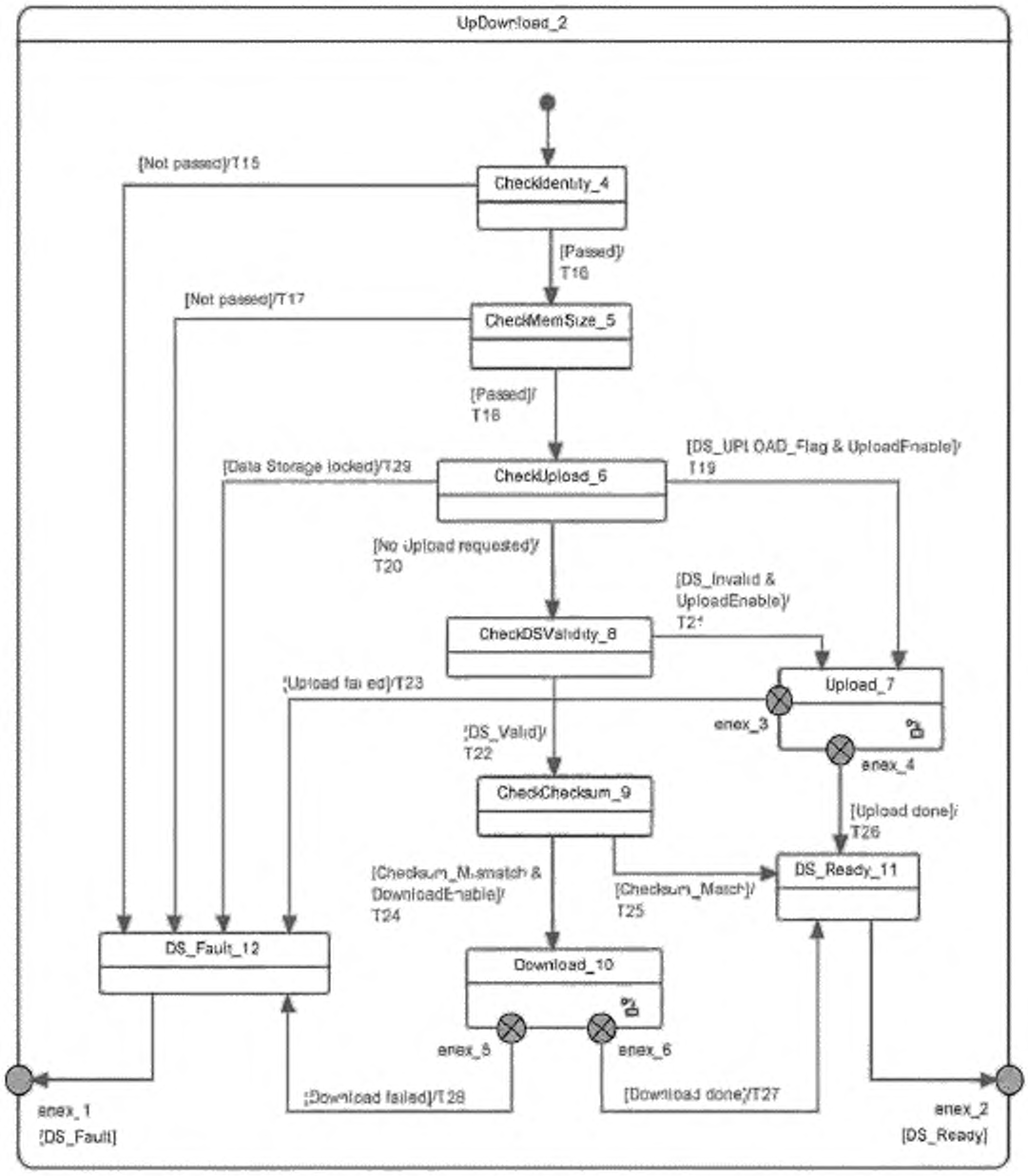


Рисунок 100 — Функциональный блок UpDownload_2 механизма DS

На рисунке 101 показан функциональный блок состояния Upload_7.

Данная конечная машина может быть вызвана механизмом DS или во время выполнения запущена Событием DS_UPLOAD_REQ.

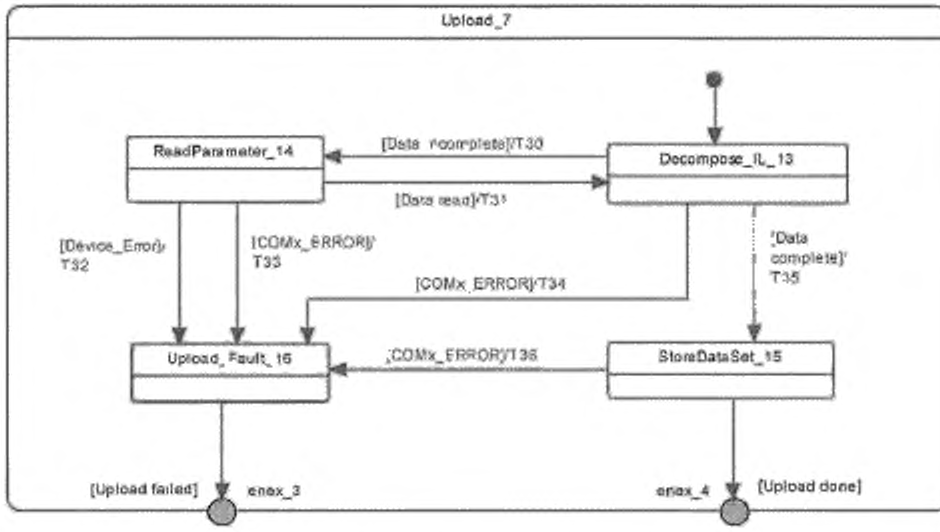


Рисунок 101 — Функциональный блок Upload_7 механизма DS

На рисунке 102 демонстрируется последовательность подкачки в DS с использованием Индекса DS, установленного в В.2.3 и таблице В.10. Структура Списка Индексов Index_List устанавливается в таблице В.11. Флаг DS_UPLOAD_FLAG сбрасывается в конце каждой последовательности (см. таблицу В.10).

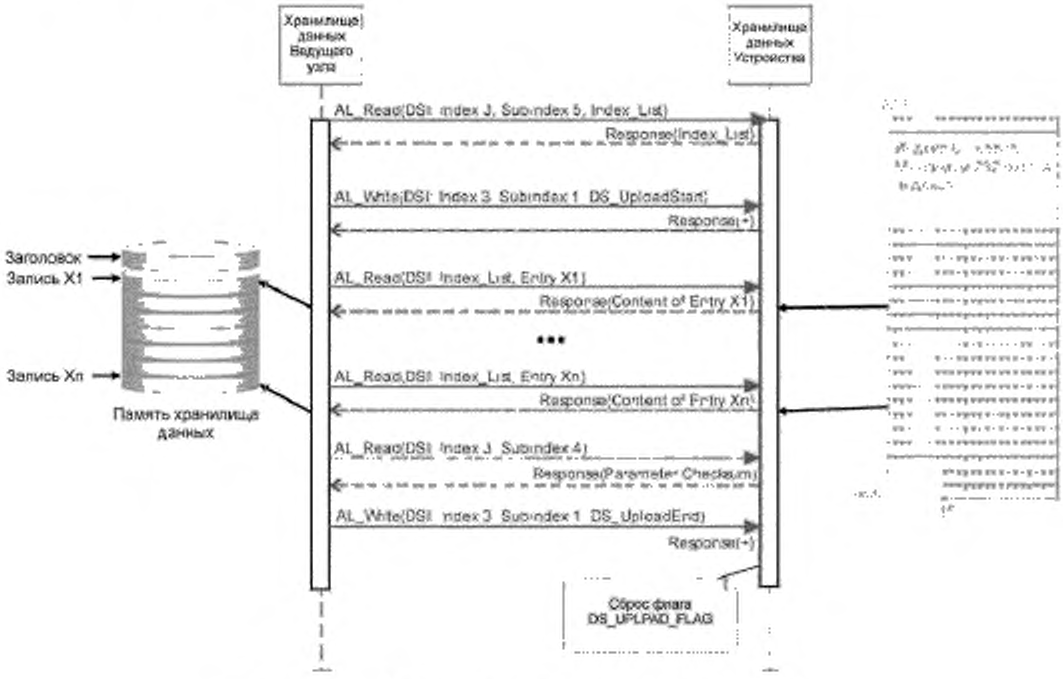


Рисунок 102 — Диаграмма последовательности подкачки в DS

На рисунке 103 показан функциональный блок состояния Download_10. Данная конечная машина может вызываться механизмом DS.

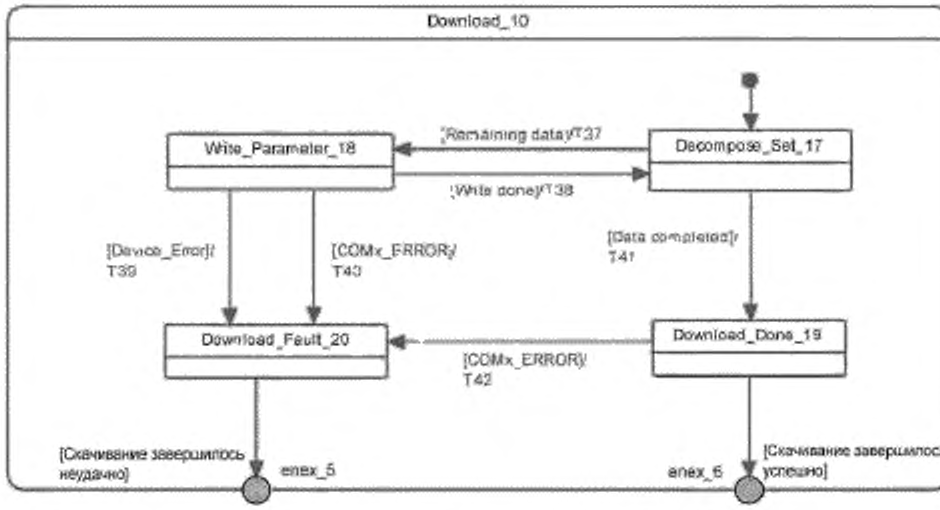


Рисунок 103 — Функциональный блок Download_10 DS

На рисунке 104 демонстрируется последовательность скачивания из DS с использованием Индекса DS, установленного в В.2.3 и таблице В.10. Структура списка Индексов Index_List устанавливается в таблице В.11. Флаг DS_UPLOAD_FLAG сбрасывается в конце каждой последовательности (см. таблицу В.10).

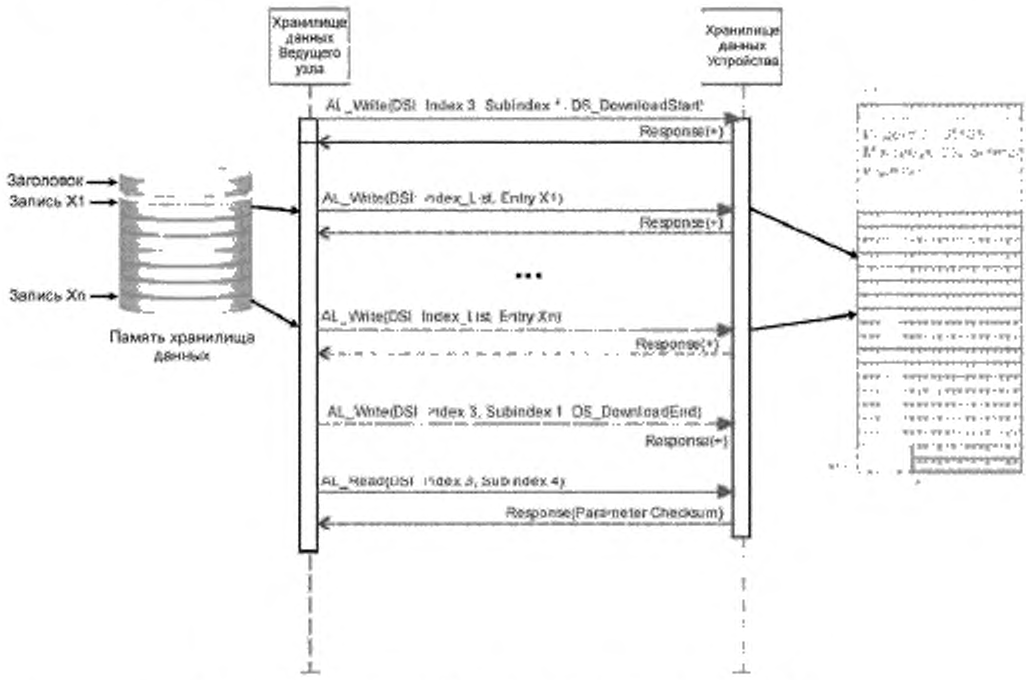


Рисунок 104 — Диаграмма последовательности скачивания из DS

В таблице 102 показаны состояния и переходы конечных машин DS.

Таблица 102 — Состояния и переходы конечных машин DS

Имя состояния	Описание состояния
CheckActivationState_0	Проверить текущее состояние конфигурации DS. Независимо от состояния связи ожидается запрос DS_Startup из управления конфигурацией или Событие DS_UPLOAD_REQ
WaitingOnDSActivity_1	Ожидание запроса подкачки, запуск Устройства, любые изменения состояния активации независимо от состояния связи Устройства
UpDownload_2	Функциональный блок для действий и проверки по подкачке или скачиванию
Off_3	Обработка DS отключена или деактивирована
SM: CheckIdentity_4	Сравнение идентификации Устройства (ИдУстройства, ИдИзготовителя) с набором параметров в Хранилище данных (см. таблицу F.2). Пустое содержание не приводит к отказу
SM: CheckMemSize_5	Сравнение размера набора данных (Индекс 3, Субиндекс 3) с доступным размером памяти Ведущего узла
SM: CheckUpload_6	Проверка флага DS_UPLOAD_FLAG в Индексе DS (см. таблицу B.10)
SM: Upload_7	Функциональный блок для действий по подкачке
SM: CheckDSValidity_8	Проверка, являются ли сохраненные на Ведущем узле данные допустимыми или недопустимыми. Ведущий узел может переходить от действий по подкачке к действиям по скачиванию. Проектировщик Ведущего узла обязан реализовать механизм проверки допустимости данных в соответствии с выбранным сценарием использования
SM: CheckChecksum_9	Проверка различий между содержанием набора данных и параметрами Устройства через контрольную сумму параметров в Индексе DS (см. таблицу B.10)
SM: Download_10	Функциональный блок для действий по скачиванию
SM: DS_Ready_11	Подготовка индикации DS_Ready для CM
SM: DS_Fault_12	Подготовка индикации DS_Fault из Identification_Fault, SizeCheck_Fault, Upload_Fault и Download_Fault для CM
SM: Decompose_IL_13	Чтение Списка индексов в Индексе DS (см. таблицу B.10). Читать содержимое Списка индексов из Устройства вход за входом (см. таблицу B.11)
SM: ReadParameter_14	Ожидание завершения чтения содержимого одного входа Списка индексов из Устройства
SM: StoreDataSet_15	Задача приложения шлюза: сохранение всего набора данных в соответствии с таблицами F.1 и F.2
SM: Upload_Fault_16	Подготовка индикации Upload_Fault из Device_Error и COM_ERROR как входа для индикации ошибки DS DS_Fault более высокого уровня
SM: Decompose_Set_17	Запись набора данных параметр за параметром в Устройство в соответствии с таблицей F.1
SM: Write_Parameter_18	Ожидание окончания записи одного параметра набора данных в Устройство
SM: Download_Done_19	Скачивание завершено. Считать обратно контрольную сумму параметров из Индекса DS в соответствии с таблицей B.10. Сохранить это значение в хранящемся наборе данных в соответствии с таблицей F.2
SM: Download_Fault_20	Подготовка индикации Download_Fault из Device_Error и COM_ERROR как входа для индикации ошибки DS DS_Fault более высокого уровня

Продолжение таблицы 102

Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	2	—
T3	2	1	OD_Unblock; показать состояние DS_Ready MC
T4	1	2	Подтвердить Событие DS_UPLOAD_REQ
T5	2	1	DS_Break (AL_Write, Index 3, Subindex 1); очистить промежуточные данные (очистка мусора); вернуть предыдущее состояние параметров; DS_Fault (см. рисунок 95); OD_Unblock
T6	3	2	—
T7	0	3	—
T8	3	1	—
T9	1	1	Очистить сохраненный набор параметров (см. таблицы F.1 и F.2)
T10	3	3	Очистить сохраненный набор параметров (см. таблицы F.1 и F.2)
T11	1	3	Очистить сохраненный набор параметров (см. таблицы F.1 и F.2)
T12	1	3	—
T13	3	3	Подтвердить Событие DS_UPLOAD_REQ; никаких дальнейших действий
T14	3	3	Послать Сообщение DS_Ready в CM
T15	4	12	Послать сообщение DS_Fault(Identification_Fault) в приложение шлюза
T16	4	5	Считать Data Storage Size в соответствии с таблицей B.10, OD_Block
T17	5	12	Послать сообщение DS_Fault(SizeCheck_Fault) в приложение шлюза
T18	5	6	Считать DS_UPLOAD_FLAG в соответствии с таблицей B.10
T19	6	7	Индекс DS 3, Субиндекс 1: DS_UploadStart (см. таблицу B.10)
T20	6	8	—
T21	8	7	Индекс DS 3, Субиндекс 1: DS_UploadStart (см. таблицу B.10)
T22	8	9	—
T23	7	12	Индекс DS 3, Субиндекс 1: DS_Break (см. таблицу B.10). Показать DS_Fault(Upload) в приложение шлюза
T24	9	10	Индекс DS 3, Субиндекс 1: DS_DownloadStart (см. таблицу B.10)
T25	9	11	—
T26	7	11	Индекс DS 3, Субиндекс 1: DS_UploadEnd; считать контрольную сумму параметров (см. таблицу B.10)
T27	10	11	—
T28	10	12	Индекс DS 3, Субиндекс 1: DS_Break (см. таблицу B.10). Послать сообщение DS_Fault(Download) в приложение шлюза

Продолжение таблицы 102

Переход	Исходное состояние	Конечное состояние	Действие
T29	6	12	Послать сообщение DS_Fault(Data Storage locked) в приложение шлюза
T30	13	14	AL_Read (Index List)
T31	14	13	—
T32	14	16	—
T33	14	16	—
T34	13	16	—
T35	13	15	Считать контрольную сумму параметров Parameter Checksum (см. таблицу B.10)
T36	15	16	—
T37	17	18	Записать параметр через услугу AL_Write
T38	18	17	—
T39	18	20	—
T40	18	20	—
T41	17	19	Индекс DS 3, Субиндекс 1: DS_DownloadEnd (см. таблицу B.10). Считать контрольную сумму параметров Parameter Checksum (см. таблицу B.10)
T42	19	20	—
Внутренние элементы		Тип	Определение
DS_Cleared		Bool	Обработка DS отключена, см. 11.2.2.6
DS_Disabled		Bool	Обработка DS деактивирована, см. 11.2.2.6
DS_Enabled		Bool	Обработка DS активирована, см. 11.2.2.6
COMx_ERROR		Bool	Обнаружена ошибка связи COMx
Device_Error		Bool	В доступе к Индексу отказано. AL_Read или AL_Write.cnf(–) с кодом ошибки 0x80
DS_Startup		Переменная	Запуск из конечной машины MC, см. рисунок 95
NoCOMx		Bool	Нет связи COMx
COMx		Bool	Связь COMx работает правильно
DS_UPLOAD_REQ		Событие	См. таблицу D.2
UploadEnable		Bool	Конфигурация обработки DS активирована, см. 11.2.2.6
DownloadEnable		Bool	Конфигурация обработки DS активирована, см. 11.2.2.6
DS_Valid		Bool	На Ведущем узле доступен допустимый набор параметров. См. описание состояния SM:CheckDSValidity_8
DS_Invalid		Bool	На Ведущем узле отсутствует допустимый набор параметров. См. описание состояния SM:CheckDSValidity_8

Окончание таблицы 102

Внутренние элементы	Тип	Определение
Checksum_Mismatch	Bool	Считанная из Устройства контрольная сумма параметров Parameter Checksum не соответствует контрольной сумме в DS (двоичное сравнение)
Checksum_Match	Bool	Считанная из Устройства контрольная сумма параметров Parameter Checksum соответствует контрольной сумме в DS (двоичное сравнение)

11.3.4 Выбор параметров DS

Проектировщик устройства определяет параметры, являющиеся частью механизма DS.

Описание данных IODD отмечает все параметры, не включенные в DS атрибутом excludedFromDataStorage (исключен из DS). Тем не менее механизм DS не рассматривает информацию из описания данных IODD, а учитывает Список параметров, считанный из Устройства.

11.4 Обмен Данными запроса (ODE)

На рисунке 105 показана конечная машина ODE на Ведущем узле. Данное поведение является обязательным для Ведущего узла.

Во время активной передачи данных механизмом DS все запросы OD блокируются.

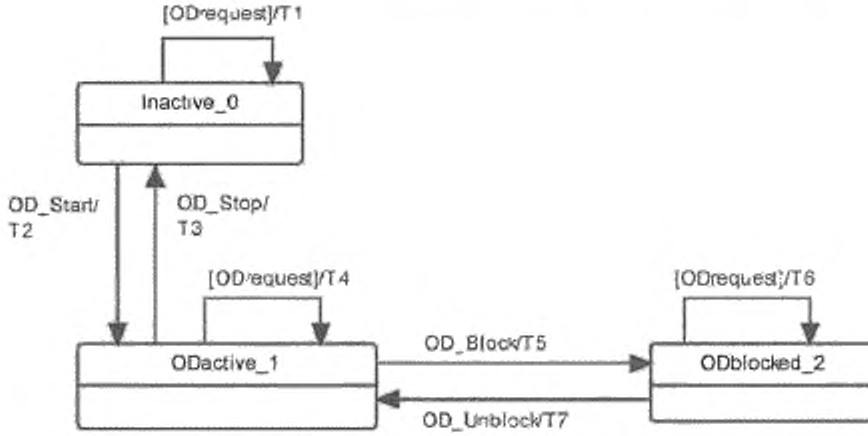


Рисунок 105 — Конечная машина Обмена OD

В таблице 103 показаны переходы состояний конечной машины Обмена OD.

Таблица 103 — Переходы состояний конечной машины ODE

Имя состояния		Описание состояния	
Inactive_0		Ожидание активации	
ODactive_1		Передача OD активна при использовании услуги AL_Read или AL_Write	
ODblocked_2		Передача OD блокирована	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	0	Доступ блокирован (неактивен): посылает сообщение «Service not available» (Услуга недоступна) приложению шлюза

Окончание таблицы 103

Переход	Исходное состояние	Конечное состояние	Действие
T2	0	1	—
T3	1	0	—
T4	1	1	Услуга AL_Read или AL_Write
T5	1	2	—
T6	2	2	Доступ временно блокирован: посылает сообщение «Service not available» (Услуга недоступна) приложению шлюза
T7	2	1	—
Внутренние элементы		Тип	Определение
ODrequest		Переменная	Чтение или запись OD, затребованных через услуги AL_Read или AL_Write

11.5 Блок диагностики (DU)

DU направляет События из AL к устройству DS или к приложению шлюза. Данные события содержат главным образом диагностическую информацию.

Основное назначение диагностической информации — эффективно уведомить оператора об опасности. Это означает:

- отсутствие переполнения диагностической информации;
- сообщение о корневых причинах особой ситуации в Устройстве или Ведущем узле и отсутствие последовательных связанных отказов;
- диагностическая информация сообщает, как обслужить или отремонтировать задействованный компонент для быстрого восстановления системы автоматики.

В интерфейсе SDCI диагностическая информация Устройств передается Ведущему узлу через Сообщения, состоящие из Описателей События EventQualifiers и Кодов Событий EventCodes (см. A.6). Соответствующий человекочитаемый текст доступен для стандартизированных кодов Событий в настоящем стандарте (см. приложение D) и для определяемых изготовителем кодов событий в соответствующем файле IODD Устройства. Стандартизованные коды Событий могут быть отображены на семантически идентичные или ближайшие диагностические определения канала полевой шины в приложении шлюза. Коды IODD, определяемые изготовителем, могут отображаться на диагностические определения конкретного канала (отдельные коды и связанная человекочитаемая информация) в файле описания устройства на полевой шине.

Технические средства полевых шин и системы мониторинга процесса (человеко-машинные интерфейсы) могут применять описание устройства на полевой шине для расшифровки полученных диагностических кодов полевой шины в человекочитаемом диагностическом тексте.

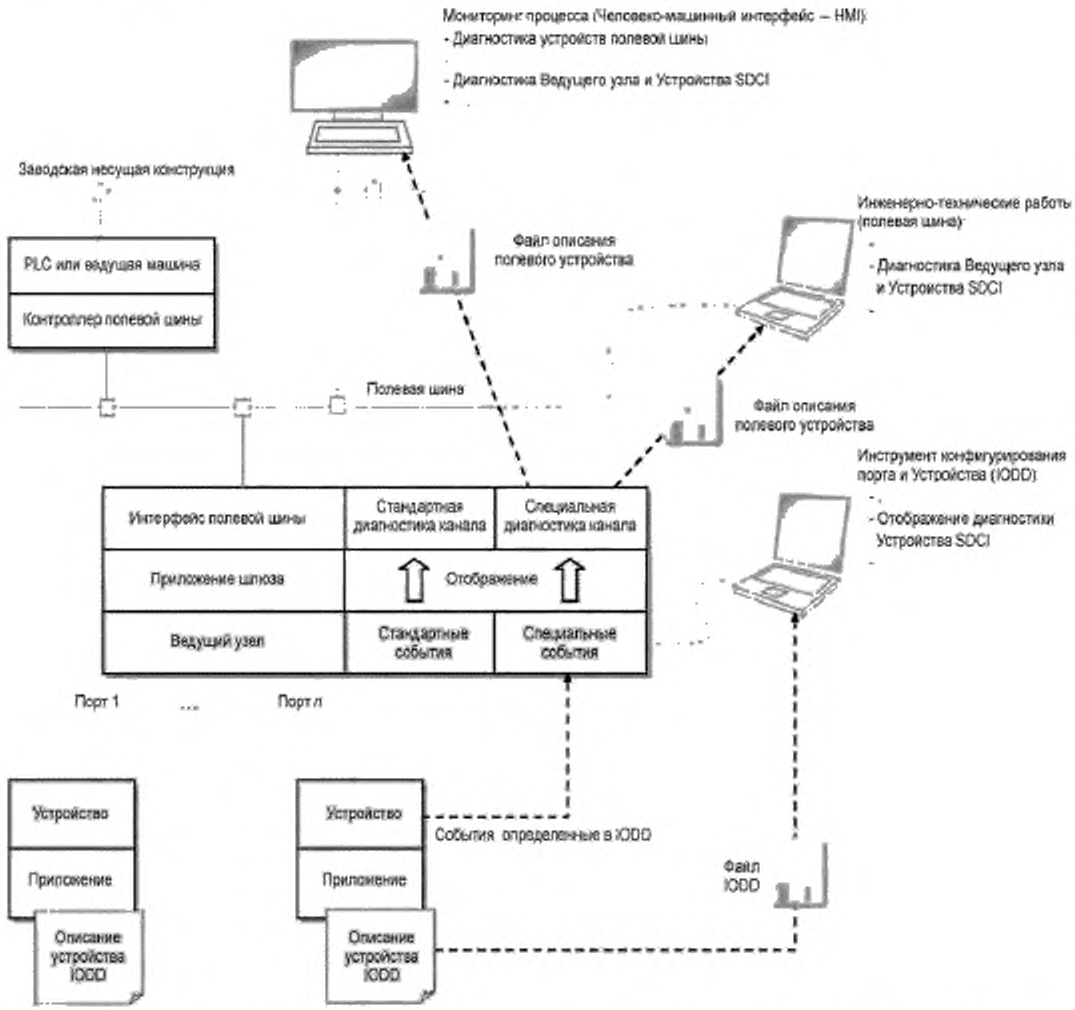
Наплывы диагностической информации предотвращаются контролем потока, который позволяет только одному Событию на Устройство передаваться в приложение Ведущего узла или шлюза в каждый момент времени.

Приложение шлюза способно запустить или остановить Блок диагностики (см. рисунок 95). Находясь в остановленном состоянии, Блок диагностики откладывает все полученные вызовы услуги AL_Event.ind до следующего своего запуска.

Специальное Событие DS_UPLOAD_REQ (см. подраздел 10.2 и таблицу D.2) перенаправляется в общее приложение Хранилища памяти на Ведущем узле. Данные события подтверждаются самим Блоком диагностики и не передаются в шлюз.

На рисунке 106 показан пример потока диагностической информации через полную систему SDCI и полевой шины.

Примечание — Поток может заканчиваться на Ведущем узле и Инструменте конфигурирования порта и Устройства (PDCT) или интегрироваться дальше в зависимости от возможностей полевой шины.



Примечание — Голубые затененные области указывают на характеристики, установленные в настоящем стандарте.

Рисунок 106 — Обзор системы распространения диагностической информации SDCI через События

11.6 Обмен Данными процесса (PDE)

11.6.1 Общие положения

Устройство PDE обеспечивает передачу PD между приложением шлюза и подключенным Устройством.

После установления связи и DS порт готов к любым передачам ODE. Передача PD становится возможной всякий раз, когда отдельный порт или все порты переключены в режим OPERATE.

11.6.2 Отображение PD

В соответствии с 11.2.2.4 входные и выходные PD отображаются на конкретный порт потока PD шлюза.

На рисунке 107 показан пример отображения PD из трех портов Ведущего узла в поток PD шлюза.

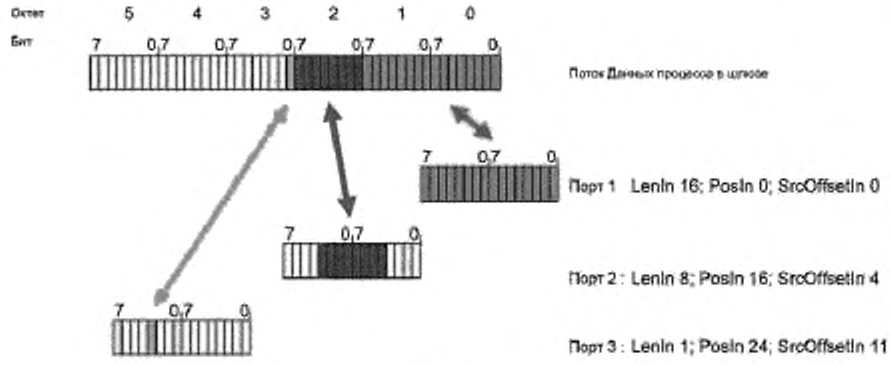


Рисунок 107 — Отображение PD из портов на поток данных шлюза

11.6.3 Допустимое/недопустимое состояние описателя PD

Примерная передача «недопустимого» состояния описателя выходных PD из AL Ведущего узла в AL Устройства показана в верхней части рисунка 108.

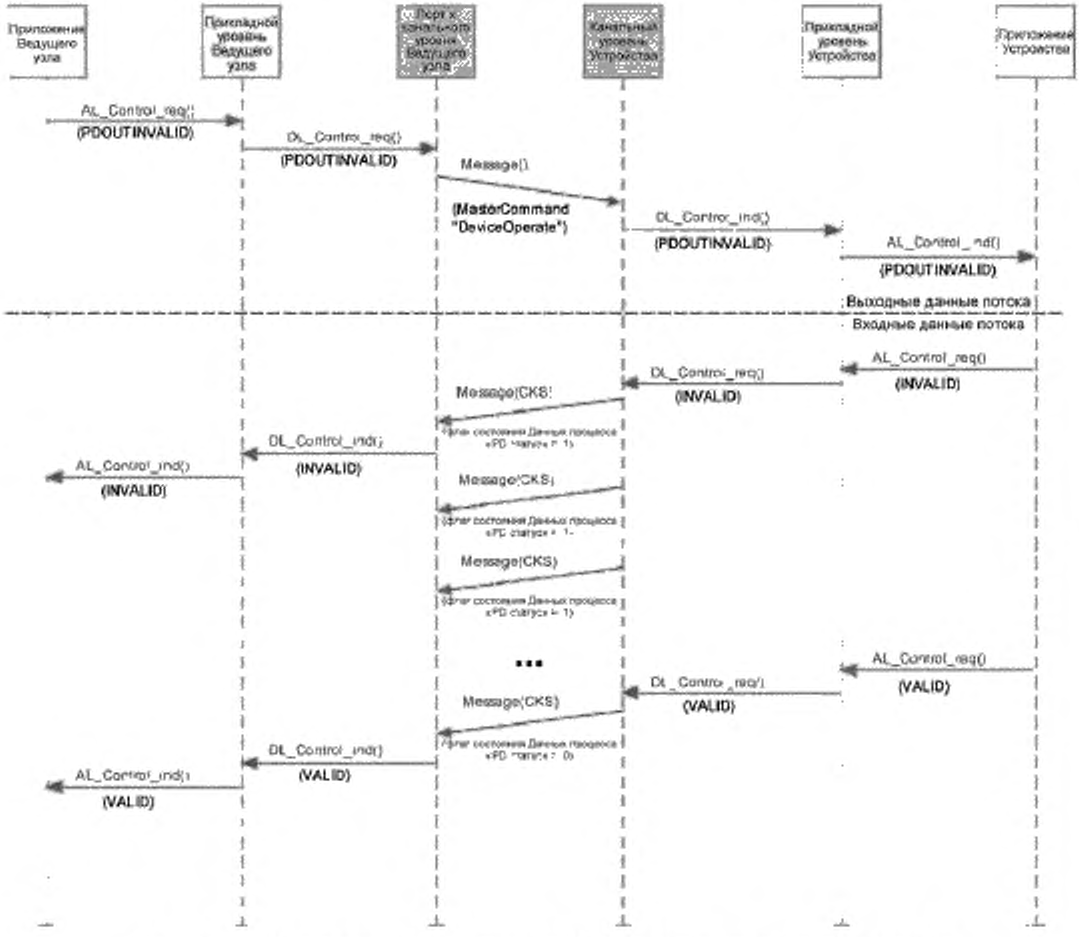


Рисунок 108 — Передача состояния описателя PD между Ведущим узлом и Устройством

Ведущий узел информирует Устройство о «допустимом/недопустимом» состоянии описателя выходных PD, посылая команду Ведущего узла (см. таблицу В.2) на страницу 1 Непосредственных параметров (см. 7.3.7.1).

Для входных PD Устройство посылает состояние описателя PD в очень коротком сообщении, состоящем из флага состояния PD «PD status» в октете Checksum/Status (CKS) (см. А.1.5). Примерная передача «допустимого» состояния описателя входных PD из AL Устройства в AL Ведущего узла показана в нижней части рисунка 108.

Любые отклонения при нахождении в режиме расслоенной передачи ведут к индикации состояния «недопустимый» описателя входных или выходных PD.

11.7 Инструмент конфигурирования порта и Устройства (PDCT)

11.7.1 Общие положения

На рисунках 93 и 106 демонстрируется необходимость инструмента для конфигурирования портов, параметризации Устройства, предоставления диагностической информации и информации об идентификации и технической поддержке. В зависимости от уровня интеграции в систему полевых шин функции PDCT могут быть сокращены, например, если конфигурация порта может быть осуществлена через файл описания полевого устройства конкретной полевой шины.

Функциональность PDCT может интегрироваться частично (навигация, передача параметров и т. п.) или полностью в технический инструмент конкретной полевой шины.

11.7.2 Пример базового расположения

На рисунке 109 показан пример базового расположения дисплея PDCT.

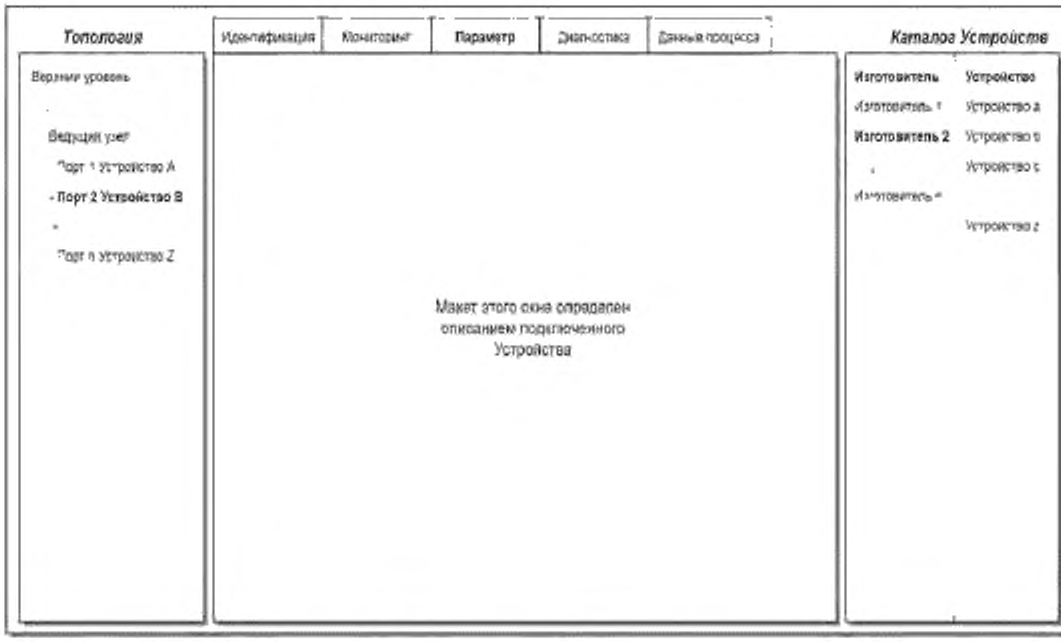


Рисунок 109 — Пример 1 базового расположения дисплея PDCT

Дисплей PDCT всегда должен предоставлять окно навигации для проекта или сетевой топологии, окно для конкретного вида выбранного Устройства, определяемого его описанием устройства IODD, и окно для доступных Устройств, базирующихся на установленных файлах IODD.

На рисунке 110 показан другой пример базового расположения дисплея PDCT.

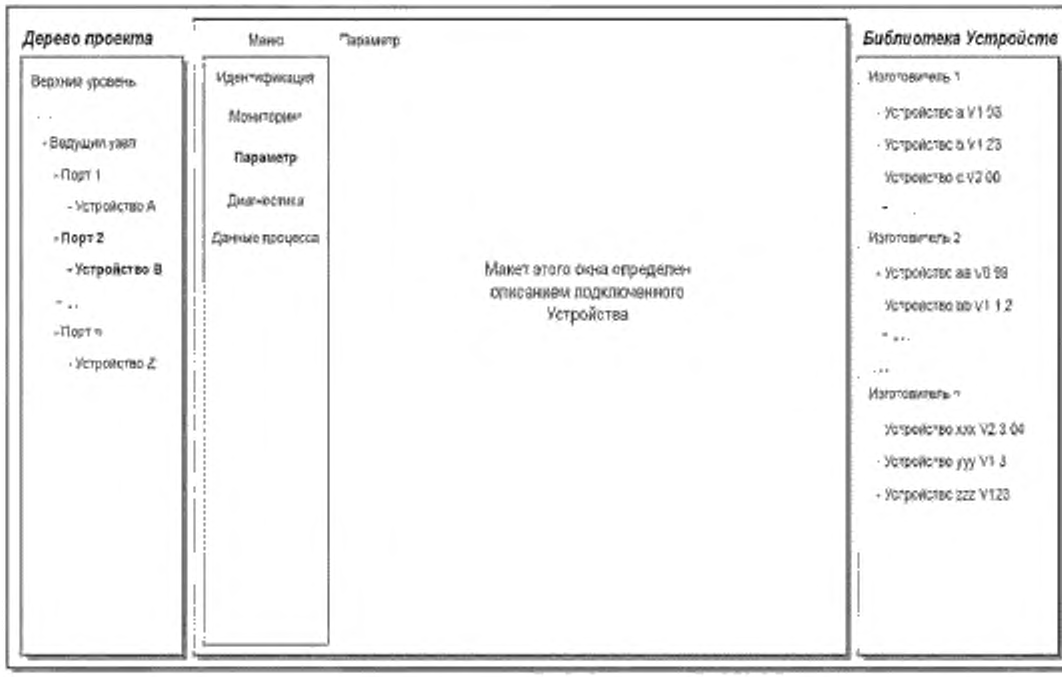


Рисунок 110 — Пример 2 базового расположения дисплея PDCT

Примечание — Дополнительная информация может быть получена из IEC/TR 62453-61.

11.8 Приложение шлюза

11.8.1 Общие положения

Приложение шлюза зависит от конкретной системы ведущего компьютера (полевая шина, PLC и т. д.). Приложения Ведущего узла встраиваются в нее. Конкретная система должна определить отображение услуг и переменных Ведущего узла.

11.8.2 Изменение конфигурации Устройства, включая DS

После каждого изменения конфигурации или параметризации Устройства (CVID и/или CDID, см. 9.2.2.2) связанный ранее сохраненный набор данных на Ведущем узле должен быть очищен или отмечен как недействительный через переменную DS_Delete.

11.8.3 Управление сервером параметров и набором правил

Ведущий узел может комбинировать целые наборы параметров подключенного Устройства с другими важными данными для его собственного функционирования и делать эти данные доступными для приложений более высоких уровней. Например, эти данные могут сохраняться на сервере параметров, к которому можно получать доступ из программы PLC для изменения параметров набора правил, таким образом, поддерживая гибкое производство.

Примечание — Структура данных, передаваемых между Ведущим узлом и сервером параметром, находится вне сферы охвата настоящего стандарта.

11.8.4 Анонимные параметры

Альтернатива к использованию Инструмента конфигурирования порта и Устройства является необходимой для некоторых интерфейсов шлюза. Для данных интерфейсов рекомендуется, чтобы интерфейсы шлюза позволяли ведущему компьютеру посылать блок из 10 безымянных октетов данных конфигурации Устройства для каждого Устройства, подключенного к Ведущему узлу. Интерфейс шлюза будет затем применять услугу AL_Write для доставки октетов для каждого Устройства на страницу 2 Непосредственных параметров соответствующего Устройства.

Примечание — Спецификации интеграции находятся вне сферы охвата настоящего стандарта.

Данный подход показан на рисунке 111.

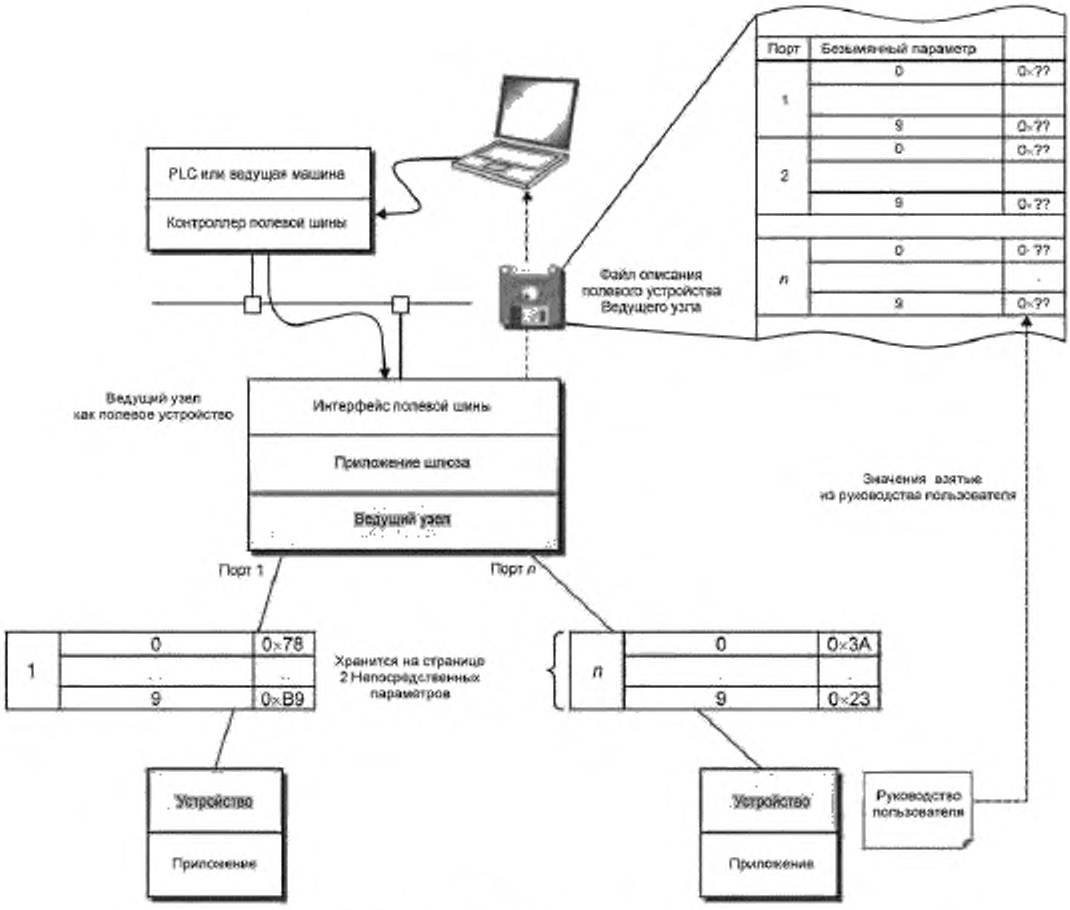


Рисунок 111 — Альтернативная конфигурация Устройства

11.8.5 Режим DIwithSDCI виртуального порта

Данный необязательный рабочий режим предоставляет возможность применять Устройство с возможностью SIO в режиме DIwithSDCI (Цифровой ввод с SDCI) и позволяет системам более высокого уровня (например, PLC) ациклически обмениваться OD. Предпочтительно это будет происходить, когда необходимо изменить параметры, при остановках технологического процесса или в интервалах диагностики.

Данный рабочий режим упрощает управляющую программу из-за невыполнения конфигурации до и после ациклического доступа.

В принципе, приложение шлюза виртуально реализует данный рабочий режим. Оно способно самостоятельно решать в отдельных состояниях, какими могут быть следующие шаги.

СМ не знает об этом рабочем режиме. Приложение шлюза считывает данные конфигурации, сохраненные МС, и применяет услуги SM и услуги AL для реализации данного рабочего режима.

При реализации режима DIwithSDCI необходимо соблюдать следующие правила:

- сигнал DI Устройства не является допустимым во время ациклического доступа приложения шлюза;

- есть вероятность, что сигнал DI обнаруживается слишком поздно. Таким образом, после следующего ациклического доступа может возникнуть Событие PDInvalid, обрыв проводов или будет обнаружена замена Устройства,

- доступ будет занимать больше времени из-за установления связи и процедур возврата в исходное состояние, включая DS;
- уровень проверки должен включать по меньшей мере TYPE_COMP, чтобы обнаруживать недопустимое Устройство.

На диаграмме состояний на рисунке 112 показаны отдельные состояния для виртуального рабочего режима DIwithSDCI.

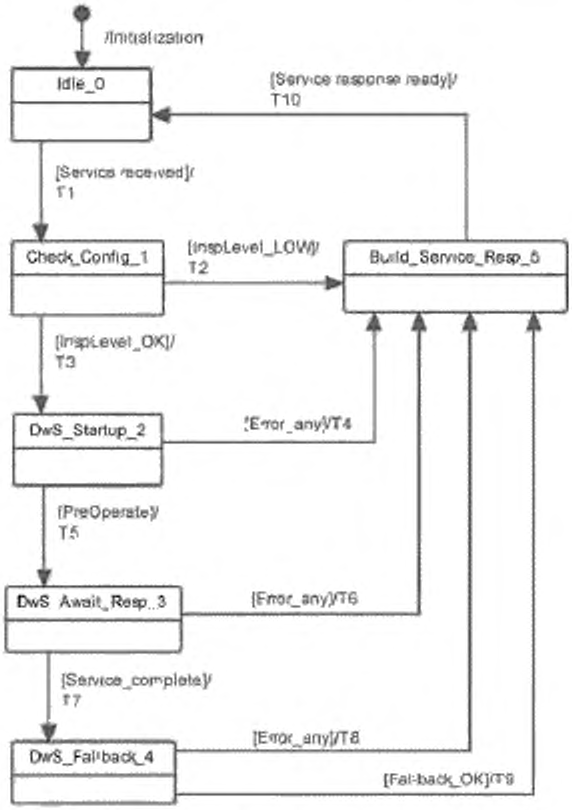


Рисунок 112 — Режим DIwithSDCI виртуального порта

В таблице 104 показаны состояния и переходы режима DIwithSDCI виртуального порта.

Таблица 104 — Переходы состояний конечной машины DIwithSDCI

Имя состояния	Описание состояния
Idle_0	Для управляющей программы более высокого уровня порт конфигурируется для рабочего режима DIwithSDCI и ожидает запросов на услуги
Check_Config_1	В данном состоянии уровень проверки порта проверяется на достаточность уровня
DwS_StartUp_2	В данном состоянии выполняется полный запуск до достижения состояния PREOPERATE. Это может включать DS и обработку Событий
DwS_Await_Resp_3	Ожидание ответов (AL_Read.rsp или AL_Write.rsp)
DwS_FallBack_4	После завершения услуг Устройство переключается обратно в режим SIO через команду Fallback (Возврат в исходное состояние) Ведущего узла и порт будет сконфигурирован как цифровой ввод DI

Окончание таблицы 104

Имя состояния		Описание состояния	
Build_Service_Resp_5		Ответ услуги (положительный или отрицательный) создается для завершения услуги управляющей программе более высокого уровня	
Переход	Исходное состояние	Конечное состояние	Действие
T1	0	1	—
T2	1	5	—
T3	1	2	Вызвать услугу SM_SetPortConfig (в COMx)
T4	2	5	Вызвать услугу SM_SetPortConfig (в DI)
T5	2	3	Вызвать услугу AL_Read.req или AL_Write.req
T6	3	5	Вызвать услугу SM_SetPortConfig (в DI)
T7	3	4	—
T8	4	5	Вызвать услугу SM_SetPortConfig (в DI)
T9	4	5	Вызвать услугу SM_SetPortConfig (в DI)
T10	5	0	Вызвать ответ на соответствующую услугу AL
Внутренние элементы		Тип	Определение
InspLevel_LOW		Bool	Не сконфигурирован необходимый уровень проверки для обнаружения правильного Устройства
InspLevel_OK		Bool	Сконфигурирован необходимый уровень проверки для обнаружения правильного Устройства
PreOperate		Bool	Установлено состояние PREOPERATE
Service_complete		Bool	Приложение шлюза получило ответ AL_Read.rsp или AL_Write.rsp
Fallback_OK		Bool	Возврат в исходное состояние завершился успешно
Error_any		Bool	Любая ошибка приведет к выходу из данного состояния

Приложение А
(обязательное)

Кодирование, временные ограничения и ошибки

А.1 Общая структура и кодирование М-последовательностей

А.1.1 Обзор

Общее понятие М-последовательностей изложено в общих чертах в 7.3.3.2. В А.1.2—А.1.6 дано подробное описание отдельных элементов М-последовательностей.

А.1.2 Управление М-последовательностями (МС)

Ведущий узел указывает способ, посредством которого данные пользователя (см. А.1.4) будут передаваться в октет управления М-последовательностью. Данное указание включает направление передачи (чтение или запись), канал связи и адрес (смещение) данных в канале связи. Структура октета управления М-последовательностью показана на рисунке А.1.



Рисунок А.1 — Управление М-последовательностью

Биты от 0 до 4: Адрес

Данные биты указывают адрес, т. е. смещение октета в данных пользователя в указанном канале связи (см. также таблицу А.1). В случае канала ISDU данные биты применяются для управления потоком данных ISDU. Адрес, который в данном случае означает позицию данных пользователя в ISDU, доступен только неявно (см. 7.3.6.2).

Биты от 5 до 6: Канал связи

Данные биты указывают канал связи для доступа к данным пользователя. Определенные значения для параметра канала связи приведены в таблице А.1.

Таблица А.1 — Значения канала связи

Значение	Определение
0	Процесс
1	Страница
2	Диагностика
3	ISDU

Бит 7 Ч/З

Данный бит указывает направление передачи данных пользователя в выбранном канале связи, т. е. доступ на запись (передача данных пользователя из Устройства в Ведущий узел) или доступ на запись (передача данных пользователя из Ведущего узла в Устройство). Определенные значения параметра Ч/З приведены в таблице А.2.

Таблица А.2 — Значения Ч/З

Значение	Определение
0	Доступ на запись
1	Доступ на чтение

Устройство не обязано поддерживать каждое из 256 значений октета управления М-последовательностью. Для доступа на чтение нереализованных адресов или каналов связи будет возвращаться значение «0». Доступ на запись к нереализованным адресам или каналам связи будет игнорироваться.

А.1.3 Контрольная сумма/тип М-последовательности (СКТ)

Тип М-последовательности передается вместе с контрольной суммой в октете проверки/типа. Структура данного октета показана на рисунке А.2.

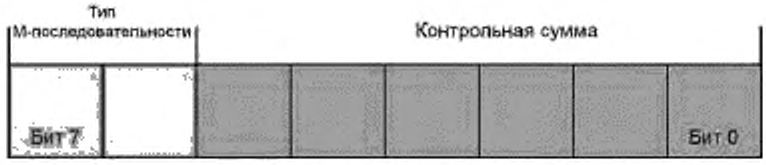


Рисунок А.2 — Октет контрольной суммы/типа М-последовательности

Биты от 0 до 5: Контрольная сумма

Данные биты содержат 6-битовую контрольную сумму сообщения для обеспечения целостности данных, см. также А.1.6 и Н.1.

Биты от 6 до 7: Тип М-последовательности

Данные биты указывают тип М-последовательности. При этом Ведущий узел указывает, как структурированы сообщения в М-последовательности. Определенные значения для параметра типа М-последовательности приведены в таблице А.3.

Таблица А.3 — Значения типов М-последовательности

Значение	Определение
0	Тип 0
1	Тип 1
2	Тип 2 (см. примечание)
3	зарезервировано
Примечание — Подтипы зависят от конфигурации PD и направления передачи PD.	

А.1.4 Данные пользователя (PD или OD)

Данные пользователя — это общий термин как для PD, так и OD. Длина данных пользователя может изменяться от 0 до 64 октетов в зависимости от типа М-последовательности и направления передачи (чтение/запись). Обзор доступных типов данных показан в таблице А.4. Эти типы данных могут быть организованы как записи (различные типы) или массивы (одинаковый тип).

Таблица А.4 — Типы данных для данных пользователя

Тип данных	Ссылка
BooleanT	См. Е.2.2
UIntegerT	См. Е.2.3
IntegerT	См. Е.2.4
StringT	См. Е.2.6
OctetStringT	См. Е.2.7
Float32T	См. Е.2.5
TimeT	См. Е.2.8
TimeSpanT	См. Е.2.9

Подробное кодирование типов данных описывается в приложении Е.

А.1.5 Контрольная сумма/состояние (CKS)

Октет контрольной суммы/состояния является частью ответного сообщения из Устройства в Ведущий узел. Его структура показана на рисунке А.3. Октет включает 6-битовую контрольную сумму, флаг допустимости или недопустимости PD и флаг События.



Рисунок А.3 — Октет контрольной суммы/состояния

Биты от 0 до 5: Контрольная сумма

Данные биты содержат 6-битовую контрольную сумму для обеспечения целостности данных ответного сообщения. См. также А.1.6 и Н.1.

Бит 6: Состояние PD

Данный бит указывает, может Устройство обеспечить допустимые PD или нет. Определенные значения параметра приведены в таблице А.5.

Флаг состояния PD применяется для Устройств с входными PD. Устройства с выходными PD всегда указывают «Process Data valid» (PD допустимы).

Если флаг состояния PD установлен в «Process Data invalid» (PD недопустимы) в сообщении, все входные PD полного цикла PD являются недопустимыми.

Таблица А.5 — Значения состояния PD

Значение	Определение
0	PD допустимы
1	PD недопустимы

Бит 7: Флаг События

Данный бит указывает инициативу Устройства, что данные категории «Событие» будут извлекаться Ведущим узлом через диагностический канал связи (см. таблицу А.1). Устройство может сообщать диагностическую информацию, такую как ошибки, предупреждения или уведомления, через ответные сообщения События. Допустимые значения параметра приведены в таблице А.6.

Таблица А.6 — Значения флага События

Значение	Определение
0	Нет События
1	Событие

А.1.6 Вычисление контрольной суммы

Контрольная сумма события предоставляет защиту целостности данных при передаче данных из Ведущего узла в Устройство и из Устройства в Ведущий узел. Каждый октет данных UART защищен битом четности UART (см. рисунок 18). Кроме этой защиты октета отдельных данных, все октеты данных UART в сообщении обрабатываются операцией XOR (исключающее ИЛИ) октет за октетом. Октет контроля/типа включен с битами контрольной суммы, установленными в «0». Результирующий октет контрольной суммы сжимается с 8 до 6 битов в соответствии с процедурой преобразования, показанной на рисунке А.4, и ее соответствующими формулами [см. уравнение (А.1)]. 6-битовая сжатая «Контрольная сумма вводится в октет контрольной суммы/типа М-последовательности (см. рисунок А.2). Также процедура применяется для защиты сообщения из Устройства в Ведущий узел. В данном случае сжатая контрольная сумма вводится в октет контрольной суммы/состояния (см. рисунок А.3).

Начальное значение 0x52 применяется при вычислении контрольной суммы во всем сообщении. Оно подвергается операции XOR с первым октетом сообщения (FC).

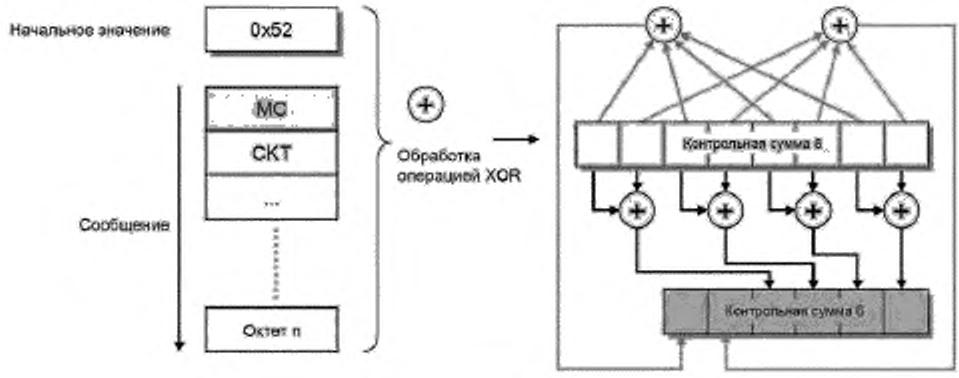


Рисунок А.4 — Принцип вычисления контрольной суммы и сжатия

Набор уравнений в формулах (А.1) детально определяет процедуру сжатия с 8 до 6 битов.

$D5_8 = D7_8$ xor $D5_8$ xor $D3_8$ xor $D1_8$
$D4_6 = D6_8$ xor $D4_8$ xor $D2_8$ xor $D0_8$
$D3_6 = D7_8$ xor $D6_8$ (А.1)
$D2_6 = D5_8$ xor $D4_8$
$D1_6 = D3_8$ xor $D2_8$
$D0_6 = D1_8$ xor $D0_8$

А.2 Типы М-последовательностей

А.2.1 Обзор

PD и OD применяют отдельные циклические и ациклические каналы связи (см. рисунок 7) для обеспечения планируемой и детерминированной доставки PD, тогда как доставка OD не влияет на характеристики передачи PD.

В интерфейсе SDCI М-последовательности обеспечивают доступ к каналам связи через октет Управления М-последовательностью. Ряд различных типов М-последовательностей удовлетворяет различным требованиям датчиков и исполнительных устройств относительно ширины их PD. На рисунке 37 приведен обзор доступных типов М-последовательностей, которые устанавливаются в А.2.2—А.2.5. В А.2.6 приводятся правила использования типов М-последовательностей.

А.2.2 М-последовательность типа TYPE_0

М-последовательность типа TYPE_0 является обязательной для всех Устройств.

М-последовательность типа TYPE_0 передает только OD. За цикл считывается или записывается один октет данных пользователя. Данная М-последовательность показана на рисунке А.5.

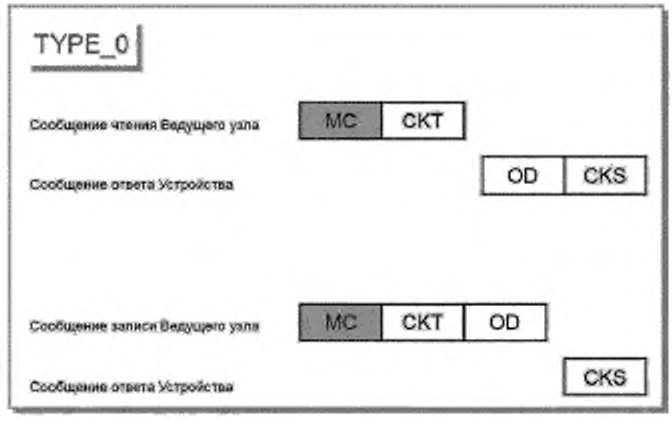


Рисунок А.5 — М-последовательность типа TYPE_0

А.2.3 М-последовательность типа TYPE_1_x

М-последовательность типа TYPE_1_x является необязательной для всех Устройств.

М-последовательность типа TYPE_1_1 показана на рисунке А.6.

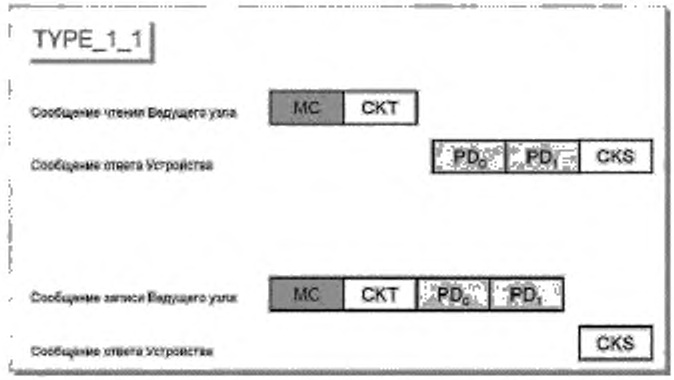


Рисунок А.6 — М-последовательность типа TYPE_1_1

За цикл считываются или записываются два октета PD. Адрес (битовое смещение) принадлежит каналу связи процесса (см. А.2.1).

В случае режима расслоения (см. 7.3.4.2) и нечетной длины PD остающиеся октеты заполняются значением 0x00.

М-последовательность типа TYPE_1_2 показана на рисунке А.7. За цикл считываются или записываются два октета OD.

При доступе на запись к OD через каналы связи страниц и диагностики вычисляется только первый октет OD. Устройство игнорирует оставшиеся октеты. Ведущий узел посылает остальные OD со значением «0x00».

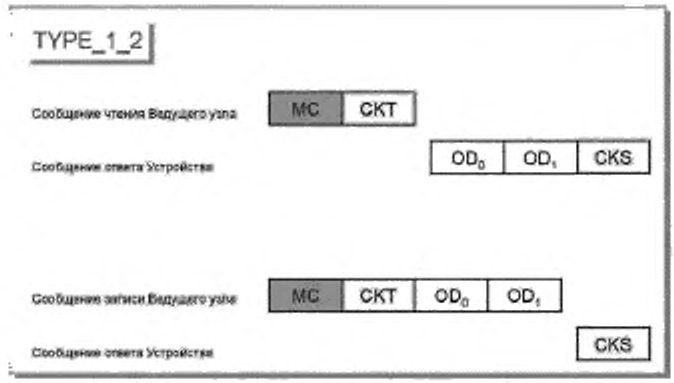


Рисунок А.7 — М-последовательность типа TYPE_1_2

М-последовательность типа TYPE_1_V, обеспечивающая переменную (расширяемую) длину сообщения, показана на рисунке А.8. За цикл считывается или записывается m октетов OD.

При доступе на запись к OD через каналы связи страниц и диагностики вычисляется только первый октет (OD0) OD. Устройство игнорирует оставшиеся октеты. Ведущий узел посылает остальные OD со значением «0x00».

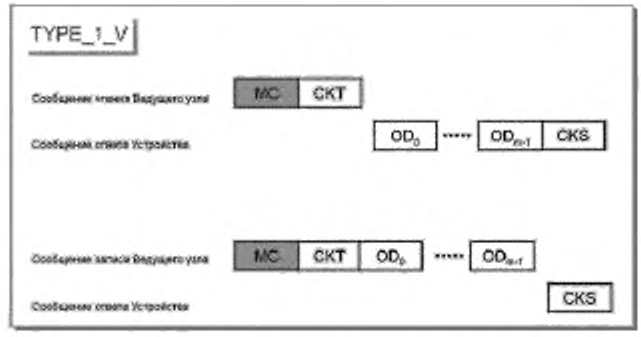


Рисунок А.8 — М-последовательность типа TYPE_1_V

А.2.4 М-последовательность типа TYPE_2_x

М-последовательность типа TYPE_2_x является необязательной для всех Устройств. Определяются М-последовательности типов от TYPE_2_1 по TYPE_2_6. М-последовательность типа TYPE_2_V обеспечивает переменную (расширяемую) длину сообщения. М-последовательность типа TYPE_2_x передает PD и OD в одном сообщении. Количество считываемых или записываемых PD и OD в каждом цикле зависит от типа. Параметр Адрес (см. рисунок А.1) принадлежит в данном случае каналу связи OD. Адрес PD определяется неявно, начиная с «0». Формат PD характеризует М-последовательности типа TYPE_2_x.

М-последовательность типа TYPE_2_1 передает один октет считываемых PD и один октет считываемых или записываемых OD за цикл. Данный тип М-последовательности показан на рисунке А.9.

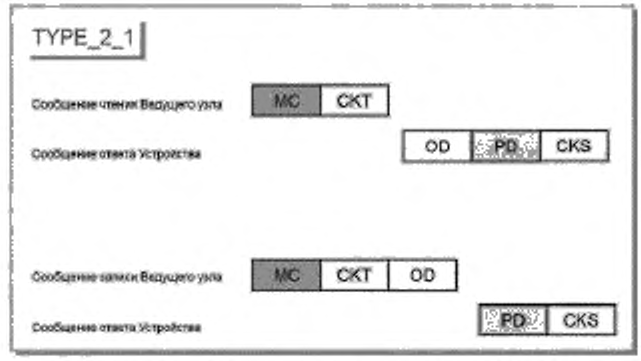


Рисунок А.9 — М-последовательность типа TYPE_2_1

М-последовательность типа TYPE_2_2 передает два октета считываемых PD и один октет OD за цикл. Данная М-последовательность показана на рисунке А.10.

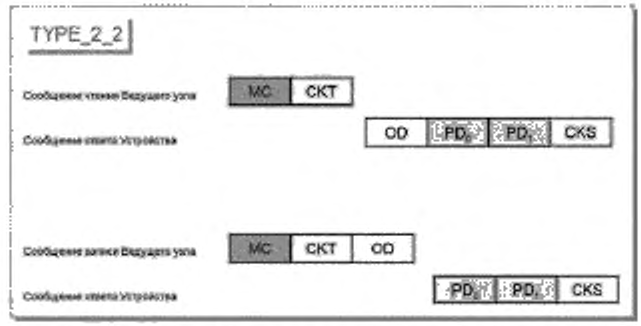


Рисунок А.10 — М-последовательность типа TYPE_2_2

М-последовательность типа TYPE_2_3 передает один октет записываемых PD и один октет считываемых или записываемых OD за цикл. Данная М-последовательность показана на рисунке А.11.

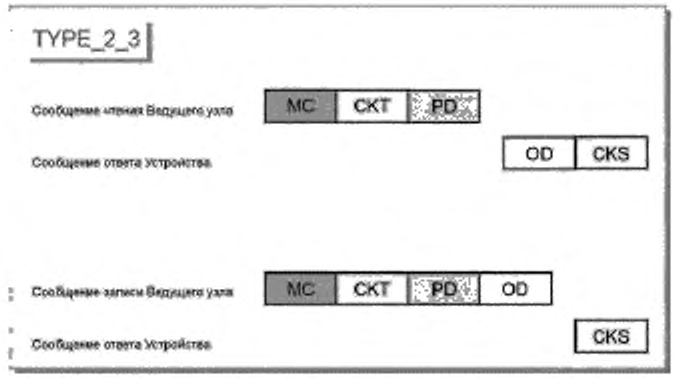


Рисунок А.11 — М-последовательность типа TYPE_2_3

М-последовательность типа TYPE_2_4 передает два октета записываемых PD и один октет считываемых или записываемых OD за цикл. Данный тип М-последовательности показан на рисунке А.12.

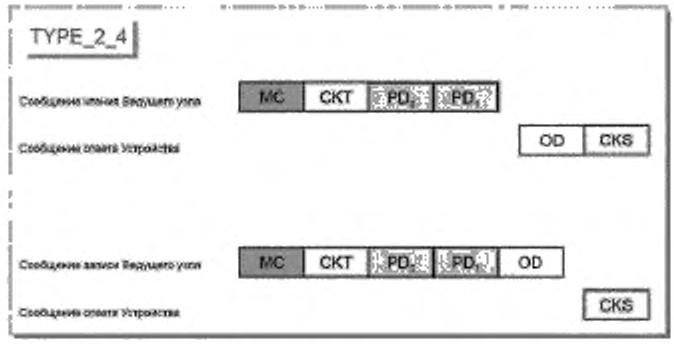


Рисунок А.12 — М-последовательность типа TYPE_2_4

М-последовательность типа TYPE_2_5 передает один октет записываемых и считываемых PD и один октет считываемых или записываемых OD за цикл. Данная М-последовательность показана на рисунке А.13.

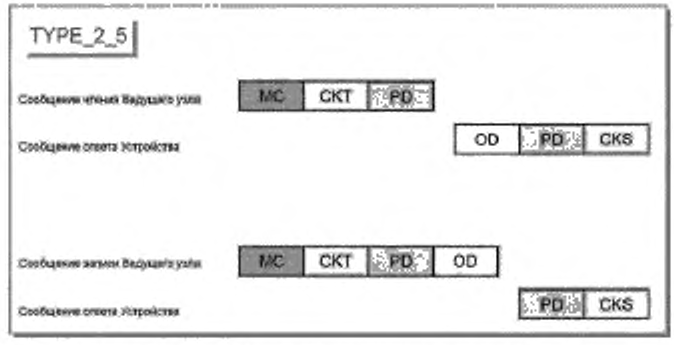


Рисунок А.13 — М-последовательность типа TYPE_2_5

М-последовательность типа TYPE_2_6 передает два октета записываемых и считываемых PD и один октет считываемых или записываемых OD за цикл. Данная М-последовательность показана на рисунке А.14.

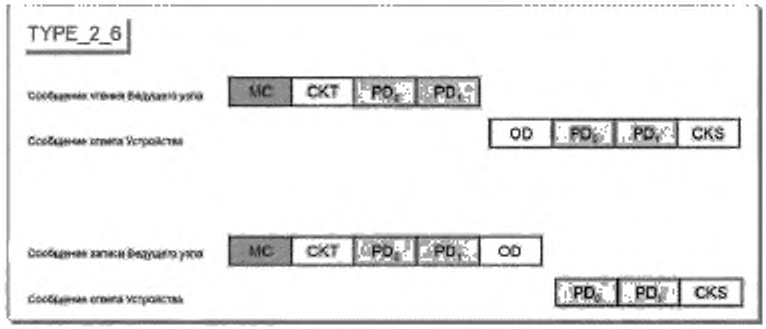


Рисунок А.14 — М-последовательность типа TYPE_2_6

М-последовательность типа TYPE_2_V передает целый набор записываемых (считываемых) входных PD в n (k) октетов за цикл. Диапазон n (k) изменяется от 0 до 32. Либо входные, либо выходные PD отсутствуют, когда n = 0 или k = 0. М-последовательность типа TYPE_2_V также передает m октетов (сегментированных) считываемых или записываемых OD за цикл, применяя адрес на рисунке А.1. Разрешенные значения m — это 1, 2, 8 и 32. Данный переменный тип М-последовательности показан на рисунке А.15.

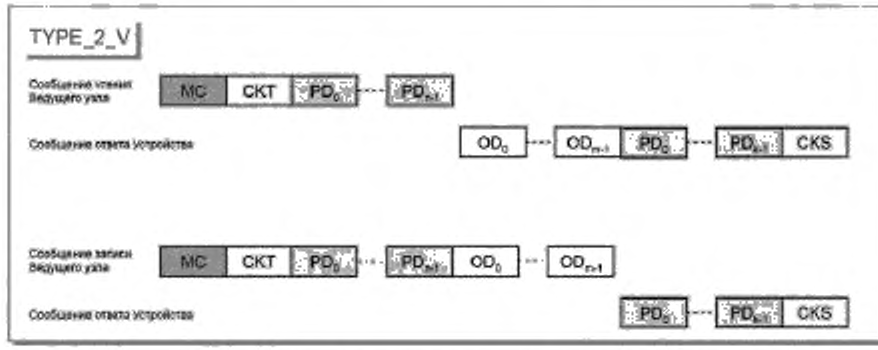


Рисунок А.15 — М-последовательность типа TYPE_2_V

При доступе на запись к OD через каналы связи страниц и диагностики вычисляется только первый октет (OD_0) OD. Устройство игнорирует оставшиеся октеты. Ведущий узел посылает остальные OD со значением «0».

А.2.5 М-последовательность типа 3

М-последовательность типа 3 зарезервирована и не будет применяться.

А.2.6 Использование типа М-последовательности для режимов STARTUP, PREOPERATE и OPERATE

В таблице А.7 приведены типы М-последовательности для режима STARTUP вместе с минимальным временем восстановления ($T_{initcyc}$), которое будет наблюдаться для реализаций Ведущего узла (см. А.3.9). Код М-последовательности ссылается на кодирование в В.1.4.

Таблица А.7 — Типы М-последовательностей для режима STARTUP

Код М-последовательности для режима STARTUP	Данные запроса	Тип М-последовательности	Минимальное время восстановления
	Октеты		T_{BIT}
н/д	1	TYPE_0	100

В таблице А.8 приведены типы М-последовательности для режима PREOPERATE вместе с минимальным временем восстановления ($T_{initcyc}$), которое будет наблюдаться для реализаций Ведущего узла.

Таблица А.8 — Типы М-последовательности для режима PREOPERATE

Код М-последовательности для режима PREOPERATE	Данные запроса	Тип М-последовательности	Минимальное время восстановления
	Октеты		T_{BIT}
0	1	TYPE_0	100
1	2	TYPE_1_2	100
2	8	TYPE_1_V	210
3	32	TYPE_1_V	550
Примечание — Минимальное время восстановления в режиме PREOPERATE является требованием для Ведущего узла.			

В таблице А.9 приведены типы М-последовательностей для режима OPERATE для устаревших Устройств. Минимальное время цикла для Ведущего узла в режиме OPERATE устанавливается в параметре MinCycleTime Устройства (см. В.1.3).

Таблица А.9 — Типы М-последовательности для режима OPERATE (устаревший протокол)

Код М-последовательности для режима OPERATE	Данные запроса	Данные процесса		Тип М-последовательности
	Октеты	Входные Данные процесса	Выходные Данные процесса	Устаревший протокол (см. [8])
0	1	0	0	TYPE_0
1	2	0	0	TYPE_1_2
Безразлично	2	От 3 до 32 октетов	От 0 до 32 октетов	TYPE_1_1/1_2 (расслоенный)
Безразлично	2	От 0 до 32 октетов	От 3 до 32 октетов	TYPE_1_1/1_2 (расслоенный)
Безразлично	1	От 1 до 8 битов	0	TYPE_2_1
Безразлично	1	От 9 до 16 битов	0	TYPE_2_2
Безразлично	1	0	От 1 до 8 битов	TYPE_2_3
Безразлично	1	0	От 9 до 16 битов	TYPE_2_4
Безразлично	1	От 1 до 8 битов	От 1 до 8 битов	TYPE_2_5

В таблице А.10 приведены типы М-последовательностей для режима OPERATE для Устройств в соответствии с настоящим стандартом. Минимальное время цикла для Ведущего узла в режиме OPERATE устанавливается в параметре MinCycleTime Устройства (см. В.1.3).

Таблица А.10 — Типы М-последовательности для режима OPERATE

Код М-последовательности для режима OPERATE	Данные запроса	Данные процесса		Тип М-последовательности
	Октеты	Входные Данные процесса	Выходные Данные процесса	
0	1	0	0	TYPE_0
1	2	0	0	TYPE_1_2
6	8	0	0	TYPE_1_V
7	32	0	0	TYPE_1_V
0	2	От 3 до 32 октетов	От 0 до 32 октетов	TYPE_1_1/1_2 (расслоенный)

Окончание таблицы А.10

Код М-последовательности для режима OPERATE	Данные запроса	Данные процесса		Тип М-последовательности
	Октеты	Входные Данные процесса	Выходные Данные процесса	
0	2	От 0 до 32 октетов	От 3 до 32 октетов	TYPE_1_1/1_2 (расслоенный)
0	1	От 1 до 8 битов	0	TYPE_2_1
0	1	От 9 до 16 битов	0	TYPE_2_2
0	1	0	От 1 до 8 битов	TYPE_2_3
0	1	0	От 9 до 16 битов	TYPE_2_4
0	1	От 1 до 8 битов	От 1 до 8 битов	TYPE_2_5
0	1	От 9 до 16 битов	От 1 до 16 битов	TYPE_2_6
0	1	От 1 до 16 битов	От 9 до 16 битов	TYPE_2_6
4	1	От 0 до 32 октетов	От 3 до 32 октетов	TYPE_2_V
4	1	От 3 до 32 октетов	От 0 до 32 октетов	TYPE_2_V
5	2	> 0 Бит, октеты	≥ 0 Бит, октеты	TYPE_2_V
5	2	≥ 0 Бит, октеты	> 0 Бит, октеты	TYPE_2_V
6	8	> 0 Бит, октеты	≥ 0 Бит, октеты	TYPE_2_V
6	8	≥ 0 Бит, октеты	> 0 Бит, октеты	TYPE_2_V
7	32	> 0 бит, октеты	≥ 0 бит, октеты	TYPE_2_V
7	32	≥ 0 бит, октеты	> 0 бит, октеты	TYPE_2_V

А.3 Временные ограничения

А.3.1 Общие положения

Взаимодействия Ведущего узла и его Устройств характеризуются несколькими ограничениями времени, которые применяются к фреймам UART, временам передачи сообщения Ведущим узлом и Устройством, дополненным временами ответа, цикла, задержки и восстановления.

А.3.2 Время бита

Время бита T_{BIT} — это время, требуемое для передачи единственного бита. Время бита является обратной величиной скорости передачи [см. формулу (А.2)]:

$$T_{BIT} = 1/(\text{скорость передачи}) \quad (A.2)$$

Значения T_{BIT} установлены в таблице 8.

А.3.3 Задержки передачи фрейма UART Ведущего узла (порты)

Задержка передачи фрейма UART t_1 порта — это промежуток времени между концом стопового бита фрейма UART и началом стартового бита следующего фрейма UART. Порт передает фреймы UART с максимальной задержкой в одно время бита [см. формулу (А.3)]:

$$0 \le t_1 \le 1\ T_{BIT} \quad (A.3)$$

А.3.4 Задержка передачи фрейма UART Устройства

Задержка передачи фрейма UART t_2 — это промежуток времени между концом стопового бита фрейма UART и началом стартового бита следующего фрейма UART. Устройство передает фреймы UART с максимальной задержкой в три времени бита [см. формулу (А.4)]:

$$0 \le t_2 \le 3\ T_{BIT} \quad (A.4)$$

А.3.5 Время реакции Устройства

Время реакции Устройства t_A — это промежуток времени между концом стопового бита последнего полученного фрейма UART порта и началом стартового бита первого посылаемого фрейма UART. Устройство наблюдает задержку по меньшей мере одного времени бита, но не более 10 времен бита [см. формулу (А.5)]:

$$1\ T_{BIT} \le t_A \le 10\ T_{BIT} \quad (A.5)$$

А.3.6 Время М-последовательности

Связь между портом и его связанными Устройствами происходит в твердо установленном графике, называемом временем М-последовательности [см. формулу (А.6)]:

$$t_{M\text{-sequence}} = (m + n)\cdot 11\cdot T_{BIT} + t_A + (m-1)\cdot t_1 + (n-1)\cdot t_2 \quad (A.6)$$

В этой формуле m — число фреймов UART, посланных портом в Устройство, и n — число фреймов UART, посланных Устройством в порт. Формула может применяться только для оценки, так как времена t_1 и t_2 могут не быть постоянными.

На рисунке А.16 показаны временные соотношения М-последовательности, состоящей из сообщения Ведущего узла (порта) и сообщения Устройства.

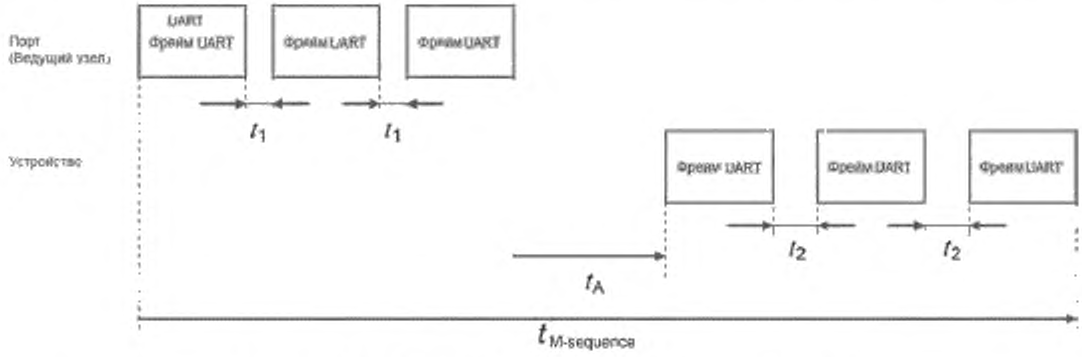


Рисунок А.16 — Временные соотношения М-последовательности

А.3.7 Время цикла

Время цикла t_{CYC} [см. формулу (А.7)] зависит от параметра MinCycleTime Устройства, конструкции и реализации Ведущего узла и числа портов

$$t_{CYC} = t_{M\text{-sequence}} + t_{idle} \quad (A.7)$$

Регулируемый параметр MasterCycleTime Устройства может применяться для проектирования специальной технологии Устройства, такого как исполнительное устройство, чтобы получить временные условия для подходящего действия по умолчанию, такого как деактивация или выключение исполнительного устройства (см. 7.3.3.5 MaxCycleTime, 10.2 и 10.7.3).

В таблице А.11 перечисляются значения рекомендуемого минимального времени цикла для указанных режимов передачи порта. Значения вычислены на базе М-последовательности типа Type_2_1.

Таблица А.11 — Рекомендуемые минимальные времена цикла

Режим передачи	t_{CYC}
COM1	18,0 мс
COM2	2,3 мс
COM3	0,4 мс

А.3.8 Время простоя

Время простоя t_{idle} получается из сконфигурированного времени цикла t_{CYC} и времени М-последовательности $t_{M\text{-sequence}}$. Что касается порта, он включает время между концом сообщения Устройства и началом следующего сообщения из Ведущего узла (порта).

Время простоя должно быть достаточным, чтобы Устройство подготовилось к получению следующего сообщения.

А.3.9 Время восстановления

Ведущий узел ожидает в течение времени восстановления $t_{initcyc}$ между двумя последовательными ациклическими доступами Устройства в режимах STARTUP или PREOPERATE (см. А.2.6).

А.4 Ошибки и способы их исправления

А.4.1 Ошибки UART

А.4.1.1 Ошибки четности

Бит четности UART (см. рисунок 18) и контрольная сумма (см. А.1.6) являются двумя независимыми механизмами, защищающими передачу данных. Это означает, что, например, две битовые ошибки в различных октетах сообщения, приводящие к правильной контрольной сумме, также могут быть обнаружены. Оба механизма приводят к одинаковой обработке ошибки.

Способ исправления: Ведущий узел повторяет свое сообщение два раза (см. 7.2.2.1). Устройства отвергают все данные с обнаруженными ошибками и не реагируют на них.

А.4.1.2 Ошибки фреймов UART

Условия для правильного обнаружения фрейма UART установлены в 5.3.3.2. Обработка ошибки происходит всякий раз, когда нарушенные формы сигнала или неправильные временные соотношения приводят к недопустимому стоповому биту UART.

Способ исправления: см. А.4.1.1.

А.4.2 Ошибки пробуждения

Импульс тока пробуждения установлен в 5.3.3.3, а процедуры пробуждения — в 7.3.2.1. Во время попыток установления связи может случиться несколько ошибок.

Способ исправления: возможны повторные попытки. Подробное описание приводится в 7.3.2.1.

А.4.3 Ошибки передачи

А.4.3.1 Ошибки контрольной суммы

Механизм контрольной суммы установлен в А.1.6. Любые ошибки контрольной суммы ведут к обработке ошибки.

Способ исправления: см. А.4.1.1.

А.4.3.2 Ошибки превышения лимита времени

Различные временные ограничения для М-последовательностей установлены в разделе А.3. Ведущий узел (порт) и устройства проверяют некоторые критические временные соотношения, такие как отсутствие синхронности в сообщениях.

Способ исправления: см. А.4.1.1.

А.4.3.3 Конфликты

Конфликт происходит всегда, когда Ведущий узел и Устройство производят одновременную посылку вследствие ошибки. Такая ошибка интерпретируется как ошибочная М-последовательность.

Способ исправления: см. А.4.1.1.

А.4.4 Ошибки протокола

Ошибки протокола происходят, например, всегда, когда последовательность сегментированной передачи ISDU является ошибочной (см. случай управления потоком в А.1.2).

Способ исправления: аварийное завершения услуги с информацией о типе ошибки (см. приложение С).

А.5 Общая структура и кодирование индексированного сервисного блока данных

А.5.1 Обзор

Назначение и общая структура индексированного сервисного блока данных (ISDU) установлена в 7.3.6.1. В А.5.2—А.5.7 дано подробное описание отдельных элементов ISDU и приведены некоторые примеры.

А.5.2 I-Service

На рисунке А.17 показана структура октета I-Service.



Рисунок А.17 — Октет I-Service

Биты от 0 до 3: Длина

Кодирование полубайта Длины ISDU установлено в таблице А.14.

Биты от 4 до 7: I-Service

Кодирование полубайта I-Service ISDU установлено в таблице А.12.

Все другие элементы структуры, установленные в 7.3.6.1, передаются как независимые октеты.

Таблица А.12 — Определение полубайта I-Service

I-Service (двоичная)	Определение		Формат Индекса
	Ведущий узел	Устройство	
0000	Нет услуги	Нет услуги	н/д
0001	Запрос записи	Зарезервировано	8-битовый Индекс
0010	Запрос записи	Зарезервировано	8-битовый Индекс и Субиндекс
0011	Запрос записи	Зарезервировано	16-битовый Индекс и Субиндекс
0100	Зарезервировано	Ответ записи (–)	<отсутствует>
0101	Зарезервировано	Ответ записи (–)	<отсутствует>
0110	Зарезервировано	Зарезервировано	
0111	Зарезервировано	Зарезервировано	
1000	Зарезервировано	Зарезервировано	
1001	Запрос чтения	Зарезервировано	8-битовый Индекс
1010	Запрос чтения	Зарезервировано	8-битовый Индекс и Субиндекс
1011	Запрос чтения	Зарезервировано	16-битовый Индекс и Субиндекс
1100	Зарезервировано	Ответ записи (–)	<отсутствует>
1101	Зарезервировано	Ответ чтения (+)	<отсутствует>
1110	Зарезервировано	Зарезервировано	
1111	Зарезервировано	Зарезервировано	

В таблице А.13 определяется синтаксис ISDU. Типы ошибок описываются в приложении С.

Таблица А.13 — Синтаксис ISDU

Имя ISDU	Структура ISDU
Запрос записи	{I-Service(0x1), LEN, Index, [Data*], CHKPDU} ^ {I-Service(0x2), LEN, Index, Subindex, [Data*], CHKPDU} ^ {I-Service(0x3), LEN, Index, Index, Subindex, [Data*], CHKPDU}
Ответ записи (+)	I-Service(0x5), Length(0x2), CHKPDU
Ответ записи (–)	I-Service(0x4), Length(0x4), ErrorType, CHKPDU
Запрос чтения	{I-Service(0x9), Length(0x3), Index, CHKPDU} ^ {I-Service(0xA), Length(0x4), Index, Subindex, CHKPDU} ^ {I-Service(0xB), Length(0x5), Index, Index, Subindex, CHKPDU}
Ответ чтения (+)	I-Service(0xD), LEN, [Data*], CHKPDU
Ответ чтения (–)	I-Service(0xC), Length(0x4), ErrorType, CHKPDU
Обозначение: LEN = {Length(0x1), ExtLength} ^ {Length}	

А.5.3 Расширенная длина ExtLength

Число октетов, переданных в этой I-Service, включая всю информацию протокола (шесть октетов), установлено в элементе Length ISDU. Если общая длина превышает 15 октетов, длина определяется, применяя информацию о расширенной длине (ExtLength). Допустимые значения Length и ExtLength приведены в таблице А.14.

Таблица А.14 — Определение полубайта Length и октета ExtLength

I-Service	Length	ExtLength	Определение
0	0	н/д	Нет услуги, Длина ISDU равна 1. Использование протокола
0	1	н/д	Устройство занято, Длина ISDU равна 1. Использование протокола
0	От 2 до 15	н/д	Зарезервировано и не будет применяться
От 1 до 15	0	н/д	Зарезервировано и не будет применяться
От 1 до 15	1	От 0 до 16	Зарезервировано и не будет применяться
От 1 до 15	1	От 17 до 238	Длина ISDU указана в ExtLength
От 1 до 15	1	От 239 до 255	Зарезервировано и не будет применяться
От 1 до 15	От 2 до 15	Н/д	Длина ISDU

А.5.4 Индекс и Субиндекс

Адрес параметра объекта данных, передаваемого с использованием ISDU, определяется в элементе Индекс. Индекс имеет диапазон значений от 0 до 65535 (см. ограничения в В.2.1). Значения Индекса 0 и 1 отвергаются Устройством.

Требования для Устройства по поддержке всех значений Индекса и Субиндекса отсутствуют. Если значения Индекса или Субиндекса не поддерживаются, Устройство посылает отрицательный ответ.

Адрес элемента данных структурированного параметра объекта данных, передаваемого с использованием ISDU, определяется в элементе Субиндекс. Субиндекс имеет диапазон значений от 0 до 255, при этом значение «0» применяется для ссылки к целому объекту данных (см. рисунок 5).

В таблице А.15 приведены форматы Индекса, используемого в ISDU, в зависимости от передаваемых параметров.

Таблица А.15 — Использование форматов Индекса

Индекс	Субиндекс	Формат Индекса ISDU
От 0 до 255	0	8-битовый Индекс
От 0 до 255	От 1 до 255	8-битовый Индекс и 8-битовый Субиндекс
От 256 до 65535	От 0 до 255	16-битовый Индекс и 8-битовый Субиндекс (см. примечание)
Примечание — См. ограничения на диапазон индекса в подразделе В.2.1.		

А.5.5 Данные

Элемент Данные может содержать объекты данных, установленные в приложении В, или объекты данных, специфические для Устройства. Длина данных соответствует значению в элементе длина Length за вычетом элементов протокола ISDU.

А.5.6 Проверка ISDU (CHKPDU)

Элемент CHKPDU обеспечивает защиту целостности данных. Отправитель вычисляет значение CHKPDU путем XOR-обработки всех октетов ISDU, включая CHKPDU с предварительным значением «0», которое затем заменяется результатом вычислений (см. рисунок А.18).

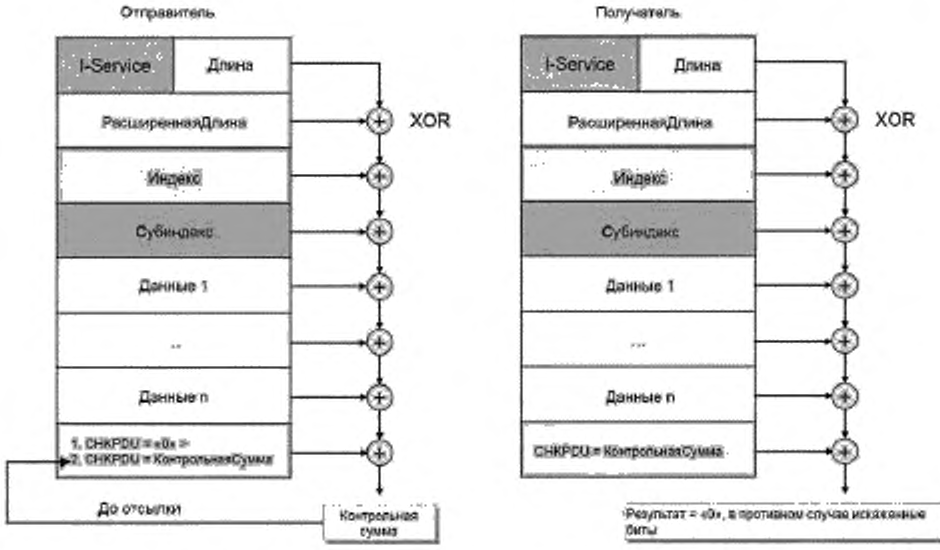


Рисунок А.18 — Проверка целостности ISDU через CHKPDU

Получатель проверяет, приводит ли XOR-обработка всех октетов ISDU к результату «0» (см. рисунок А.18). Если результат отличен от «0», должна производиться обработка ошибки. См. также А.1.6.

А.5.7 Примеры ISDU

На рисунках А.19 приведены типичные примеры форм запроса ISDU, которые объясняются в следующих параграфах.



1) Общая РасширеннаяДлина ISDU = *n* (от 1 до 238), Длина = 1 («0001»).

Рисунок А.19 — Примеры форм запроса для ISDU

Запрос ISDU в примере 1 включает один элемент Индекса, разрешающий адресацию от 0 до 254 (см. таблицу А.15). В этом запросе Субиндекс равен «0», и все содержимое Индекса — это Данные 1 со старшим октетом и Данные 2 с младшим октетом. Общая длина равна 5 («0101»).

Запрос ISDU в примере 2 включает один элемент Индекса, разрешающий адресацию от 0 до 254, и один элемент Субиндекса, разрешающий адресацию элемента структуры данных. Общая длина равна 6 («0110»).

Запрос ISDU в примере 3 включает два элемента Индекса, разрешающих адресацию от 256 до 65535 (см. таблицу А.15), и элемент Субиндекса, позволяющий адресовать элемент структуры данных. Общая длина равна 7 («0111»).

Запрос ISDU в примере 4 включает один элемент Индекса и элемент расширенной длины ExtLength, показывающий число элементов ISDU (*n*), делая возможной длину от 17 до 238. В данном случае элемент Длина имеет значение «1».

«Пустой» запрос в примере 5 применяется, чтобы показать, что никакой услуги не требуется.

На рисунке А.20 приведены типовые примеры ответных ISDU, которые объясняются в следующих разделах.



[1)] Минимальная длина = 2 («0010»)
[2)] Общая РасширеннаяДлина ISDU = *n* (от 17 до 238);
Длина = 1 («0001»)

Рисунок А.20 — Примеры ответных ISDU

Ответное ISDU в примере 1 показывает минимальное значение 2 для элемента Длина («0010»).

ISDU ответа в примере 2 показывает два элемента Данных и общую длину 4 в элементе Длина («0100»). Данные 1 переносят старший октет, а Данные 2 — младший октет.

Ответный ISDU в примере 3 показывает элемент расширенной длины ExtLength, указывающий число элементов в ISDU (*n*), допускающий длину от 17 до 238. В данном случае элемент Длина имеет значение «1».

ISDU ответа «Занято» в примере 4 применяется, когда в текущий момент времени Устройство не в состоянии ответить на запрос чтения из Ведущего узла из-за необходимой подготовки к ответу.

На рисунке А.21 показаны типовые примеры как ISDU запросов чтения, так и ISDU запросов записи, которые объясняются в следующих параграфах.

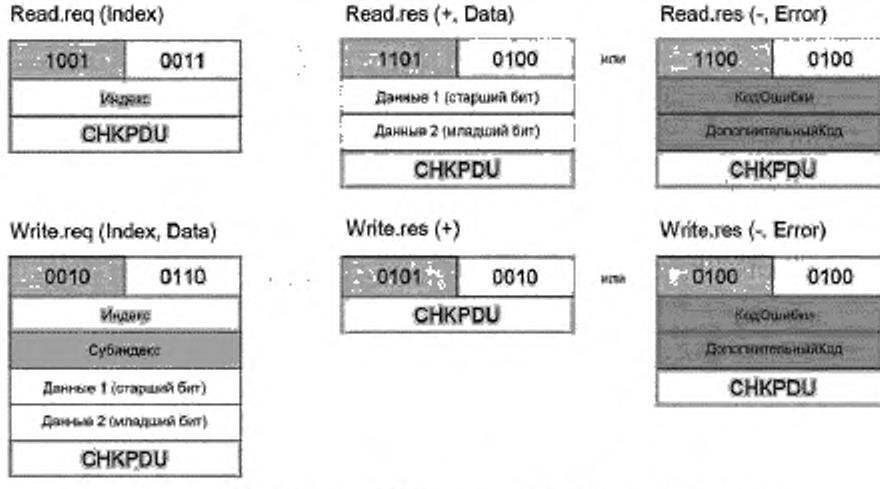


Рисунок А.21 — Примеры ISDU запросов чтения и запросов записи

Код I-Service запроса чтения — «1001». В соответствии с таблицей А.13 он включает элемент Индекс. Успешный (+) ответ на запрос чтения из Устройства с кодом «1101» показан рядом с запросом с двумя элементами Данных. Общая длина равна 4 («0100»). Неуспешный (–) ответ на запрос чтения из Устройства с кодом «1100» показан рядом справа. Он содержит Тип ошибки с двумя элементами Данных: Код ошибки и Дополнительный код (см. приложение С).

Код I-Service запроса записи — «0010». В соответствии с таблицей А.13 он включает элемент Индекс и элемент Субиндекс. Успешный (+) ответ Устройства на запрос записи с кодом «0101» показан рядом, он не содержит элементов Данных. Общая длина равна 2 («0100»). Неуспешный (–) ответ Устройства на запрос чтения с кодом «1100» показан рядом справа. Он содержит Тип ошибки с двумя элементами Данных: Код ошибки и Дополнительный код (см. приложение С).

А.6 Общая структура и кодирование События

А.6.1 Общие положения

Назначение и общая структура памяти События определены в 7.3.8.1 и таблице 56. Данная память размещает код события StatusCode, несколько описателей события EventQualifiers и их соответствующие коды событий EventCode. Кодирование данных элементов памяти устанавливается в А.6.1—А.6.5.

А.6.2 Код состояния StatusCode типа 1 (без деталей)

На рисунке А.22 показана структура данного Кода состояния.

Примечание 1 — Код состояния типа 1 применяется только в Событиях, сгенерированных устаревшими устройствами (см. 7.3.8.1).

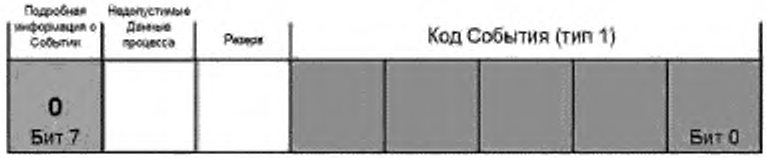


Рисунок А.22 — Структура Кода состояния типа 1

Биты от 0 до 4: Код События типа 1

Кодирование этой структуры данных перечислено в таблице А.16. Коды Событий отражены в кодах Событий типа 2, как перечислено в приложении D. Дополнительная информация приводится в 7.3.8.2.

Таблица А.16 — Отображение Кодов Событий типа 1

Код События типа 1	Код События типа 2	Экземпляр	Тип	Режим
****1	0xFF80	Приложение	Уведомление	Однократное Событие
***1*	0xFF80	Приложение	Предупреждение	Однократное Событие
1	0x6320	Приложение	Ошибка	Однократное Событие
*1***	0xFF80	Приложение	Ошибка	Однократное Событие
1****	0xFF10	Приложение	Ошибка	Однократное Событие
Обозначения: * — безразлично; типа 2 — см. таблицы D.1 и D.2.				

Бит 5: Зарезервирован

Данный бит зарезервирован и будет установлен в нуль в Коде состояния типа 1.

Бит 6: Зарезервирован

Примечание 2 — Данный бит применяется в устаревшем протоколе (см. [8]) для индикации недопустимых PD Pdinvalid.

Бит 7: Подробная информация о Событии

Данный бит указывает, что подробная информация о событии отсутствует. В Коде состояния типа 1 он всегда установлен в нуль.

А.6.3 Код состояния StatusCode типа 2 (с деталями)

На рисунке А.23 показана структура Кода состояния типа 2.



Рисунок А.23 — Структура Кода состояния типа 2

Биты от 0 до 5: Активированные События

Каждый бит привязан к Событию в памяти (см. 7.3.8.1), как показано на рисунке А.24. Бит 0 связан с Событием 1, бит 1 связан с Событием 2 и т. д. Бит со значением «1» указывает, что соответствующие Описатель События EventQualifier и Код События EventCode были введены в память в допустимых форматах. Бит со значением «0» указывает на недопустимый вход.

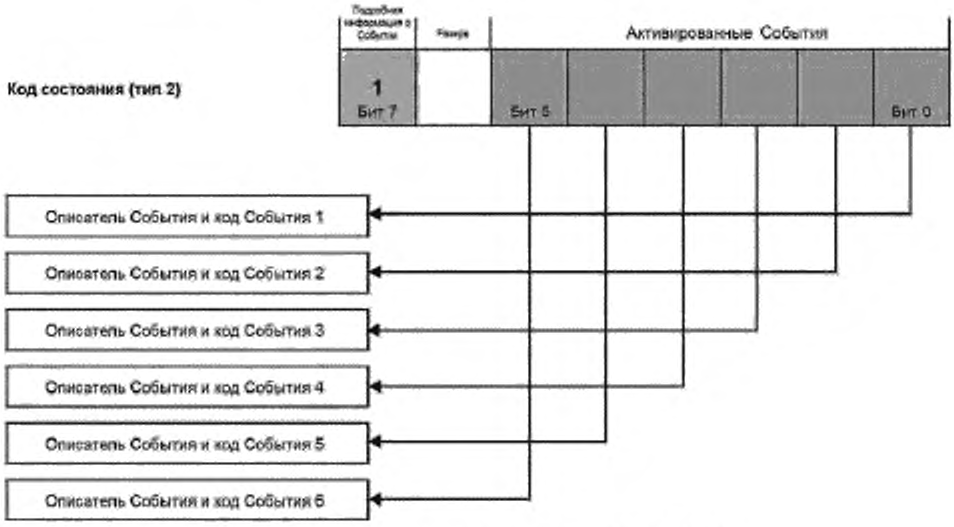


Рисунок А.24 — Индикация активированных Событий

Бит 6: Зарезервирован

Данный бит зарезервирован и устанавливается в нуль.

Примечание — Данный бит применяется в версии устаревшего протокола (см. [8]) для индикации недопустимых PD Pdinvalid.

Бит 7: Подробная информация о Событии

Данный бит указывает, что доступна подробная информация о Событии. В Коде состояния типа 2 он всегда установлен.

А.6.4 Описатель События EventQualifier

Структура описателя События показана на рисунке А.25.

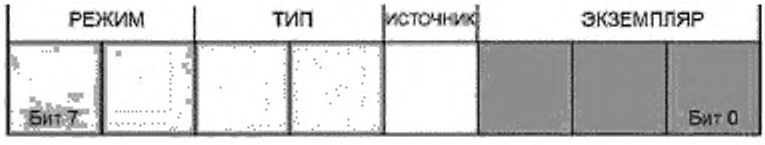


Рисунок А.25 — Структура описателя События

Биты от 0 до 2: ЭКЗЕМПЛЯР

Данные биты указывают конкретный источник (экземпляр) События, таким образом оптимизируя его вычисление на стороне получателя. Допустимые значения поля ЭКЗЕМПЛЯР приведены в таблице А.17.

Таблица А.17 — Значения поля ЭКЗЕМПЛЯР

Значение	Определение
0	Неизвестен
От 1 до 3	Зарезервированы
4	Приложение
От 5 до 7	Зарезервированы

Бит 3: ИСТОЧНИК

Данный бит указывает источник События. Допустимые значения ИСТОЧНИКА приведены в таблице А.18.

Таблица А.18 — Значения бита ИСТОЧНИК

Значение	Определение
0	Устройство (удаленный источник)
1	Ведущий узел (местный источник)

Биты от 4 до 5: ТИП

Данные биты показывают категорию События. Допустимые значения поля ТИП приведены в таблице А.19.

Таблица А.19 — Значения поля ТИП

Значение	Определение
0	Зарезервированы
1	Уведомление
2	Предупреждение
3	Ошибка

Биты от 6 до 7: РЕЖИМ

Данные биты показывают режим События. Допустимые значения поля РЕЖИМ приведены в таблице А.20.

Таблица А.20 — Значения поля РЕЖИМ

Значение	Определение
0	Зарезервировано
1	Однократное Событие
2	Событие исчезает
3	Событие появляется

А.6.5 Код События EventCode

Вход Код События содержит идентификатор фактического События. Допустимые значения кодов Событий приведены в приложении D.

Приложение В
(обязательное)

Параметры и команды

В.1 Страницы 1 и 2 Непосредственных параметров

В.1.1 Обзор

В принципе, разработчик Устройства располагает большим пространством памяти для параметров и команд, как показано на рисунке 5. Однако небольшие датчики с ограниченным числом параметров и ограниченными ресурсами нуждаются в простом подмножестве параметров. Интерфейс SDCI предлагает так называемые страницы 1 и 2 Непосредственных параметров с упрощенными методами доступа (канал связи страниц в соответствии с таблицей А.1) для удовлетворения данного требования.

Диапазон Непосредственных параметров структурирован, как показано на рисунке В.1. Он разделен на страницу 1 и страницу 2.

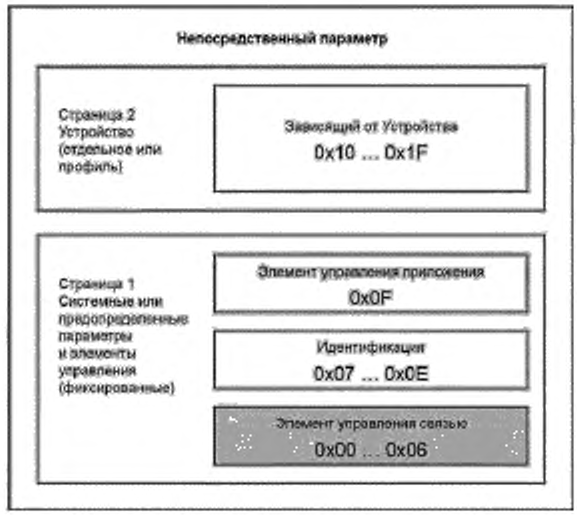


Рисунок В.1 — Классификация и отображение Непосредственных параметров

Диапазон страницы 1 — от 0x00 до 0x0F. Он включает следующие категории параметров:

- управление связью;
- параметры идентификации;
- управление приложением.

AL Ведущего узла предоставляет доступ только для чтения к странице 1 Непосредственных параметров (см. 8.2.1) через Индекс 0. Отдельные октеты могут считываться через Индекс 0 и соответствующий Субиндекс. Субиндекс 1 указывает на адрес 0x00, а Субиндекс 16 — на адрес 0x0F.

Диапазон страницы 2 — от 0x10 до 0x1F. Данная страница включает параметры, используемые при необходимости технологией конкретного Устройства. AL Ведущего узла предоставляет доступ чтения/записи к странице 2 Непосредственных параметров в форме объектов данных (см. 8.2.1) через Индекс 1. Отдельные октеты могут записываться или считываться через Индекс 1 и соответствующий Субиндекс. Субиндекс 1 указывает на адрес 0x10, а Субиндекс 16 — на адрес 0x1F.

Устройство всегда возвращает «0» при доступе на запись к адресам Непосредственных параметров, которые не реализованы (например, в случае зарезервированных адресов параметров или неподдерживаемых необязательных параметров). Устройство игнорирует доступ на запись к нереализованным параметрам.

Структура страниц 1 и 2 Непосредственных параметров устанавливается в таблице В.1.

Таблица В.1 — Страницы 1 и 2 Непосредственных параметров

Адрес	Имя параметра	Доступ	Ссылка на реализацию	Описание
Страница 1 Непосредственных параметров				
0x00	Команда Ведущего узла	З	Обязательный/ см. В.1.2	Команда Ведущего узла: переключить рабочие состояния (см. примечание 1)
0x01	Время цикла Ведущего узла	Ч/З	Обязательный/ см. В.1.3	Фактическая длительность цикла, используемая Ведущим узлом для адресации Устройства. Может применяться как параметр для мониторинга PD
0x02	Минимальное время цикла MinCycleTime	Ч	Обязательный/ см. В.1.3	Минимальная длительность цикла, поддерживаемая Устройством. Это рабочая характеристика Устройства, зависящая от его технологии и реализации
0x03	Возможности М-последовательности	Ч	Обязательный/ см. В.1.4	Информация о реализованных опциях, связанных с М-последовательностями и физической конфигурацией
0x04	ID ревизии	Ч/З	Обязательный/ см. В.1.5	ID версии используемого протокола для реализации (устанавливается в 0x11)
0x05	Входные PD	Ч	Обязательный/ см. В.1.6	Объем и структура входных PD из Устройства в Ведущий узел
0x06	Выходные PD	Ч	Обязательный/ см. В.1.7	Объем и структура выходных PD из Ведущего узла в Устройство
0x07	ID Изготовителя 1 (старший байт)	Ч	Обязательный/ см. В.1.8	Уникальный ID Изготовителя (см. примечание 2)
0x08	ID Изготовителя 2 (младший байт)			
0x09	ID Устройства 1 (октет 1, старший байт)	Ч/З	Обязательный/ см. В.1.9	Уникальная идентификация Устройства, размещенная изготовителем
0x0A	ID Устройства 2 (октет 1)			
0x0B	ID Устройства 3 (октет 0, младший байт)			
0x0C	ID функции 1 (старший байт)	Ч	См. В.1.10	Зарезервировано (Изготовитель установит оба октета в «0x00»)
0x0D	ID функции 2 (младший байт)			
0x0E		Ч	Зарезервировано	
0x0F	Команда системы	З	Необязательный/см. В.1.11	Интерфейс команд только для пользователей приложений и Устройств без поддержки ISDU (см. примечание 1)
Страница 2 Непосредственных параметров				
0x10… 0x1F	Определяемые поставщиком	Необязательный	Необязательный/см. В.1.12	Параметры, определяемые Устройством

Примечание 1 — Операция чтения возвращает неопределенные значения.

Примечание 2 — ID Изготовителя назначается консорциумом IO-Link.

B.1.2 Команды Ведущего узла

Приложение Ведущего узла может проверить состояние Устройства или контролировать его поведение с помощью команды Ведущего узла (см. 7.3.7).

Допустимые значения для данных параметров установлены в таблице B.2.

Таблица B.2 — Типы команд Ведущего узла

Значение	Команда Ведущего узла	Описание
От 0x00 до 0x59	Зарезервировано	
0x5A	Fallback	Переход из связи в режим SIO. Устройство будет выполнять данный переход после истечения трех времен цикла Ведущего узла и до истечения 500 мс после команды Ведущего узла
От 0x5B до 0x94	Зарезервировано	
0x95	MasterIdent	Указывает на версию Ведущего узла выше 1.0
0x96	DeviceIdent	Начинает проверку страницы Непосредственных параметров на предмет измененных входов
0x97	DeviceStartup	Переключает Устройство из состояния OPERATE или PREOPERATE в состояние STARTUP
0x98	ProcessDataOutputOperate	Допустимые выходные PD
0x99	DeviceOperate	Выходные PD недопустимы или недоступны. Переключает Устройство из состояния STARTUP или PREOPERATE в состояние OPERATE
0x9A	DevicePreoperate	Переключает Устройство из состояния STARTUP в состояние PREOPERATE
От 0x9B до 0xFF	Reserved	

B.1.3 Время цикла Ведущего узла MasterCycleTime и минимальное время цикла MinCycleTime

Время цикла Ведущего узла MasterCycleTime является параметром Ведущего узла и устанавливает фактическое время цикла конкретного порта.

Минимальное время цикла MinCycleTime — это параметр Устройства, который информирует Ведущий узел о кратчайшем времени цикла, поддерживаемом Устройством.

Использование времени цикла Ведущего узла и минимального времени цикла описано в A.3.7. Структура данных двух параметров показана на рисунке B.2.

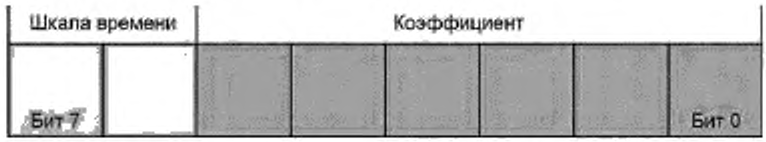


Рисунок B.2 — Минимальное время цикла MinCycleTime

Биты от 0 до 5: Multiplier

Данные биты содержат 6-битовый коэффициент для вычисления Времени цикла Ведущего узла или Минимального времени цикла. Допустимые значения — от 0 до 63.

Биты от 6 до 7: TimeBase

Данные биты устанавливают шкалу времени для вычисления MasterCycleTime или MinCycleTime.

Когда все биты нулевые (двоичный код 0x00), Устройство не имеет минимального времени цикла. В данном случае Ведущий узел будет применять вычисленный наихудший случай временных характеристик типа M-последовательности, используемой Устройством, и максимальные времена для t_A и t_2 (см. A.3.4—A.3.6).

Допустимые комбинации для шкалы времени и коэффициента приведены в таблице B.3 вместе с результирующими значениями для MasterCycleTime и MinCycleTime.

Таблица В.3 — Возможные значения MasterCycleTime и MinCycleTime

Код шкалы времени	Значение шкалы времени	Вычисление	Время цикла
00	0,1 мс	Multiplier × TimeBase	От 0,4 мс до 6,3 мс
01	0,4 мс	6,4 мс +	От 6,4 мс до 31,6 мс
10	1,6 мс	32,0 мс +	От 32,0 мс до 132,8 мс
11	Зарезервировано	Зарезервировано	Зарезервировано
Примечание — Значение 0,4 вытекает из минимально возможного времени передачи в соответствии с А.3.7.			

В.1.4 Параметр возможностей M-последовательностей M-sequenceCapability

Структура параметра M-sequenceCapability показана на рисунке В.3.

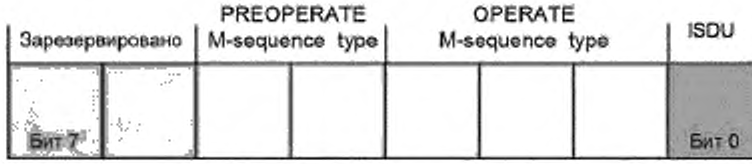


Рисунок В.3 — Параметр M-sequenceCapability

Бит 0: ISDU

Данный бит указывает, поддерживается канал связи ISDU или нет. Допустимые значения приведены в таблице В.4.

Таблица В.4 — Значения бита ISDU

Значение	Определение
0	ISDU не поддерживается
1	ISDU поддерживается

Биты от 1 до 3: Кодирование OPERATE M-sequencetype

Данный параметр указывает допустимые типы М-последовательности во время состояния OPERATE. Допустимые коды приведены в таблице А.9 для устаревших Устройств и в таблице А.10 — для Устройств, соответствующих настоящему стандарту.

Биты от 4 до 5: Кодирование

Данный параметр указывает допустимые типы М-последовательности во время состояния PREOPERATE. Допустимые коды приведены в таблице А.8.

Биты от 6 до 7: Зарезервировано

Данные биты зарезервированы и устанавливаются в нуль в новой версии спецификации.

В.1.5 Идентификатор редакции RevisionID (RID)

Параметр идентификатора редакции RevisionID — это двухцифровой номер версии протокола SDCI, используемого в настоящее время на Устройстве. Его структура показана на рисунке В.4. Начальное значение RevisionID при включении питания — это внутреннее значение для протокола RevisionID. Оно может быть перезаписано (см. 10.6.3) до следующего включения питания.

Настоящая редакция стандарта указывает версию протокола 1.1.

Примечание — Устаревшая версия 1.0 протокола определена в [8].

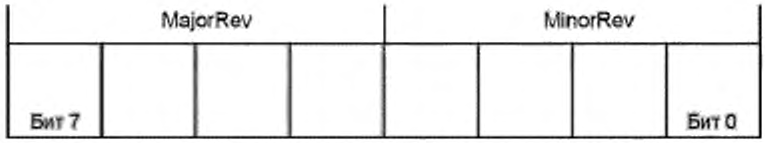


Рисунок В.4 — Идентификатор редакции RevisionID

Биты от 0 до 3: MinorRev

Данные биты содержат младшую цифру номера версии, например, 0 для версии протокола 1.0. Разрешенные значения — от 0x0 до 0xF.

Биты от 4 до 7: MajorRev

Данные биты содержат старшую цифру номера версии, например 1 для версии протокола 1.0. Разрешенные значения — от 0x0 до 0xF.

В.1.6 Параметр ProcessDataIn

Структура параметра ProcessDataIn показана на рисунке В.5.

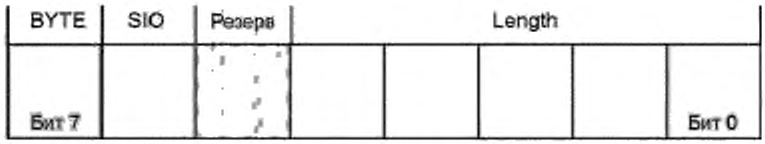


Рисунок В.5 — Параметр ProcessDataIn

Биты от 0 до 4: Length

Данные биты содержат длину входных данных (PD из Устройства в Ведущий узел) в единицах длины, назначенных в бите БАЙТ параметра. Допустимые коды установлены в таблице В.6.

Бит 5: Резерв

Данный бит зарезервирован и устанавливается в нуль в этой версии спецификации.

Бит 6: SIO

Данный бит указывает, предоставляет ли Устройство коммутационный сигнал в режиме SIO. Допустимые значения для SIO приведены в таблице В.5.

Таблица В.5 — Значения SIO

Значение	Определение
0	Режим SIO не поддерживается
1	Режим SIO поддерживается

Бит 7: BYTE

Данный бит указывает единицу измерения для Length. Допустимые значения для BYTE и окончательное определение длины PD совместно с Length приведены в таблице В.6.

Таблица В.6 — Разрешенные комбинации полей BYTE и Length

BYTE	Length	Определение
0	0	Нет PD
0	1	1-битовые PD, структурированные в битах

Окончание таблицы В.6

BYTE	Length	Определение
0	*n* (2—15)	*n*-битовые PD, структурированные в битах
0	16	16-битовые PD, структурированные в битах
0	От 17 до 31	Зарезервировано
1	0, 1	Зарезервировано
1	2	3-октетные PD, структурированные в октетах
1	*n* (3—30)	(*n* + 1)-октетные PD, структурированные в октетах
1	31	32-октетные PD, структурированные в октетах

В.1.7 Параметр ProcessDataOut

Структура параметра ProcessDataOut такая же, как структура параметра ProcessDataIn, за исключением того, что бит 6 (SIO) зарезервирован.

В.1.8 Идентификатор изготовителя VendorID (VID)

Данные октеты содержат распространенное уникальное значения для каждого изготовителя.

Примечание — Идентификатор изготовителя назначается консорциумом IO-Link.

В.1.9 Идентификатор Устройства DeviceID (DID)

Данные октеты содержат используемый в настоящее время идентификатор Устройства. Значение «0» не разрешено. Начальное значение идентификатора Устройство — это внутреннее значение DeviceID. Оно может быть перезаписано (см. 10.6.2) до следующего включения питания.

Примечание — Параметры связи MinCycleTime, M-sequenceCapability, ProcessDataIn и ProcessDataOut могут быть изменены для достижения совместимости с требуемым DeviceID.

В.1.10 Идентификатор функции FunctionID (FID)

Данный параметр будет определен в последующей версии.

В.1.11 Параметр SystemCommand

Только Устройства без поддержки ISDU будет применять параметр SystemCommand на странице 1 Непосредственных параметров. Реализация параметра SystemCommand является необязательной. Подробное описание функций параметра SystemCommand приводится в таблице В.9.

Примечание — Параметр SystemCommand на странице 1 Непосредственных параметров не предоставляет положительного или отрицательного ответа при выполнении выбранной функции.

В.1.12 Специфическая для Устройства страница 2 Непосредственных параметров

Специфические для Устройства Непосредственные параметры — это набор параметров, доступный для специфической технологии Устройства. Реализация специфических для Устройства Непосредственных параметров не является обязательной.

Примечание — Полный перечень параметров страницы 2 Непосредственных параметров является доступным для чтения или записи через индекс 1 (см. В.1.1).

В.2 Предопределенные параметры Устройства

В.2.1 Обзор

Многие различные технологии и конструкции датчиков и исполнительных устройств требуют отдельного и легкого доступа к сложным параметрам и командам за пределами возможностей страницы 2 Непосредственных параметров. С точки зрения Ведущего узла эти сложные параметры и команды называются объектами данных приложения. Так называемые контейнеры ISDU являются средствами обмена объектами данных приложения или неполными объектами данных. Индекс ISDU применяется для адресации объектов данных. На рисунке В.6 показано общее отображение объектов данных для передачи ISDU.



Рисунок В.6 — Индексное пространство объектов данных ISDU

В разделе В.2 содержатся определения и требования для реализации приложений специальных технических Устройств. Правила реализации для параметров и команд устанавливаются в таблице В.7.

Таблица В.7 — Правила реализации для параметров и команд

Номер правила	Спецификация правила
1	Все параметры Индекса должны быть считываемыми и записываемыми, как целый объект данных через Субиндекс 0
2	Приложения специальных технических устройств должны разрешать конфликты зависимых наборов параметров во время параметризации
3	Длительность запроса услуги ISDU ограничена (см. таблицу 97). Приложение Ведущего узла может завершить услуги ISDU после такого превышения лимита времени
4	Команды приложения (например, настройка в режиме обучения, возврат к заводским установкам и т. п.) рассматриваются как параметры. Инициированное выполнение команд приложения подтверждается положительным ответом услуги — AL_Write.res(+). Отрицательный ответ услуги — AL_Write.res(–) — указывает на неуспешное выполнение команды приложения. В обоих случаях должны учитываться ограничения на превышение лимита времени ISDU (см. таблицу 97)

В таблице В.8 устанавливаются назначения объектов данных (параметров и команд) диапазону Индекса ISDU.

Значения индексов, превышающие 2, относятся к ISDU.

Таблица В.8 — Назначение Индекса объектов данных (параметров Устройства)

Индекс (десятичный)	Имя объекта	Доступ	Длина	Тип данных	M/O/C	Примечание
0x0000 (0)	Страница 1 Непосредственных параметров	Ч		RecordT	M	Перенаправлено на канал связи страниц (см. 10.7.5)
0x0001 (1)	Страница 2 Непосредственных параметров	Ч/З		RecordT	M	Перенаправлено на канал связи страниц (см. 10.7.5)
0x0002 (2)	Команда системы	З	1 Октет	UIntegerT	M/O	Определение кода команды (см. В.2.2)
0x0003 (3)	Индекс DS	Ч/З	Переменная	RecordT	M	Набор объектов данных для хранения (см. В.2.3)
0x0004—0x000B (4—11)	Зарезервировано					Зарезервировано для исключительных операций
0x000C (12)	Блокировки доступа к Устройству	Ч/З	2 Октета	RecordT	C	Стандартизированные функции блокировки Устройства (см. В.2.4)
0x000D (13)	Характеристики профиля	Ч	Переменная	ArrayT of UIntegerT16	O	Характеристики профиля (см. В.2.5)
0x000E (14)	Описатель входных PD	Ч	Переменная	ArrayT of OctetStringT3	O	Зарезервировано для профиля Устройства (см. В.2.6)
0x000F (15)	Описатель выходных PD	Ч	Переменная	ArrayT of OctetStringT3	O	Зарезервировано для профиля Устройства (см. В.2.7)
0x0010 (16)	Имя изготовителя	Ч	Максимум 64 октета	StringT	M	Справочный (см. В.2.8)
0x0011 (17)	Текст изготовителя	Ч	Максимум 64 октета	StringT	O	Дополнительная информация об изготовителе (см. В.2.9)
0x0012 (18)	Название продукта	Ч	Максимум 64 октета	StringT	M	Подробное описание продукта или имя типа (см. В.2.10)
0x0013 (19)	ID продукта	Ч	Максимум 64 октета	StringT	O	Идентификация продукта или типа (подробное описание см. в В.2.11)
0x0014 (20)	Текст продукта	Ч	Максимум 64 октета	StringT	O	Описание функции или характеристики Устройства (см. В.2.12)
0x0015 (21)	Заводской номер	Ч	Максимум 16 октетов	StringT	O	Заводской номер, определяемый изготовителем (см. В.2.13)
0x0016 (22)	Версия оборудования	Ч	Максимум 64 октета	StringT	O	В формате, определяемом изготовителем (см. В.2.14)
0x0017 (23)	Версия встроенного ПО	Ч	Максимум 64 октета	StringT	O	В формате, определяемом изготовителем (см. В.2.15)
0x0018 (24)	Ярлык, связанный с приложением	Ч/З	Минимум 16, максимум 32 октета	StringT	O	Положение или функция ярлыка определяется пользователем (см. В.2.16)

Окончание таблицы В.8

Индекс (десятичный)	Имя объекта	Доступ	Длина	Тип данных	М/О/С	Примечание
0x0019—0x001F (25—31)	Зарезервировано				О	
0x0020 (32)	Счетчик ошибок	Ч	2 октета	UIntegerT	О	Ошибки с момента включения питания или сброса (см. В.2.17)
0x0021—0x0023 (33—35)	Зарезервировано					
0x0024 (36)	Состояние Устройства	Ч	1 октет	UIntegerT	О	Содержит текущее состояние Устройства (см. В.2.18)
0x0025 (37)	Подробное состояние Устройства	Ч	Переменная	arrayT of OctetStringT3	О	См. В.2.19
0x0026—0x0027 (38—39)	Зарезервировано					
0x0028 (40)	Ввод PD	Ч	Длина pd	Зависит от Устройства	О	Читать последние допустимые PD из канала Pin (см. В.2.19)
0x0029 (41)	Вывод PD	Ч	Длина pd	Зависит от Устройства	О	Читать последние допустимые PD из канала PDout (см. В.2.21)
0x002—0x002F (42—47)	Зарезервировано					
0x0030 (48)	Время сдвига	Ч/З	1 октет	RecordT	О	Синхронизация приложения Устройства и М-последовательности (см. В.2.22)
0x0031—0x003F (49—63)	Зарезервировано для профиля					
0x0040—0x00FE (64—254)	Предпочтительный Индекс					Зависит от Устройства (8 бит)
0x00FF (255)	Зарезервировано					
0x0100—0x3FFF (256—16 383)	Расширенный Индекс					Зависит от Устройства (16 бит)
0x4000—0x4FFF (16 384—20 479)	Индекс, зависящий от профиля					Зарезервировано для профиля Устройства
0x5000—0xFFFF (20 480—65 535)	Зарезервировано					

Обозначения:
М — обязательный;
О — необязательный;
С — условный.

B.2.2 Параметр SystemCommand

Устройства с поддержкой ISDU применяют Индекс 0x0002 ISDU для получения параметра команды системы SystemCommand. Команды будут подтверждаться. Положительное подтверждение указывает полное и правильное завершения затребованной команды. Отрицательное подтверждение указывает, что команда не может быть реализована или закончилась с ошибкой. Команда системы будет выполняться в пределах менее 5 с для соответствия временным требованиям ISDU (см. таблицу 97).

Реализация функциональной возможности SystemCommand обязательна для Ведущего узла и необязательна для Устройств. Кодирование параметра SystemCommand определено в таблице B.9.

Таблица B.9 — Кодирование параметра SystemCommand (ISDU)

Команда (шестнадцатеричная)	Команда (десятичная)	Имя команды	M/O	Определение
0x00	0	Зарезервировано		
0x01	1	ParamUploadStart	O	Начать подкачку параметра
0x02	2	ParamUploadEnd	O	Прекратить подкачку параметра
0x03	3	ParamDownloadStart	O	Начать скачивание параметра
0x04	4	ParamDownloadEnd	O	Прекратить скачивание параметра
0x05	5	ParamDownloadStore	O	Завершить параметризацию и запустить DS
0x06	6	ParamBreak	O	Отменить все команды работы с параметрами
От 0x07 до 0x3F	От 7 до 63	Зарезервировано		
От 0x40 до 0x7F	От 64 до 127	Зарезервировано для профиля		
0x80	128	Сброс Устройства	O	
0x81	129	Сброс приложения	O	
0x82	130	Восстановить заводские установки	O	
От 0x83 до 0x9F	От 131 до 159	Зарезервировано		
От 0xA0 до 0xFF	От 160 до 255	Определяется изготовителем		

Примечание — См. 10.3.

Обозначения:
M — обязательный;
O — необязательный.

Команда системы 0x05 (ParamDownloadStore) будет реализовываться в соответствии с 10.4.2, когда Устройство предоставляет параметры, подлежащие сохранению через механизм Хранилища памяти, т. е. параметр Index_List в Индексе 0x0003 не является пустым (см. таблицу B.10).

Реализация системных команд от 0x01 до 0x06, требуемых для блочной параметризации в соответствии с 10.3.5, является необязательной. Однако все данные команды будут реализовываться только вместе (для системной команды 0x05 правило для DS играет решающую роль).

Опции параметра SystemCommand на странице 1 Непосредственных параметров описаны в B.1.11.

B.2.3 Индекс Хранилища данных

В таблице B.10 описаны назначения Индекса DS.

Таблица B.10 — Назначения Индекса DS

Индекс	Субиндекс	Доступ	Имя параметра	Кодирование	Тип данных
0x0003	01	Ч/З	DS_Command	0x00: Зарезервировано 0x01: DS_UploadStart 0x02: DS_UploadEnd 0x03: DS_DownloadStart 0x04: DS_DownloadEnd 0x05: DS_Break От 0x06 до 0xFF: Зарезервировано	UintegerT8 (8 битов)
	02	Ч	State_Property	Бит 0 Зарезервировано Биты 1 и 2: Состояние DS 0b00: Неактивно 0b01: Подкачка 0b10: Скачивание 0b11: DS блокировано Биты от 3 до 6: Зарезервировано Бит 7: DS_UPLOAD_FLAG «1»: Ожидает запрос DS_UPLOAD_REQ «0»: Нет запроса DS_UPLOAD_REQ	UintegerT8 (8 битов)
	03	Ч	Data_Storage_ Size	Число октетов для сохранения всей требуемой информации для замены Устройства (см. 10.4.5). Максимальный размер — 2048 октетов	UintegerT16 (16 битов)
	04	Ч	Parameter_ Checksum	Индикация редакции набора параметров: Контрольная сумма или счетчик изменений (см. 10.4.8)	UintegerT32 (32 бита)
	05	Ч	Index_List	Список сохраняемых индексов параметра (см. таблицу B.11)	OctetStringT (переменная)

Индекс 0x0003 DS содержит всю информацию, которая будет применяться для обработки DS. Данный параметр зарезервирован для служебных обменов информацией между Ведущим узлом и Устройством; Ведущий узел блокирует все запросы доступа из приложения шлюза к данному Индексу (см. рисунок 4). Параметры в данном Индексе 0x0003 определены следующим образом.

DS_Command (Команда DS)

Данный октет содержит команды DS для Устройства.

State_Property (Состояние и свойства)

Данный октет показывает текущее состояние механизма DS. Бит 7 хранится в энергонезависимой памяти. Ведущий узел проверяет Данный бит при запуске и выполняет подкачку параметра, если требуется.

Data_Storage_Size (Размер DS)

Данные четыре октета предоставляют затребованный размер памяти как количество октетов для сохранения всей информации, необходимой для замены Устройства, включая структурную информацию (Индекс, Субиндекс). Тип данных — UintegerT32 (целое без знака длиной 32 бита). Максимальный размер — 2048 октетов. Элементы, которые следует учитывать при вычислении размера, приведены в таблице F.1.

Parameter_Checksum (Контрольная сумма параметров)

Данная контрольная сумма применяется для обнаружения изменений в наборе параметров без чтения всех параметров. Значение контрольной суммы вычисляется в соответствии с процедурой, приведенной в 10.4.8. Устройство изменяет контрольную сумму при любом изменении параметра из набора параметров. Различные наборы параметров имеют различные контрольные суммы. Рекомендуется, чтобы Устройство сохраняло Данный параметр локально в энергонезависимой памяти.

Index_List (Список индексов)

Структура Списка индексов определена в таблице B.11. Каждый список индексов Index_List может иметь до 70 входов (см. таблицу 97).

Таблица В.11 — Структура списка индексов Index_List

Вход	Адрес	Определение	Тип данных
X1	Индекс	Индекс первого сохраняемого параметра	Unsigned16
	Субиндекс	Субиндекс первого сохраняемого параметра	Unsigned8
X2	Индекс	Индекс следующего сохраняемого параметра	Unsigned16
	Субиндекс	Субиндекс следующего сохраняемого параметра	Unsigned8
.....	.....	.....	.....
Xn	Индекс	Индекс последнего сохраняемого параметра	Unsigned16
	Субиндекс	Субиндекс последнего сохраняемого параметра	Unsigned8
Xn+1	Индекс	Маркер окончания Termination_Marker 0x0000: Конец индексного списка Index_List > 0x0000: Новый индекс, содержащий индексный список	Unsigned16

Большие наборы параметров могут обрабатываться через сцепленные Списки индексов. Последние два октета Списка индексов содержат Маркер окончания. Значение «0» указывает на конец Списка индексов. В случае сцепления Маркер окончания устанавливается в следующий Индекс, содержащий Список индексов. Структура следующего Списка индексов такая, как определено в таблице В.11. Таким образом, цепочка списков оканчивается, когда обнаруживается Маркер окончания со значением «0».

В.2.4 Параметр DeviceAccessLocks (Блокировки доступа к Устройству)

Параметр блокировок доступа к Устройству позволяет управлять поведением Устройства. Стандартизованные функции Устройства могут конфигурироваться независимо через флаги, определенные в данном параметре. Конфигурация блокировки доступа к Устройству может быть изменена перезаписью данного параметра. Фактическая установка конфигурации доступна через доступ на чтение данного параметра. Тип данных данного параметра: RecordT of BooleanT. Доступ разрешен только через Субиндекс 0. Данный параметр — необязательный. Если он реализован, то должен находиться в энергонезависимой памяти.

Определены следующие категории блокировки доступа к Устройству:

- доступ на запись параметров (необязательный);
- доступ к DS (обязательный, если Устройство поддерживает Хранилище данных);
- доступ к локальной параметризации (необязательный);
- доступ к локальному интерфейсу пользователя (необязательный).

Возможности блокирования Устройства приведены в таблице В.12.

Таблица В.12 — Возможности блокирования Устройства

Бит	Категория	Определение
0	Доступ (на запись) к параметрам (необязательный)	0: не блокирован (по умолчанию) 1: блокирован
1	Доступ к DS (обязательный, если Устройство поддерживает DS)	0: не блокирован (по умолчанию) 1: блокирован (см. примечание)
2	Доступ к локальной параметризации (необязательный)	0: не блокирован (по умолчанию) 1: блокирован
3	Доступ к локальному интерфейсу пользователя (необязательный)	0: не блокирован (по умолчанию) 1: блокирован
4—15	Зарезервировано	
Примечание — До принятия каких-либо действий Ведущий узел считывает параметр State_Property/ State of Data Storage (см. таблицу В.10).		

Доступ (на запись) к параметрам

Если Данный бит установлен, доступ на запись ко всем параметрам Устройства через интерфейс связи SDCI запрещен для всех параметров чтения/записи, кроме параметра DeviceAccessLocks. Доступ на чтение никак не

затрагивается. Устройство будет реагировать отрицательным ответом на услугу — нет доступа — на доступ на запись, если доступ к параметрам блокирован.

Механизм блокировки доступа (на запись) к параметрам не будет блокировать скачивания из DS (между DS_DownloadStart и DS_DownloadEnd или DS_Break).

DataStorage (DS)

Если в Устройстве установлен Данный бит, механизм DS отключен (см. 10.4.2 и 11.3.3). В данном случае Устройство будет реагировать на доступ записи (в своем Индексе Хранилища памяти) отрицательным ответом на услугу — нет доступа — (см. B.2.3). Доступ на запись к Индексам Хранения данных никак не затрагивается.

Данная установка также указывается в параметре StateProperty в Индексе DS.

Localparameterization (локальная параметризация)

Если Данный бит установлен, параметризация через локальные элементы управления на Устройстве запрещена.

Localuserinterface (локальный интерфейс пользователя)

Если Данный бит установлен, функционирование человеко-машинного интерфейса на Устройстве запрещено.

B.2.5 Параметр ProfileCharacteristic (Характеристики профиля)

Данный параметр содержит список идентификаторов профиля ProfileIdentifiers (PID's), соответствующий Профилю Устройства, реализованному в Устройстве.

Примечание — Подробная информация приведена в [7].

B.2.6 Параметр PD InputDescriptor (Описатель ввода PD)

Данный параметр содержит описание структуры данных входных PD для Устройства с профилем.

Примечание — Подробная информация приведена в [7].

B.2.7 Параметр PD OutputDescriptor (Описатель вывода PD)

Данный параметр содержит описание структуры данных входных PD для Устройства с профилем.

Примечание — Подробная информация приведена в [7].

B.2.8 Параметр VendorName (Имя изготовителя)

Параметр VendorName содержит только одно из имен изготовителя, перечисленных для назначенного идентификатора изготовителя VendorID. Данный параметр является объектом данных только для чтения. Тип данных: StringT максимальной фиксированной длины 64. Данный параметр является обязательным.

Примечание — Список имен изготовителей, связанный с данным VendorID, поддерживается консорциумом IO-Link.

B.2.9 Параметр VendorText (Текст изготовителя)

Данный параметр содержит дополнительную информацию об изготовителе. Параметр является объектом данных только для чтения. Тип данных: StringT максимальной фиксированной длины 64. Данный параметр является обязательным.

B.2.10 Параметр ProductName (Имя продукта)

Параметр ProductName содержит полное имя продукта. Параметр является объектом данных только для чтения. Тип данных: StringT максимальной фиксированной длины 64. Данный параметр является обязательным.

Примечание — Ожидается, что соответствующий вход в списке вариантов Устройства файла IODD будет соответствовать этому параметру.

B.2.11 Параметр ProductID (Идентификатор продукта)

Параметр ProductID будет содержать продукт, определяемый изготовителем, или идентификацию типа Устройства. Параметр является объектом данных только для чтения. Тип данных: StringT максимальной фиксированной длины 64. Данный параметр является обязательным.

B.2.12 Параметр ProductText (Текст продукта)

Параметр ProductText будет содержать дополнительную информацию о продукте для Устройства, такую как категория продукта (например, Подавление фотоэлектрического фона, Ультразвуковой датчик расстояния, Датчик давления и т. п.). Параметр является объектом данных только для чтения. Тип данных: StringT максимальной фиксированной длины 64. Данный параметр является обязательным.

B.2.13 Параметр SerialNumber (Заводской номер)

Параметр SerialNumber будет содержать уникальный определяемый изготовителем код для каждого отдельного Устройства. Параметр является объектом данных только для чтения. Тип данных: StringT максимальной фиксированной длины 16. Данный параметр является необязательным.

B.2.14 Параметр HardwareRevision (Версия оборудования)

Параметр HardwareRevision будет содержать определяемый изготовителем код для версии оборудования Устройства. Параметр является объектом данных только для чтения. Тип данных: StringT максимальной фиксированной длины 64. Данный параметр является необязательным.

B.2.15 Параметр FirmwareRevision (Версия встроенного программного обеспечения)

Параметр FirmwareRevision будет содержать определяемый изготовителем код для версии встроенного программного обеспечения. Параметр является объектом данных только для чтения. Тип данных: StringT максимальной фиксированной длины 64. Данный параметр является необязательным.

B.2.16 Параметр ApplicationSpecificTag (Ярлык, связанный с приложением)

Параметр ApplicationSpecificTag будет предоставляться как объект данных чтения/записи для приложения пользователя. Он может применяться как «ярлык функции» (роль Устройства) или «ярлык положения» (местоположение Устройства). Тип данных: StringT с минимальной фиксированной длиной 16 и максимальной фиксированной длиной 32. По умолчанию рекомендуется заполнять Данный параметр значением «***». Данный параметр необязательный.

Примечание — В процессах автоматизации, как правило, эта длина равна 32 октетам.

B.2.17 Параметр ErrorCount (Счетчик ошибок)

Параметр ErrorCount предоставляет информацию о количестве ошибок, произошедших в Устройстве со времени включения питания или сброса в исходное состояние. Использование данного параметра определяется либо изготовителем, либо Устройством. Тип данных: UIntegerT длиной 16 битов. Параметр является объектом данных только для чтения. Данный параметр необязательный.

B.2.18 Параметр DeviceStatus (Состояние Устройства)

B.2.18.1 Обзор

Параметр DeviceStatus будет предоставлять информацию об условиях Устройства (диагностику). Тип данных: UIntegerT длиной 8 битов. Параметр является объектом данных только для чтения. Данный параметр необязательный.

Определяются условия Устройства, показанные в таблице B.13. Они будут генерироваться приложениями Устройства. Параметр DeviceStatus может считываться любой программой PLC или инструментальными средствами, такими как модуль управления активами (см. раздел 11).

В таблице B.13 перечислена различная информация состояния Устройства. Критерии для таких индикаций определены в B.2.18.2—B.2.18.5.

Таблица B.13 — Параметр состояния устройства DeviceStatus

Значение	Определение
0	Устройство функционирует правильно
1	Требуется техническое обслуживание (см. B.2.18.2)
2	Не удовлетворяет техническим условиям (см. B.2.18.3)
3	Функциональная проверка (см. B.2.18.4)
4	Отказ (см. B.2.18.5)
5—255	Зарезервировано

B.2.18.2 Требуется техническое обслуживание

Хотя PD допустимы, внутренняя диагностика указывает, что устройство близко к потере способности правильного функционирования.

Пример — Загрязнение оптических линз, формирование отложений, низкий уровень смазочного вещества.

B.2.18.3 Не удовлетворяет техническим условиям

Хотя PD допустимы, внутренняя диагностика показывает, что Устройство функционирует за пределами установленного диапазона измерений или климатических условий.

Пример — Электропитание, вспомогательная энергия, температура, пневматическое давление, воздействие магнитного поля, вибрации, ускорения, световые помехи, образование пузырьков в жидкостях.

B.2.18.4 Функциональная проверка

PD временно недопустимы из-за намеренных манипуляций на Устройстве.

Пример — Калибровка, настройка в режиме обучения, регулировка положения, моделирование.

B.2.18.5 Отказ

PD недопустимы из-за нарушения функционирования устройства или его периферийных устройств. Устройство не способно выполнять его целевую функцию.

B.2.19 Параметр DetailedDeviceStatus (Подробное состояние Устройства)

Параметр DetailedDeviceStatus будет предоставлять информацию о текущих Событиях, ожидающих обработки, на Устройстве. События ТИПА «Ошибка» или «Предупреждение» и РЕЖИМ «Событие появляется» (см. A.6.4) будут вводиться в список Подробного состояния Устройства с описателем События EventQualifier и кодом События EventCode. При появлении События с РЕЖИМОМ «Event disappears» (Событие исчезает) соответствующий код Подробного состояния Устройства будет установлен в Описатель События «0x00» и код события «0x0000». Таким образом, Данный параметр всегда представляет текущее диагностическое состояние Устройства. Параметр является объектом данных только для чтения. Тип данных: ArrayT с максимальным числом элементов массива (входов Событий) 64. Число элементов массива данного параметра зависит от Устройства. При выключении питания или сбросе Устройства в исходное состояние содержимое всех элементов массива переходит к начальным установкам — Описателю События «0x00» и Коду События «0x0000». Данный параметр необязательный.

В таблице B.14 определена структура параметра DetailedDeviceStatus.

Таблица B.14 — Параметр DetailedDeviceStatus (Индекс 0x0025)

Субиндекс	Имя объекта	Тип данных	Комментарий
1	Error_Warning_1	3 октета	Все октеты 0x00: Нет Ошибок и Предупреждений Октет 1: Определитель События Октеты 2, 3: Код События
2	Error_Warning_2	3 октета	
3	Error_Warning_3	3 октета	
4	Error_Warning_4	3 октета	
…			
n	Error_Warning_n	3 октета	

Проектировщик может выбирать реализацию в виде статического списка (одна фиксированная позиция для каждого События с конкретным Кодом События) или динамического списка (каждый вход События хранится в следующей свободной позиции массива). Доступ по Субиндексу не разрешается для динамических списков.

B.2.20 Параметр ProcessDataInput (Ввод PD)

Параметр ProcessDataInput будет предоставлять последние допустимые входные PD из приложения Устройства. Тип данных и структура данного параметра идентичны входным Данным процесса в канале связи PD. Параметр является объектом данных только для чтения. Данный параметр необязательный.

B.2.21 Параметр ProcessDataOutput (Выход PD)

Параметр ProcessDataOutput будет предоставлять последние допустимые выходные PD, записанные в приложение Устройства. Тип данных и структура данного параметра идентичны выходным Данным процесса в канале связи PD. Параметр является объектом данных только для чтения. Данный параметр необязательный.

B.2.22 Параметр OffsetTime (Время сдвига)

Параметр OffsetTime позволяет приложению Устройства синхронизировать циклы M-последовательностей DL через регулируемые времена сдвига. Тип данных: RecordT. Доступ возможен только через Субиндекс 0. Параметр является объектом данных чтения/записи. Данный параметр необязательный.

Структура параметра показана на рисунке B.7.

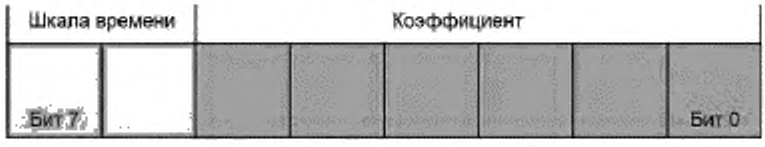


Рисунок B.7 — Структура параметра OffsetTime

Биты от 0 до 5: Multiplier (Коэффициент)

Данные биты содержат 6-битовый коэффициент для вычисления Времени сдвига. Допустимые значения данного коэффициента — от 0 до 63.

Биты от 6 до 7: TimeBase (Шкала времени)

Данные биты устанавливают шкалу времени для вычисления OffsetTime.

Допустимые комбинации для Шкалы времени и Коэффициента приведены в таблице В.15 вместе с результирующими значениями для Времени сдвига. Установка в ноль значений Коэффициента и Шкалы времени деактивирует синхронизацию за счет Времени сдвига. Значение OffsetTime не должно превышать MasterCycleTime (см. В.1.3).

Таблица В.15 — Кодирование Шкалы времени и значения Времени сдвига

Код шкалы времени	Значение шкалы времени	Вычисление	Время сдвига
00	0,01 мс	Multiplier × TimeBase	От 0,01 до 0,63 мс
01	0,04 мс	0,64 мс + Multiplier × TimeBase	От 0,64 до 3,16 мс
10	0,64 мс	3,20 мс + Multiplier × TimeBase	От 3,20 до 43,52 мс
11	2,56 мс	44,16 мс + Multiplier × TimeBase	От 44,16 до 126,08 мс

В.2.23 Параметры профиля

Индексы с 0x0031 по 0x003F зарезервированы для профиля Устройства.

Примечание — Подробная информация приведена в [7].

В.2.24 Предпочтительный Индекс

Предпочтительные Индексы (от 0x0040 до 0x00FE) могут применяться для определяемых изготовителем функций Устройства. Данный диапазон индексов считается предпочтительным из-за меньших протокольных накладных расходов в ISDU и, следовательно, большей пропускной способности для маленьких объектов данных по сравнению с расширенным индексом Extended Index (см. В.2.25).

В.2.25 Расширенный Индекс

Расширенные Индексы (от 0x0100 до 0x3FFF) могут применяться для определенных изготовителем функций Устройства.

В.2.26 Специфический для профиля Индекс (зарезервировано)

Индексы с 0x4000 по 0x4FFF зарезервированы для профиля Устройства.

Примечание — Подробная информация приведена в [7].

Приложение С
(обязательное)

Типы ошибок (ошибки ISDU)

С.1 Общие положения

Тип ошибки применяется с отрицательным подтверждением услуги ISDU (см. А.5.2 и таблицу А.13). Он указывает причину отрицательного подтверждения услуги Read (Чтение) или Write (Запись). Источник ошибки может находиться в Ведущем узле (локальный) или в Устройстве (удаленный).

Тип ошибки ErrorType состоит из двух октетов, основной причины ошибки и более конкретной информации:

- кода ошибки ErrorCode (старший октет);
- дополнительного кода AdditionalCode (младший октет).

Тип ошибки ErrorType представляет информацию об особой ситуации, источнике и экземпляре. Допустимые значения ErrorTypes и критерии их развертывания приведены в подразделах С.2 и С.3. Другие значения типов ошибки зарезервированы и не будут применяться.

С.2 Типы ошибок, связанные с приложением

С.2.1 Обзор

Допустимые типы ошибок, обусловленные приложением Устройства, приведены в таблице С.1.

Таблица С.1 — Типы ошибок

Особая ситуация	Код ошибки	Дополнительный код	Наименование	Определение
Ошибка приложения Устройства — без уточнения	0x80	0x00	APP_DEV	См. С.2.2
Индекс недоступен	0x80	0x11	IDX_N OTAVAIL	См. С.2.3
Субиндекс недоступен	0x80	0x12	SUBIDX_NOTAVAIL	См. С.2.4
Услуга временно недоступна	0x80	0x20	SERV_N OTAVAIL	См. С.2.5
Услуга временно недоступна — локальное управление	0x80	0x21	SERV_NOTAVAIL_LOCCTRL	См. С.2.6
Услуга временно недоступна — управление Устройством	0x80	0x22	SERV_NOTAVAIL_DEVCTRL	См. С.2.7
В доступе отказано	0x80	0x23	IDX_NOT_WRITEABLE	См. С.2.8
Значение параметра вне диапазона	0x80	0x30	PAR_VALOUTOFRNG	См. С.2.9
Значение параметра выше предела	0x80	0x31	PAR_VALGTLI M	См. С.2.10
Значение параметра ниже предела	0x80	0x32	PAR_VALLTLIM	См. С.2.11
Превышение длины параметра	0x80	0x33	VAL_LENOVRRUN	См. С.2.12
Недостаточная длина параметра	0x80	0x34	VAL_LENUNDRUN	См. С.2.13
Функция недоступна	0x80	0x35	FUNC_NOTAVAIL	См. С.2.14
Функция временно недоступна	0x80	0x36	FUNC_UNAVAILTEMP	См. С.2.15
Недопустимый набор параметров	0x80	0x40	PAR_SETINVALID	См. С.2.16
Несогласованный набор параметров	0x80	0x41	PAR_SETINCONSIST	См. С.2.17
Приложение не готово	0x80	0x82	APP_DEVNOTRDY	См. С.2.18
Определяется изготовителем	0x81	0x00	UNSPECIFIC	См. С.2.19
Определяется изготовителем	0x81	0x01 to 0xFF	VENDOR_SPECIFIC	См. С.2.19

C.2.2 Ошибка приложения Устройства — без уточнения

Данный тип ошибки будет применяться, если приложение Устройства отказало в требуемой услуге и подробная информация об особой ситуации недоступна.

C.2.3 Индекс недоступен

Данный тип ошибки будет применяться, когда доступ на чтение или запись осуществляется по несуществующему Индексу.

C.2.4 Индекс недоступен

Данный тип ошибки будет применяться, когда доступ на чтение или запись осуществляется по несуществующему Субиндексу.

C.2.5 Служба временно недоступна

Данный тип ошибки будет применяться, если параметр недоступен для услуги чтения или записи из-за текущего состояния приложения Устройства.

C.2.6 Услуга временно недоступна — локальное управление

Данный тип ошибки будет применяться, если параметр недоступен для услуги чтения или записи из-за происходящей локальной операции в Устройстве (например, операции или параметризации через встроенную панель управления Устройством).

C.2.7 Услуга временно недоступна — управление Устройством

Данный тип ошибки будет применяться, если услуга чтения или записи недоступна из-за удаленно запущенного состояния приложения устройства (например, параметризации вовремя удаленно запущенной операции настройки в режиме обучения или калибровки).

C.2.8 В доступе отказано

Данный тип ошибки будет применяться, если услуга записи пытается получить доступ к параметру только для чтения.

C.2.9 Значение параметра вне диапазона

Данный тип ошибки будет применяться, если делается попытка доступа на чтение к параметру, значение которого находится за пределами его допустимого диапазона значений.

C.2.10 Значение параметра выше предела

Данный тип ошибки будет применяться для услуги записи параметра выше его заданного диапазона значений.

C.2.11 Значение параметра ниже предела

Данный тип ошибки будет применяться для услуги записи параметра ниже его заданного диапазона значений.

C.2.12 Превышение длины параметра

Данный тип ошибки будет применяться, когда содержимое услуги записи параметра больше, чем указанная длина параметра. Такой же код ошибки будет применяться, если объект данных слишком большой для обработки приложением Устройства (например, ограничение буфера ISDU).

C.2.13 Недостаточная длина параметра

Данный тип ошибки будет применяться, когда содержимое услуги записи параметра меньше, чем указанная длина (например, доступ записи значения типа Unsigned16 в параметр типа Unsigned32).

C.2.14 Функция недоступна

Данный тип ошибки будет применяться для услуги записи со значением команды, не поддерживаемой приложением Устройства (например, команда системы с нереализованным значением).

C.2.15 Функция временно недоступна

Данный тип ошибки будет применяться для услуги записи со значением команды, вызывающим функцию Устройства, недоступную из-за текущего состояния приложения Устройства (например, команда системы).

C.2.16 Недопустимый набор параметров

Данный тип ошибки будет применяться, если значения, посланные через передачу одиночного параметра, не согласуются с другими фактическими наборами параметров (например, перекрывающиеся установки для двоичных данных, см. 10.3.4).

C.2.17 Несогласованный набор параметров

Данный тип ошибки будет применяться при завершении передачи блочного параметра с командой ParamDownloadEnd или ParamDownloadStore, если проверка правдоподобия показывает несогласованности (см. 10.3.5 и B.2.2).

C.2.18 Приложение не готово

Данный тип ошибки будет применяться, если услуга чтения или записи отвергнута из-за временно недоступного приложения (например, контроллеры периферийного оборудования во время запуска).

C.2.19 Определяется изготовителем

Данный тип ошибки будет передаваться прямо обрабатывающим элементам более высокого уровня, как ошибка (не предупреждение) Ведущим узлом.

С.3 Производные типы ошибки

С.3.1 Обзор

Производные типы ошибки генерируются на прикладном уровне Ведущего узла и вызываются внутренними особыми ситуациями или ситуациями, полученными из Устройства. В таблице С.2 приведены установленные производные типы ошибок.

Таблица С.2 — Производные типы ошибок

Особая ситуация	Код ошибки	Дополнительный код	Наименование	Определение
Ведущий узел — ошибка связи	0x10	0x00	COM_ERR	См. С.3.2
Ведущий узел — Превышение лимита времени ISDU	0x11	0x00	I-SERVICE_TIMEOUT	См. С.3.3
Событие Устройства — ошибка ISDU (DL, Error, single shot, 0x5600)	0x11	0x00	I-SERVICE_TIMEOUT	См. С.3.4
Событие Устройства — недопустимый сервисный примитив (AL, Error, single shot, 0x5800)	0x11	0x00	I-SERVICE_TIMEOUT	См. С.3.5
Ведущий узел — ошибка контрольной суммы ISDU	0x56	0x00	M_ ISDU_CHECKSUM	См. С.3.6
Ведущий узел — недопустимый сервисный примитив ISDU	0x57	0x00	M_ ISDU_ILLEGAL	См. С.3.7
Событие Устройства — переполнение буфера ISDU (DL, Error, single shot, 0x5200)	0x80	0x33	VAL_LENOVRRUN	См. С.3.8 и С.2.12 События из устаревших устройств будут перенаправляться в режим совместимости с этим производным типом ошибки

С.3.2 Ведущий узел — ошибка связи

Ведущий узел генерирует отрицательный ответ на услугу с данным типом ошибки, если ошибка связи произошла во время услуги чтения или записи, например, соединение SDCI прервано.

С.3.3 Ведущий узел — превышение лимита времени ISDU

Ведущий узел генерирует отрицательный ответ на услугу с данным типом ошибки, если услуга Read или Write выполняется дольше, чем установленный I-Service лимит времени (см. таблицу 97) в Ведущем узле.

С.3.4 Событие Устройства — ошибка ISDU

Если Ведущий узел получил событие с описателем события (см. A.6.4): DL, Error, Event single shot и кодом события 0x5600, то генерируется и возвращается инициатору запроса отрицательный ответ на услугу (см. С.3.3).

С.3.5 Событие Устройства — недопустимый сервисный примитив ISDU

Если Ведущий узел получил событие с описателем события (см. A.6.4): AL, Error, Event single shot и кодом события 0x5600, то генерируется и возвращается инициатору запроса отрицательный ответ на услугу (см. С.3.3).

С.3.6 Ведущий узел — ошибка контрольной суммы ISDU

Ведущий узел генерирует отрицательный ответ на услугу с данным типом ошибки, если его DL обнаруживает ошибку контрольной суммы ISDU.

С.3.7 Ведущий узел — недопустимый сервисный примитив ISDU

Ведущий узел генерирует отрицательный ответ на услугу с данным типом ошибки, если его DL обнаруживает недопустимый сервисный примитив ISDU.

С.3.8 Событие Устройства — переполнение буфера ISDU

Если Ведущий узел получил событие с описателем события (см. A.6.4): DL, Error, Event single shot и кодом события 0x5200, то генерируется и возвращается инициатору запроса отрицательный ответ на услугу, указывающий превышение длины параметра (см. С.2.12).

Приложение D
(обязательное)

Коды ошибок (диагностическая информация)

D.1 Общие положения

Концепция Событий описана в 7.3.8.1, а общая структура и кодирование событий установлены в разделе А.6. Всякий раз, когда код состояния указывает Событие в случае особой ситуации в Устройстве или Ведущем узле, будет предоставлен соответствующий код события в качестве диагностической информации. Как установлено в разделе А.6, запись События содержит код события EventCode в дополнение к описателю события EventQualifier. Код события EventCode определяет действительную особую ситуацию. Допустимые значения кода события приведены в таблице D.1, все другие значения кода событий зарезервированы и не будут применяться.

D.2 Коды событий для Устройств

В таблице D.1 приведены используемые идентификаторы кодов событий и их определения. Коды событий созданы приложением специальных технических Устройств (экземпляр = APP).

Таблица D.1 — Коды событий

Код События	Определение и рекомендуемые действия по технической поддержке	Значение состояния Устройства (примечание 1)	ТИП (примечание 2)
0x0000	Нет нарушений функционирования	0	Уведомление
0x1000	Общее нарушение функционирования — неизвестная ошибка	4	Ошибка
От 0x1001 до 0x17FF	Зарезервировано		
От 0x1800 до 0x18FF	Определяется изготовителем		
От 0x1900 до 0x3FFF	Зарезервировано		
0x4000	Отказ по температуре — Перегрузка	4	Ошибка
От 0x4001 до 0x420F	Зарезервировано		
0x4210	Превышение температуры Устройства — Очистить источник тепла	2	Предупреждение
От 0x4211 до 0x421F	Зарезервировано		
0x4220	Недобор температуры устройства — Изолировать устройство	2	Предупреждение
От 0x4221 до 0x4FFF	Зарезервировано		
0x5000	Отказ оборудования Устройства — Заменить Устройство	4	Ошибка
От 0x5001 до 0x500F	Зарезервировано		
0x5010	Нарушение функционирования компонента — Отремонтировать или заменить	4	Ошибка

Продолжение таблицы D.1

Код События	Определение и рекомендуемые действия по технической поддержке	Значение состояния Устройства (примечание 1)	ТИП (примечание 2)
0x5011	Потеря энергонезависимой памяти — Проверить батареи	4	Ошибка
0x5012	Батареи разряжены — Заменить батареи	2	Предупреждение
От 0x5013 до 0x50FF	Зарезервировано		
0x5100	Общий отказ питания — Проверить наличие	4	Ошибка
0x5101	Предохранитель перегорел или разомкнут — Заменить предохранитель	4	Ошибка
От 0x5102 до 0x510F	Зарезервировано		
0x5110	Превышение напряжения первичного источника питания — Проверить допуски	2	Предупреждение
0x5111	Недостаточное напряжение первичного источника питания — Проверить допуски	2	Предупреждение
0x5112	Отказ вторичного источника питания (порт класса B) — Проверить допуски	2	Предупреждение
От 0x5113 до 0x5FFF	Зарезервировано		
0x6000	Отказ программного обеспечения Устройство — Проверить версию встроенного программного обеспечения	4	Ошибка
От 0x6001 до 0x631F	Зарезервировано		
0x6320	Ошибка параметра — Проверить данные и значения	4	Ошибка
0x6321	Параметр отсутствует — Проверить данные	4	Ошибка
От 0x6322 до 0x634F	Зарезервировано		
0x6350	Параметр изменен — Проверить конфигурацию	4	Ошибка
От 0x6351 до 0x76FF	Зарезервировано		
0x7700	Обрыв проводов на подчиненном устройстве — Проверить установку.	4	Ошибка
От 0x7701 до 0x770F	Обрыв проводов на подчиненных устройствах 1 … 15 — Проверить установку	4	Ошибка
0x7710	Короткое замыкание — Проверить установку	4	Ошибка
0x7711	Плохое заземление — Проверить установку	4	Ошибка
От 0x7712 до 0x8BFF	Зарезервировано		
0x8C00	Отказ специального технического приложения — Сбросить Устройство в исходное состояние	4	Ошибка
0x8C01	Осуществляется моделирование — Проверить рабочий режим	3	Предупреждение

Окончание таблицы D.1

Код События	Определение и рекомендуемые действия по технической поддержке	Значение состояния Устройства (примечание 1)	ТИП (примечание 2)
От 0x8C02 до 0x8C0F	Зарезервировано		
0x8C10	Превышение диапазона переменной процесса — Неопределенные PD	2	Предупреждение
От 0x8C11 до 0x8C1F	Зарезервировано		
0x8C20	Превышение диапазона измерений — Проверить приложение	4	Ошибка
От 0x8C21 до 0x8C2F	Зарезервировано		
0x8C30	Превышение диапазона переменной процесса — Неопределенные PD	2	Предупреждение
От 0x8C31 до 0x8C3F	Зарезервировано		
0x8C40	Требуется техническое обслуживание — Очистка	1	Уведомление
0x8C41	Требуется техническое обслуживание — Заправка	1	Уведомление
0x8C42	Требуется техническое обслуживание — Замена быстроизнашивающихся деталей и расходных материалов	1	Уведомление
От 0x8C43 до 0x8C9F	Зарезервировано		
От 0x8CA0 до 0x8DFF	Определяется изготовителем		
От 0x8E00 до 0xAFFF	Зарезервировано		
От 0xB000 до 0xBFFF	Зарезервировано для профиля		
От 0xC000 до 0xFEFF	Зарезервировано		
От 0xFF00 до 0xFFFF	Коды Событий, специфических для SDCI (см. таблицу D.2)		
Примечание 1 — См. B.2.18. Примечание 2 — См. таблицу A.19.			

В таблице D.2 перечисляются базовые События SDCI, связанные с SM, приложениями Устройства и Ведущего узла, и устанавливается, как они кодируются. Может сообщаться о других типах Событий, но в настоящем стандарте они не устанавливаются. Обработка данных Событий Ведущим узлом определяется изготовителем.

Таблица D.2 — Коды базовых Событий SDCI

Особая ситуация[a)]	Источник	Экземпляр	Наименование	Код События	Действие	Примечание
Модуль управления системой						
Индикация режима	LOCAL	DL	NEW_SLAVE	0xFF21	Остановка PD	См. раздел 11

Окончание таблицы D.2

Особая ситуация[a)]	Источник	Экземпляр	Наименование	Код События	Действие	Примечание
Потеря связи Устройства	LOCAL	APP	DEV_COM_LOST	0xFF22	—	См. раздел 11
Несоответствие определений DS	LOCAL	APP	DS_IDENT_ MISMATCH	0xFF23	—	См. раздел 11
Переполнение буфера DS	LOCAL	APP	DS_BUFFER _OVER-FLOW	0xFF24	—	См. раздел 11
Отказ в доступе к параметрам DS	LOCAL	APP	DS_ACCESS_DENIED	0xFF25	—	См. раздел 11
Неопределенные						
Неправильная сигнализация о Событиях	LOCAL	DL	EVENT	0xFF31	Индикация Событий	См. раздел 11
Специальные приложения Устройства						
Запрос подкачки в DS	REMOTE	APP	DS_UPLOAD _REQ	0xFF91	Индикация Событий	
Зарезервировано	REMOTE	APP		0xFF98	Индикация Событий	Не должны применяться

[a)] Все События имеют код состояния типа 2 (с деталями), тип EventQualifier — «Уведомление», режим EventQualifier — «Однократный».

Приложение Е
(обязательное)

Типы данных

Е.1 Общие положения

В данном приложении устанавливаются основные и составные типы данных. Структура и аспекты передачи типов данных демонстрируются на примерах для единичного использования или использования в упакованном виде.

Примечание — Большое число примеров содержится в [6].

Е.2 Основные типы данных

Е.2.1 Общие положения

Кодирование основных типов данных показано для единичного использования, которое характеризуется следующим:

- PD состоят из одного основного типа данных;
- параметр состоит из одного основного типа данных;
- доступ по Субиндексу (> 0) к отдельным элементам данных параметров из составных типов данных (массивы, записи).

Е.2.2 Тип данных BooleanT

Тип данных BooleanT представляет тип данных, который может иметь только два различных значения, а именно TRUE и FALSE. Тип данных определяется в таблице Е.1. Для единичного использования кодирование показано в таблице Е.2. Отправитель всегда применяет 0xFF для 'TRUE' или 0x00 для 'FALSE'. Получатель может интерпретировать диапазон от 0x01 до 0xFF для 'TRUE' и интерпретировать 0x00 для 'FALSE' для упрощения реализаций. Упакованная форма демонстрируется в таблице Е.22 и на рисунке Е.8.

Таблица Е.1 — Тип данных BooleanT

Имя типа данных	Диапазон значений	Разрешение	Длина
BooleanT	TRUE/FALSE	—	1 бит или 1 октет

Таблица Е.2 — Кодирование типа данных BooleanT

Бит	7	6	5	4	3	2	1	0	Значения
TRUE	1	1	1	1	1	1	1	1	0xFF
FALSE	0	0	0	0	0	0	0	0	0x00

Е.2.3 Тип данных UIntegerT

Тип данных UIntegerT представляет целое число без знака, занимающее от 2 до 64 битов. Число размещено и выровнено по правому краю в следующих допустимых октетных контейнерах: 1, 2, 4 или 8. Биты заполнения высокого порядка заполняются значением «0». Примеры кодирования показаны на рисунке Е.1.

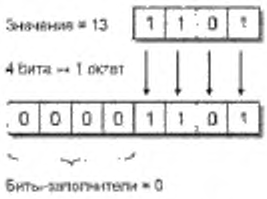


Рисунок Е.1, лист 1 — Примеры кодирования типа данных UIntegerT

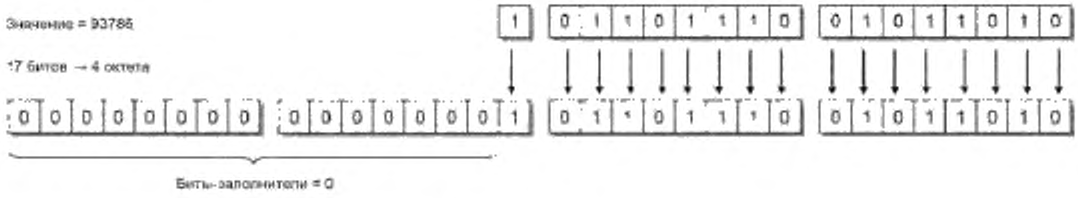


Рисунок Е.1, лист 2

Тип данных UIntegerT определяется в таблице Е.3 для единичного использования.

Таблица Е.3 — Тип данных UIntegerT

Имя типа данных	Диапазон значений	Разрешение	Длина
UIntegerT	$0 \ldots 2^{\text{ДлинаВБитах}-1}$	1	1 октет, или 2 октета, или 4 октета, или 8 октетов
Примечание 1 — Биты заполнения высокого порядка заполняются значением «0». Примечание 2 — Самый значащий (старший) октет посылается первым.			

Е.2.4 Тип данных IntegerT

Тип данных UIntegerT представляет целое число со знаком, занимающее от 2 до 64 битов. Число размещено в следующих допустимых октетных контейнерах: 1, 2, 4 или 8 октетов, выровнено в правую сторону и расширено в соответствии со знаком на выбранное число битов. Тип данных определяется в таблице Е.4 для единичного использования. SN представляет знаковый бит с «0» для всех положительных чисел и нуля и «1» — для всех отрицательных чисел. Биты заполнения наполняются содержимым знакового бита (SN).

Таблица Е.4 — Тип данных IntegerT

Имя типа данных	Диапазон значений	Разрешение	Длина
IntegerT	$-2^{\text{ДлинаВБитах}-1} \ldots 2^{\text{ДлинаВБитах}-1}$	1	1 октет, или 2 октета, или 4 октета, или 8 октетов
Примечание 1 — Биты заполнения наполняются значением знакового бита (SN). Примечание 2 — Самый значащий (старший) октет посылается первым (наименьший относительный номер октета в таблице Е.5).			

Четыре возможности кодирования в контейнерах приведены в таблицах Е.5—Е.8.

Таблица Е.5 — Кодирование типа данных IntegerT (8 октетов)

Бит	7	6	5	4	3	2	1	0	Контейнер
Октет 1	SN	2^{62}	2^{61}	2^{60}	2^{59}	2^{58}	2^{57}	2^{56}	8 октетов
Октет 2	2^{55}	2^{54}	2^{53}	2^{52}	2^{51}	2^{50}	2^{49}	2^{48}	
Октет 3	2^{47}	2^{46}	2^{45}	2^{44}	2^{43}	2^{42}	2^{41}	2^{40}	
Октет 4	2^{39}	2^{38}	2^{37}	2^{36}	2^{35}	2^{34}	2^{33}	2^{32}	
Октет 5	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	
Октет 6	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Октет 7	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Октет 8	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	

Таблица Е.6 — Кодирование типа данных IntegerT (4 октета)

Бит	7	6	5	4	3	2	1	0	Контейнер
Октет 1	SN	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	4 октета
Октет 2	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Октет 3	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Октет 4	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	

Таблица Е.7 — Кодирование типа данных IntegerT (2 октета)

Бит	7	6	5	4	3	2	1	0	Контейнер
Октет 1	SN	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2 октета
Octet 2	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	

Таблица Е.8 — Кодирование типа данных IntegerT (1 октет)

Бит	7	6	5	4	3	2	1	0	Контейнер
Октет 1	SN	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	1 октет

Примеры кодирования в контейнерах показаны на рисунке Е.2.

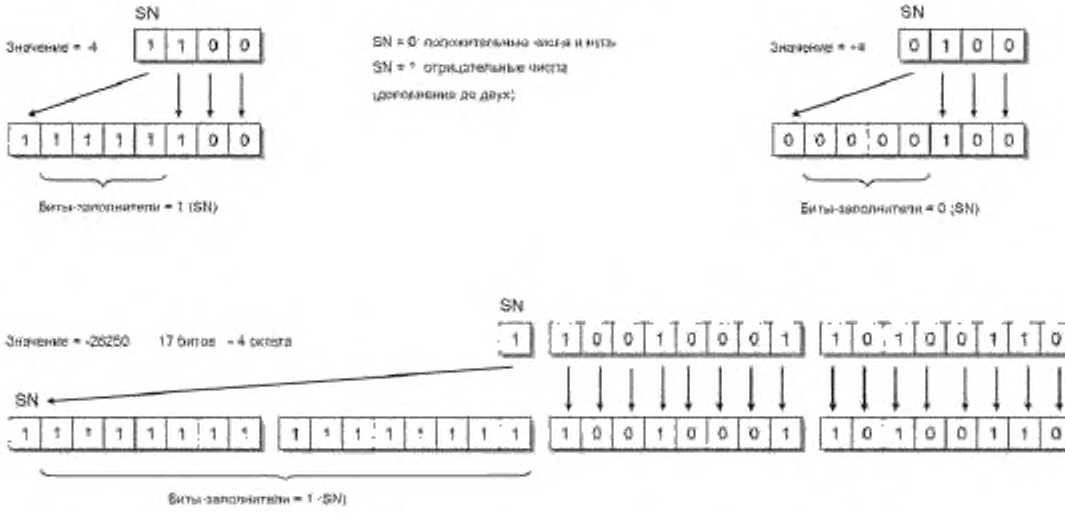


Рисунок Е.2 — Примеры кодирования типа данных IntegerT

Е.2.5 Тип данных Float32T

Тип данных Float32T представляет число, установленное стандартом IEEE 754—2008, число с одинарной точностью (32 бита). В таблице Е.9 дается определение, а в таблице Е.10 — кодирование. SN представляет знак с «0» для всех положительных чисел и нуля и «1» для всех отрицательных чисел.

Таблица Е.9 — Тип данных Float32T

Имя типа данных	Диапазон значений	Разрешение	Длина
Float32T	См. IEEE 754—2008	См. IEEE 754—2008	4 октета

Таблица Е.10 — Кодирование типа данных Float32T

Биты	7	6	5	4	3	2	1	0
Октет 1	SN	Экспонента (E)						
	2^0	2^7	2^6	2^5	2^4	2^3	2^2	2^1
Октет 2	(E)	Дробь (F)						
	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
Октет 3	Дробь (F)							
	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}
Октет 4	Дробь (F)							
	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}	2^{-21}	2^{-22}	2^{-23}

Для реализации отрицательных значений экспоненты применяется специальный механизм кодирования экспоненты, работающий следующим образом.

Экспонента (E) типа данных Float32T кодируется, применяя смещенное двоичное представление с нулевым смещением, равным 127, известным также как смещение экспоненты в стандарте IEEE 754—2008.

Emin = 0x01 – 0x7F = –126

Emax = 0xFE – 0x7F = 127

Смещение экспоненты = 0x7F = 127

Таким образом, как определено смещенным двоичным представлением, чтобы получить действительное значение экспоненты, надо вычесть 127 из хранящегося значения экспоненты.

E.2.6 Тип данных StringT

Тип данных StringT представляет упорядоченную последовательность символов (знаков) с переменной или постоянной длиной в октетах (максимум 232 октета), закодированных в US-ASCII (7 бит) или UTF-8. UTF-8 применяет один октет для всех символов ASCII и до четырех октетов для всех других символов. Значение 0x00 недопустимо применять в качестве символа. В таблице Е.11 дается определение типа данных.

Таблица Е.11 — Тип данных StringT

Имя типа данных	Кодировка	Стандарты	Длина
StringT	US-ASCII	См. ИСО/МЭК 646	Любая длина строки символов с максимум 232 октетами
	UTF-8	См. ИСО/МЭК 10646	
Примечание — Длина может быть определена из описания устройства IODD Устройства через атрибут 'fixedLength' (фиксированнаяДлина).			

Экземпляр типа данных StringT может быть короче, чем определено атрибутом IODD 'fixedLength'. Значение 0x00 будет применяться для заполнения неиспользованных октетов. Строки символов могут передаваться с их фактической длиной в случае отдельного использования (см. рисунок Е.3). Возможна оптимизация при передаче путем опущения октетов заполнения, если не установлен атрибут IODD 'fixedLengthRestriction' (Ограничение фиксированной длины). Получатель может определить исходную длину из длины ISDU или путем нахождения первого символа NULL (0x00) (см. А.5.2 и А.5.3).

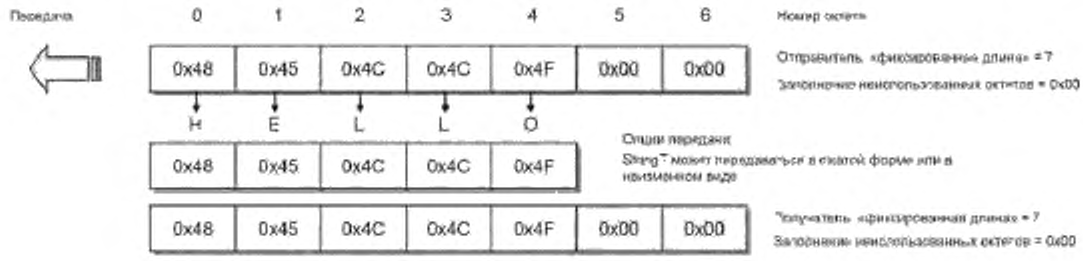


Рисунок Е.3 — Одиночный доступ к типу данных StringT

E.2.7 Тип данных OctetStringT

Тип данных OctetStringT представляет упорядоченную последовательность октетов с фиксированной длиной (максимум 232 октета). В таблице E.12 дается определение, а на рисунке E.4 приводится пример кодирования для фиксированной длины 7.

Таблица E.12 — Тип данных OctetStringT

Имя типа данных	Диапазон значений	Стандарты	Длина
OctetStringT	0x00 ... 0xFF в октете	—	Фиксированная длина с максимум 232 октетами
Примечание — Длина может быть определена из описания устройства IODD Устройства через атрибут 'fixedLength' (фиксированная длина).			

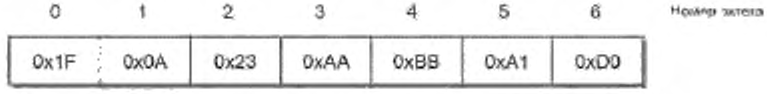


Рисунок E.4 — Пример кодирования типа данных OctetStringT

E.2.8 Тип данных TimeT

Тип данных TimeT базируется на стандарте RFC 5905 и составлен из двух значений без знака, которые выражают сетевое время, относящееся к конкретной дате. Его семантика отличается от стандарта RFC 5905, как показано на рисунке E.5. В таблице E.13 дается определение, а в таблице E.14 — пример кодирования типа данных TimeT.

Первым элементом является 32-битовое целое без знака, которое предоставляет сетевое время в секундах 1900-01-01 0.00,00 (ВКВ) или с 2036-02-07 6.28,16(ВКВ) для значений времени меньших, чем 0x9DFF4400, что представляет 1984-01-01 0:00,00 (ВКВ). Возобновление через 136 лет не является автоматически обнаруживаемым и должно поддерживаться приложением. Второй элемент является 32-битовым целым без знака, представляющим дробную часть секунд в $1/2^{32}$ с.

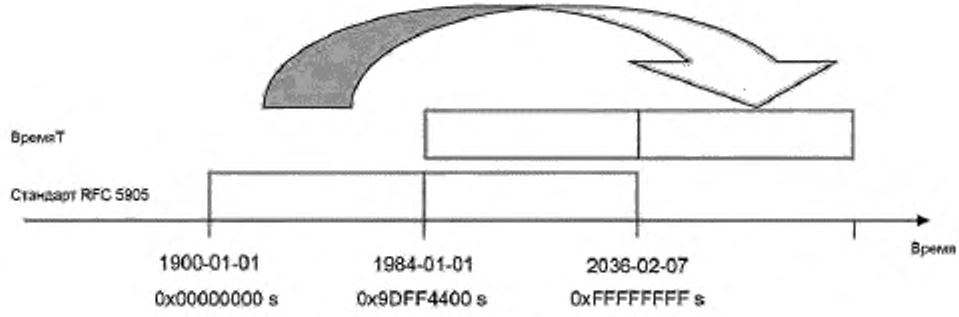


Рисунок E.5 — Определение типа данных TimeT

Таблица E.13 — Тип данных TimeT

Имя типа данных	Диапазон значений	Разрешение	Длина
TimeT	Октеты с 1 по 4 (см. таблицу E.14): $0 \le i \le (2^{32}-1)$	с (секунды)	8 октетов (32-битовое целое без знака + 32-битовое целое без знака)
	Октеты с 5 по 8 (см. таблицу E.14): $0 \le i \le (1/2^{32}-1)$	$(1/2^{32})$ с	
Примечание — 32-битовое целое без знака является обычным типом данных в вычислительной технике.			

Таблица Е.14 — Кодирование типа данных TimeT

Бит	7	6	5	4	3	2	1	0	Определения
Октет 1	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	Время в секундах с 1900-01-01 0.00,00 или с 2036-02-07 6.28,16, когда значение времени меньше, чем 0x9DFF4400.00000000
Октет 2	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Октет 3	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Октет 4	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	
Октет 5	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	Дробная часть секунды. Единица приращения $1/(2^{32})$ с
Октет 6	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Октет 7	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Октет 8	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	
	MSB							LSB	MSB = старший бит LSB = младший бит

Е.2.9 Тип данных TimeSpanT

Тип данных TimeSpanT — это 64-битовое значение дополнения до двухдвоичного числа длиной восемь октетов, представляющее разность в сетевом времени в долях секунды, равных 1/232 секунды. В таблице Е.15 дается определение типа данных TimeSpanT, а в таблице Е.16 — его кодирование.

Таблица Е.15 — Тип данных TimeSpanT

Имя типа данных	Диапазон значений	Разрешение	Длина
TimeSpanT	Октеты с 1 по 8 (см. таблицу Е.16): - $2^{63} \le i \le (2^{63}-1)$	$(1/2^{32})$ с	8 октетов (64-битовое целое)
Примечание — 64-битовое целое является обычным типом данных в вычислительной технике.			

Таблица Е.16 — Кодирование типа данных TimeSpanT

Бит	7	6	5	4	3	2	1	0	Определения
Октет 1	2^{63}	2^{62}	2^{61}	2^{60}	2^{59}	2^{58}	2^{57}	2^{56}	Дробная часть секунды как 64-битовое целое. Единица приращения равна $1/(2^{32})$ с
Октет 2	255	2^{54}	2^{53}	2^{52}	2^{51}	2^{50}	2^{49}	2^{48}	
Октет 3	2^{47}	2^{46}	2^{45}	2^{44}	2^{43}	2^{42}	2^{41}	2^{40}	
Октет 4	2^{39}	2^{38}	2^{37}	2^{36}	2^{35}	2^{34}	2^{33}	2^{32}	
Октет 5	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	
Октет 6	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Октет 7	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Октет 8	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	
	MSB							LSB	MSB = старший бит LSB = младший бит

Е.3 Составные типы данных

Е.3.1 Общие положения

Составные типы данных состоят только из комбинаций основных типов данных. Составной тип данных состоит из нескольких основных типов данных, объединенных в последовательности октетов. Неиспользованное пространство будет заполняться значениями «0».

E.3.2 Тип данных ArrayT

Тип данных ArrayT, адресуемый Индексом, представляет собой структуру данных с элементами данных одного и того же типа данных. Отдельные элементы данных адресуются Субиндексом. Субиндекс 0 адресует весь массив в пространстве Индекса. Правила структурирования для массивов приведены в таблице E.17.

Таблица E.17 — Правила структурирования для типа данных ArrayT

Номер правила	Спецификация правила
1	Элементы данных Субиндекса упакованы в строку без пробелов, описывающую последовательность октетов
2	Элемент данных с наибольшим Субиндексом *n* располагается выровненным вправо в последовательности октетов
3	Типы данных UIntegerT и IntegerT с длиной ≥ 58 битов < 64 не разрешены

В таблице E.18 и на рисунке E.9 приведены примеры доступа к массиву. Его содержимым является набор параметров одного и того же основного типа данных.

Таблица E.18 — Пример доступа к типу данных ArrayT

Индекс	Субиндекс	Смещение	Элементы данных	Тип данных
66	1	12	0x2	IntegerT, 'bitLength' (длина в битах) = 3
	2	9	0x6	
	3	6	0x4	
	4	3	0x7	
	5	0	0x5	

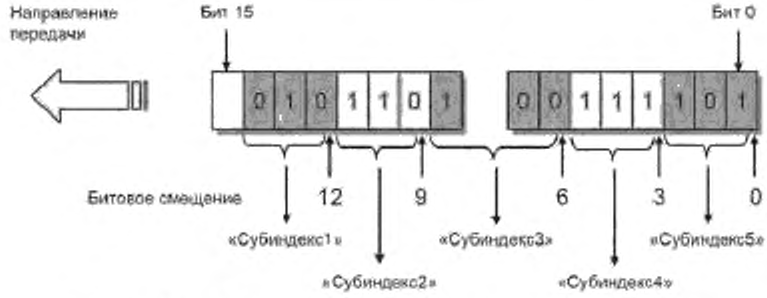


Рисунок E.6 — Пример структуры данных для типа данных ArrayT

E.3.3 Тип данных RecordT

Запись, адресуемая Индексом, представляет собой структуру данных с элементами данных различных типов данных. Субиндекс разрешает адресацию отдельных элементов данных в записи в определенных позициях.

Примечание — Позиции битов в типе данных могут быть получены из описания данных IODD конкретного Устройства.

Правила структурирования для записей приведены в таблице E.19.

Таблица E.19 — Правила структурирования для типа данных RecordT

Номер правила	Спецификация правила
1	Субиндексы в описании данных IODD будут перечисляться в порядке возрастания от 1 до *n*, описывая последовательность октетов. Пробелы в списке Субиндексов не разрешены

Окончание таблицы Е.19

Номер правила	Спецификация правила
2	Смещения битов в последовательности октетов будут всегда указываться (строгий порядок в IODD может отсутствовать)
3	Смещение битов начинается с последнего октета в последовательности. Данный октет может начинаться со смещением 0 для каждого младшего бита и смещением 7 для старшего бита
4	Следующие типы данных всегда будут выравниваться по границам октета: Float32T, StringT, OctetStringT, TimeT и TimeSpanT
5	Типы данных UIntegerT и IntegerT длиной ≥ 58 бит всегда будут выравниваться на одной стороне границы октета
6	Для типов данных UIntegerT и IntegerT с длиной ≥ 8 битов настоятельно рекомендуется всегда производить выравнивание на одной стороне границы октета
7	Для типов данных UIntegerT и IntegerT с длиной < 8 настоятельно рекомендуется не пересекать границы октета
8	Позиция бита не будет применяться более чем одним элементом данных

В таблице 20 дается пример 1 для доступа к типу данных RecordT. Он состоит из изменяющихся параметров Status (Состояние), Text (Испытания) и Value (Значение).

Таблица Е.20 — Пример 1 для доступа к типу данных RecordT

Индекс	Субиндекс	Смещение	Элементы данных							Тип данных	Наименование
47	1	88	0x23	0x45						UIntegerT, 'bitLength' = 16	Состояние
	2	32	H	E	L	L	O	0x00	0x00	StringT, 'fixedLength' = 7	Текст
	3	0	0x56	0x12	0x22	0x34				UIntegerT, 'bitLength' = 32	Значение
Примечание — 'bitLength' и 'fixedLength' определены в описании Устройства IODD конкретного Устройства.											

В таблице 21 дается пример 2 для доступа к типу данных RecordT. Он состоит из изменяющихся параметров Level (Уровень), Min (Мин.) и Max (Макс.). На рисунке Е.7 показана соответствующая структура данных.

Таблица Е.21 — Пример 2 для доступа к типу данных RecordT

Индекс	Субиндекс	Смещение	Элементы данных			Тип данных	Наименование
46	1	2	0x32	0xF1		UIntegerT, 'bitLength' = 14	Уровень
	2	-{}-1	FALSE			BooleanT	Мин.
	3	0	TRUE			BooleanT	Макс.
Примечание — 'bitLength' и 'fixedLength' определены в описании Устройства IODD конкретного Устройства.							

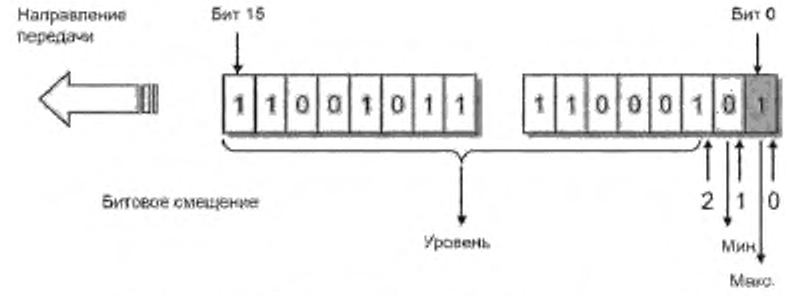


Рисунок Е.7 — Пример 2 структуры типа данных RecordT

В таблице Е.22 приведен пример 3 доступа к типу данных RecordT. Структура состоит из изменяющихся параметров с именами от Control до Enable. На рисунке Е.8 показана соответствующая структура типа данных RecordT из примера 3 со смещениями битов.

Таблица Е.22 — Пример 3 доступа к типу данных RecordT

Индекс	Субиндекс	Смещение	Элементы данных		Тип данных	Наименование
45	1	32	TRUE		BooleanT	NewBit
	2	33	FALSE		BooleanT	DR4
	3	34	FALSE		BooleanT	CR3
	4	35	TRUE		BooleanT	CR2
	5	38	TRUE		BooleanT	Control
	6	16	0xF8	0x23	OctetStringT, 'fixedLength' = 2	Setpoint
	7	8	0x41		StringT, 'fixedLength' = 1	Unit
	8	0	0xC3		OctetStringT, 'fixedLength' = 1	Enable
Примечание — Описатель 'fixedLength' определен в описании Устройства IODD конкретного Устройства.						

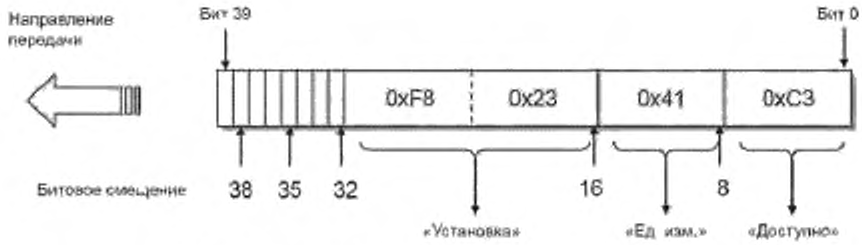


Рисунок Е.8 — Пример 3 структуры типа данных RecordT

На рисунке Е.9 показан запрос выборочной записи переменной в типе записи RecordT из примера 3 и запрос записи целой записи типа данных RecordT (см. А.5.7).

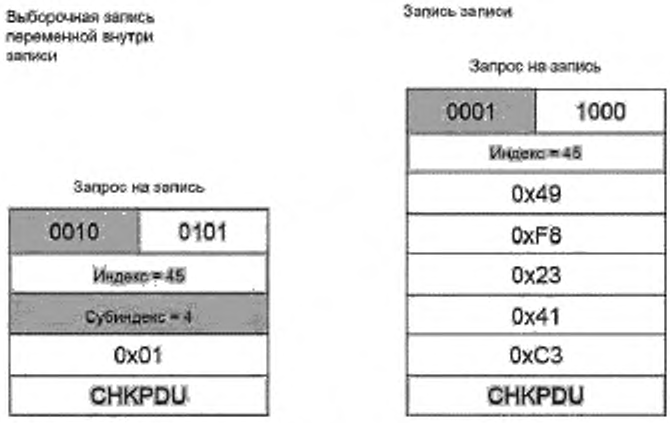


Рисунок Е.9 — Запрос записи для примера 3

Приложение F
(обязательное)

Структура объекта данных Хранилища данных

В таблице F.1 дается структура объекта данных DS на Ведущем узле (см. 11.3.2).

Таблица F.1 — Структура сохраненного объекта данных DS

Часть	Имя параметра	Определение	Тип данных
Объект 1	ISDU_Index	Индекс ISDU (от 0 до 65 535)	Unsigned16
	ISDU_Subindex	Индекс ISDU (от 0 до 255)	Unsigned8
	ISDU_Length	Длина последующей записи	Unsigned8
	ISDU_Data	Запись длины ISDU_Length	Record
Объект 2	ISDU_Index	Индекс ISDU (от 0 до 65 535)	Unsigned16
	ISDU_Subindex	Индекс ISDU (от 0 до 255)	Unsigned8
	ISDU_Length	Длина последующей записи	Unsigned8
	ISDU_Data	Запись длины ISDU_Length	Record
...			
Объект *n*	ISDU_Index	Индекс ISDU (от 0 до 65 535)	Unsigned16
	ISDU_Subindex	Индекс ISDU (от 0 до 255)	Unsigned8
	ISDU_Length	Длина последующей записи	Unsigned8
	ISDU _Data	Запись длины ISDU_Length	Record

Устройство будет вычислять требуемый размер памяти, суммируя объекты от 1 до *n* (см. таблицу B.10, Субиндекс 3).

Ведущий узел будет локально сохранять информацию заголовка, установленную в таблице F.2. См таблицу B.10.

Таблица F.2 — Связанная информация заголовка, сохраненная в объектах данных Хранилища памяти

Часть	Имя параметра	Определение	Тип данных
Заголовок	Контрольная сумма параметров	32-битовая Контрольная сумма или счетчик изменений (см. 10.4.8)	Unsigned32
	VendorID (IDИзготовителя)	См. B.1.8	Unsigned16
	DeviceID (IDУстройства)	См. B.1.9	Unsigned32
	FunctionID (IDФункции)	См. B.1.10	Unsigned16

Приложение G
(обязательное)

Соответствие Ведущего узла и Устройства

G.1 Требования электромагнитной совместимости (ЭМС)

G.1.1 Общие положения

Требования ЭМС настоящего приложения относятся только к части интерфейса SDCI конкретного Ведущего узла или Устройства. Технологические функции Устройства и его соответствующие требования ЭМС находятся вне области применения настоящего стандарта. С этой целью будут применяться специфические производственные стандарты для Устройства. Для Ведущего узла, как правило, требования ЭМС для периферийных устройств устанавливаются в МЭК 61131-2 или МЭК 61000-6-2.

Для обеспечения надлежащих условий эксплуатации интерфейса SDCI тестовые конфигурации, определенные в G.1.6 (Ведущий узел) или G.1.7 (Устройство), будут поддерживаться в течение всех испытаний ЭМС. Тесты, установленные в производственных стандартах на испытываемое оборудование, могут альтернативно выполняться в режиме SIO.

G.1.2 Условия эксплуатации

Рекомендуется оценивать SDCI на стадии запуска со временами цикла, приведенными в таблице G.1. В большинстве случаев это приводит к минимальным временным требованиям для выполнения данных тестов. В качестве альтернативы SDCI может оцениваться во время нормальной эксплуатации Устройства при условии, что требуемое число M-последовательностей, определенных в таблице G.1, осуществлялось во время прохождения каждого теста.

G.1.3 Критерии эффективности

a) Критерий эффективности A

Интерфейс SDCI, работающий на среднем времени цикла, как определено в таблице G.1, не должен показывать более шести обнаруженных ошибок M-последовательностей на числе M-последовательностей, указанном в таблице G.1.

Таблица G.1 — Условия тестирования ЭМС для SDCI

Скорость передачи	Ведущий узел		Устройство		Максимальное число ошибок в M-последовательностях
	t_{CYC}	Число M-последовательностей типа TYPE_2_5 (чтение) (6 октетов)	t_{CYC}	Число M-последовательностей типа TYPE_0 (чтение) (4 октета)	
4,8 кбит/с	18,0 мс	300 (6 000)	100 T_{BIT} (20,84 мс)	350 (7 000)	6
38,4 кбит/с	2,3 мс	450 (9 000)	100 T_{BIT} (2,61 мс)	500 (10 000)	6
230,4 кбит/с	0,4 мс	700 (14 000)	100 T_{BIT} (0,44 мс)	800 (16 000)	6
Примечание — Число M-последовательностей вычисляется в соответствии с алгоритмом H.2 и округляется. Большее число M-последовательностей (в скобках) требуется, если определенный тест (например, испытания на устойчивость к наносекундным импульсным помехам) применяет воздействия только в пакетном режиме (см. таблицу G.2).					

b) Критерий эффективности B

Частота появления ошибок в критерии A должна быть удовлетворительной после, но не во время тестирования. Не допускается никаких изменений реального рабочего режима (например, постоянной потери связи или сохраненных данных).

G.1.4 Требуемые испытания на устойчивость

В таблице G.2 определяются испытания на ЭМС, которые должны быть выполнены.

Таблица G.2 — Уровни испытаний на ЭМС

Явление	Уровень испытаний	Критерий эффективности	Ограничения
Электростатические разряды (ЭСР) МЭК 61000-4-2	Грозовой разряд: ± 8 кВ Контактный разряд: ± 4 кВ	B	См. G.1.4, перечисление a)
Радиочастотное электромагнитное поле. С амплитудной модуляцией МЭК 61000-4-3	80—1000 МГц 10 В/м 1400—2000 МГц 3 В/м 1400—2000 МГц 1 В/м	A	См. G.1.4, перечисление a) и G.1.4, перечисление b)
Наносекундные импульсные помехи МЭК 61000-4-4	± 1 кВ	A	Только 5 кГц. Число M-последовательностей в таблице G.1 будет увеличено в 20 раз из-за отношения импульс/цикл 15 мс/300 мс. См. G.1.4, перечисление c)
	± 2 кВ	B	
Импульс перенапряжения МЭК 61000-4-5	Не требуется для соединений SDCI (длина соединения SDCI ограничена 20 м)		—
Радиочастотные помехи общего вида МЭК 61000-4-6	0,15—80 МГц 10 V электромагнитного поля	A	См. G.1.4, перечисление b) и G.1.4, перечисление d)
Краткосрочные провалы и прерывания напряжения МЭК 61000-4-11	Не требуется для соединений SDCI		

Как определено в таблице G.2, дополнительно устанавливаются следующие требования:

a) Так как это явление также воздействует на все испытываемое устройство, существующие специфические для Устройства производственные стандарты имеют приоритет над остальными уровнями испытаний, приведенными выше.

b) Испытания будут проводиться с размером шага 1 % и задержкой 1 с. Если одиночная M-последовательность происходит на определенной частоте, Данная частота будет испытана до передачи числа M-последовательностей в соответствии с таблицей G.1 или пока не произойдет 6 ошибок в M-последовательностях.

c) Время испытаний изменяется в зависимости от скорости передачи. Время испытаний составит по меньшей мере одну минуту (с числом переданных M-последовательностей и допустимым количеством ошибок, увеличенными соответственно).

d) Ожидается, что явление наиболее вероятно будет влиять на обработку внутренних аналоговых сигналов испытываемого устройства и с очень маленькой вероятностью функциональностью связи SDCI. Поэтому существующие специфические для устройства производственные стандарты будут иметь приоритет над уровнями испытаний, определенными выше.

G.1.5 Требуемые проверки излучения

Определение пределов излучения не входит в область применения настоящего стандарта. Применяются требования семейства специфических для Устройства производственных стандартов, как правило, для общих промышленных сред согласно МЭК 61000-6-4.

Все проверки излучения будут проводиться на максимально доступной скорости передачи с самым малым временем цикла.

G.1.6 Тестовая конфигурация для Ведущего узла

G.1.6.1 Общие правила

Для испытаний Ведущего узла применяются следующие правила:

- на следующих диаграммах установки для испытаний показаны только кабели SDCI и источника питания. Все остальные кабели рассматриваются как требуемые соответствующими производственными стандартами;
- заземление Ведущего узла и Устройства производится в соответствии с соответствующими производственными стандартами или руководством по эксплуатации;
- если не оговорено иное, кабель SDCI будет иметь общую длину 20 м. Излишняя длина укладывается в индуктивную бухту диаметром 0,3 м, уложенную, где применимо, на высоте 0,1 м над базовым заземлением;
- где применимо, вспомогательные Устройства будут размещаться на 10 см выше базового заземления RefGND;

- типовая тестовая конфигурация состоит из Ведущего узла и двух Устройств, за исключением радиочастотных испытаний общего вида, где будет применяться только одно Устройство;
- каждый порт будет удовлетворять требованиям ЭМС.

G.1.6.2 Электростатические разряды

На рисунке G.1 показана установка для испытаний на устойчивость к электростатическим разрядам в соответствии с МЭК 61000-4-2.

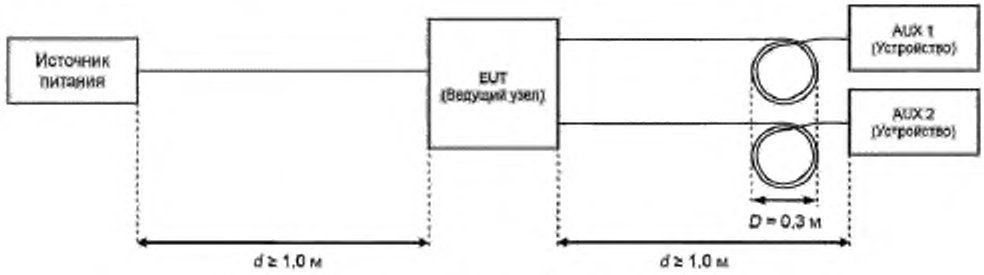


Рисунок G.1 — Установка для испытаний на устойчивость к электростатическим разрядам (Ведущий узел)

G.1.6.3 Радиочастотное электромагнитное поле

На рисунке G.2 показана установка для испытаний на устойчивость к радиочастотному электромагнитному полю в соответствии с МЭК 61000-4-3.

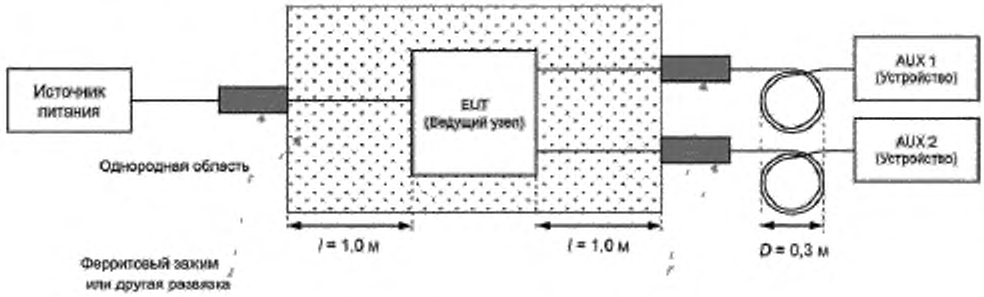


Рисунок G.2 — Установка для испытаний на устойчивость к радиочастотному электромагнитному полю (Ведущий узел)

G.1.6.4 Наносекундные импульсные помехи

На рисунке G.3 показана установка для испытаний на устойчивость к наносекундным импульсным помехам в соответствии с МЭК 61000-4-4. Никаких соединений в линию SDCI от AUX 2 не требуется.

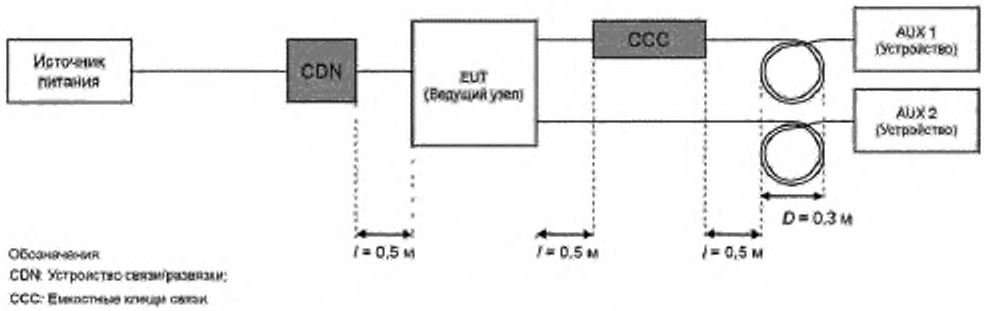


Рисунок G.3 — Установка для испытаний на устойчивость к наносекундным импульсным помехам (Ведущий узел)

G.1.6.5 Радиочастотные помехи общего вида

На рисунке G.4 показана установка для испытаний на устойчивость к радиочастотным помехам общего вида в соответствии с МЭК 61000-4-6.

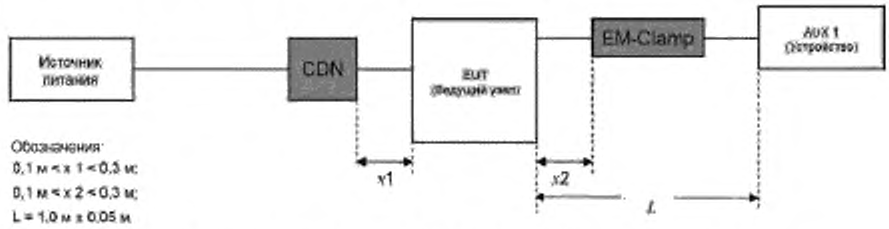


Рисунок G.4 — Установка для испытаний на устойчивость к радиочастотным помехам общего вида (Ведущий узел)

G.1.7 Тестовая конфигурация для Устройства

G.1.7.1 Общие правила

Для испытаний Устройства применяются следующие правила:

- на следующих диаграммах установки для испытаний показаны только кабели SDCI и источника питания. Все остальные кабели рассматриваются как требуемые соответствующими производственными стандартами;
- заземление Ведущего узла и Устройства в соответствии с соответствующими производственными стандартами или руководством пользователя;
- если не оговорено иное, кабель SDCI будет иметь общую длину 20 м. Излишняя длина укладывается в индуктивную бухту диаметром 0,3 м, расположенную на высоте 0,1 м над базовым заземлением;
- где применимо, вспомогательные Устройства будут размещаться на 10 см выше базового заземления RefGND;
- испытания с Устройством AUX 2 являются необязательными.

G.10.7.2 Электростатические разряды

На рисунке G.5 показана установка для испытаний на устойчивость к электростатическим разрядам в соответствии с МЭК 61000-4-2.

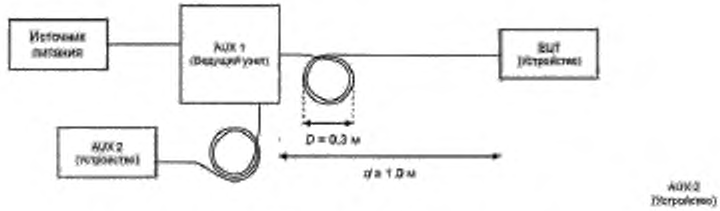


Рисунок G.5 — Установка для испытаний на устойчивость к электростатическим разрядам (Устройство)

G.1.7.3 Радиочастотное электромагнитное поле

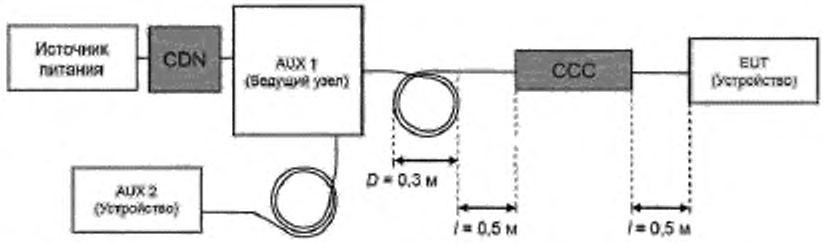
На рисунке G.6 показана установка для испытаний на устойчивость к радиочастотному электромагнитному полю в соответствии с МЭК 61000-4-3.



Рисунок G.6 — Установка для испытаний на устойчивость к радиочастотному электромагнитному полю (Устройство)

G.1.7.4 Наносекундные импульсные помехи

На рисунке G.7 показана установка для испытаний на устойчивость к наносекундным импульсным помехам в соответствии с МЭК 61000-4-4.



Обозначения:

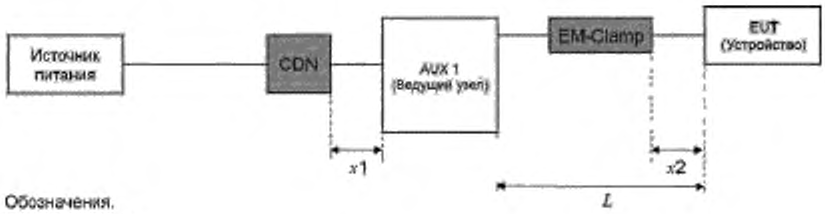
CDN: Устройство связи-развязки, здесь используется только для развязки;

CCC: Емкостные клещи связи.

Рисунок G.7 — Установка для испытаний на устойчивость к наносекундным импульсным помехам (Устройство)

G.1.7.5 Радиочастотные помехи общего вида

На рисунке G.8 показана установка для испытаний на устойчивость к радиочастотным помехам общего вида в соответствии с МЭК 61000-4-6.



Обозначения.

0,1 м < x 1 < 0,3 м;

0,1 м < x 2 < 0,3 м;

L = 1,0 м ± 0,05 м.

Рисунок G.8 — Установка для испытаний на устойчивость к радиочастотным помехам общего вида (Устройство)

G.2 Стратегии испытаний на соответствие

G.2.1 Испытания Устройства

Ведущий узел AUX 1 (см. рисунок G.5) будет непрерывно посылать M-последовательность типа TYPE_0 (читать страницу 2 Непосредственных параметров) сообщения со временем цикла, определенным в таблице G.1, и подсчитывать отсутствующие и ошибочные ответы Устройства. Оба показателя добавляются и отображаются.

Примечание — Подробные инструкции по испытанию Устройства устанавливаются в [9].

G.2.2 Испытания Ведущего узла

Устройство AUX 1 (см. рисунок G.1) будет применять M-последовательности типа TYPE_2_5. Его выходные PD будут генерироваться 8-битовым генератором случайных или псевдослучайных чисел. Ведущий узел будет копировать входные PD любых полученных сообщений Устройства в выходные PD следующего посылаемого сообщения Ведущего узла. Время цикла устанавливается в соответствии с таблицей G.1. Устройство AUX 1 будет сравнивать выходные PD с предварительно посланными входными PD и подсчитывать число отклонений. Устройство будет также подсчитывать число отсутствующих (не полученных в ожидаемое время цикла) или полученных искаженных сообщений Ведущего узла. Все показатели добавляются и отображаются.

Примечание 1 — Отклонение посланных и полученных PD указывает Устройству AUX 1, что испытываемое устройство (Ведущий узел) не получило сообщение Устройства.

Примечание 2 — Подробные инструкции по испытанию Устройства устанавливаются в [9].

Приложение Н
(справочное)

Вероятности остаточных ошибок

Н.1 Вероятность остаточных ошибок механизма целостности данных в интерфейсе SDCI

На рисунке Н.1 показана вероятность остаточных ошибок (REP) механизма целостности данных SDCI, состоящего из процедуры целостности данных контрольной суммы («XOR6») как определено в А.1.6, и четности UART. Диаграмма ссылается на МЭК 60870-5-1 с его классом I2 целостности данных для минимального расстояния Хемминга, равного 4 (красная пунктирная линия).

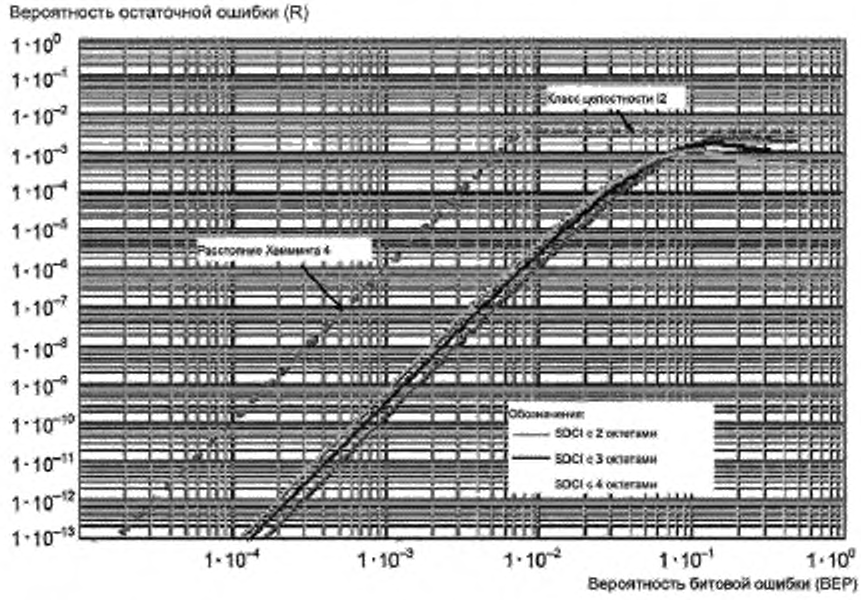


Рисунок Н.1 — Вероятность остаточной ошибки механизма целостности данных SDCI

Голубая линия показывает кривую остаточной ошибки для длины данных в два октета. Черная линия показывает кривую остаточной ошибки для длины данных в три октета. Фиолетовая линия показывает кривую остаточной ошибки для длины данных в четыре октета.

Н.2 Получение условий испытаний по ЭМС

Критерий эффективности А в G.1.3 выводится из требований, установленных в МЭК 61158-2 относительно чувствительности к помехам и частоты появления ошибок (ссылка, «фреймы» транслируются в «сообщения» в рамках настоящего стандарта):

- *только один необнаруженный ошибочный фрейм за 20 лет 1600 фреймов/с;*
- *отношение необнаруженных фреймов к обнаруженным фреймам не будет превышать 10^{-6};*
- *испытания по ЭМС не будет показывать более шести ошибочных фреймов в 100 000 фреймов.*

В применении к SDCI первое требование преобразуется в уравнение (Н.1). Данное уравнение позволяет определить значение BEP. Данное уравнение может быть разрешено в численном виде

$$F20 \cdot R\,(BEP) \le 1, \qquad (Н.1)$$

где F20 — число сообщений за 20 лет;

R(BEP) — вероятность остаточной ошибки контрольной суммы и механизм четности (рисунок Н.1);

BEP — вероятность битовой ошибки на рисунке Н.1.

Цель испытаний по ЭМС — доказать, что BEP связи SDCI удовлетворяет значению, определенному на первом шаге. Максимальное число обнаруженных искаженных сообщений выбрано равным 6 из чисто практических сообщений. Число требуемых тестовых сообщений SDCI может быть определено с помощью уравнения (H.2) и значения BEP, определенного на первом шаге.

$$11\ \mathrm{NoTF} \geq (1/\mathrm{BEP} \cdot 1/\mathrm{BitPerF} \cdot \mathrm{NopErr}), \qquad \text{(H.2)}$$

где NoTF — число тестовых сообщений;

BitPerF — число битов в сообщении;

NopErr — максимальное число обнаруженных искаженных сообщений = 6.

Уравнение (H.2) справедливо только в предположении, что сообщения с одной битовой ошибкой более частое, чем сообщения с большим числом ошибок. M-последовательность состоит из двух сообщений. Поэтому вычисленное число тестовых сообщений следует разделить на два при определении числа M-последовательностей в таблице G.1.

Приложение I
(справочное)

Примерная последовательность передачи ISDU

На рисунке I.1 показан пример передачи ISDU, применяя услугу AL_Read с 16-битовым Индексом и Субиндексом для 19 октетов данных пользователя, с отображением на M-последовательность типа TYPE_2_5 для датчиков и с прерыванием в случае передачи События.

		Ведущий узел				Устройство			
		FC	CKT	PD	OD	OD	PD	CKS	
Комментарий (состояние, действие) (см. таблицу 46)	№ цикла	R W Com Chan, Flow CTRL 1 бит 2 бита 5 бит	Frame Typ 2 бита	ChK 6 бит	PD 8 бит	Данные запроса: Ведущий узел 8 бит	Данные запроса: Устройство 8 бит	PD	CHD E PD	Комментарий (состояние. действие)
Idle_1	0	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx		OnReq idle
ISDURequest_2, передача	1	0111 0000	10 xxxxxx	xxxxxxxx	1011 0101		xxxxxxxx	0 0 xxxxxx		ISDURequest_2, прием
ISDURequest_2, передача	2	0110 0001	10 xxxxxx	xxxxxxxx	Index(hi)		xxxxxxxx	0 0 xxxxxx		ISDURequest_2, прием
ISDURequest_2, передача	3	0110 0010	10 xxxxxx	xxxxxxxx	Index(lo)		xxxxxxxx	0 0 xxxxxx		ISDURequest_2, прием
ISDURequest_2, передача	4	0110 0011	10 xxxxxx	xxxxxxxx	Subindex		xxxxxxxx	0 0 xxxxxx		ISDURequest_2, прием
ISDURequest_2, передача	5	0110 0100	10 xxxxxx	xxxxxxxx	CHKPDU		xxxxxxxx	0 0 xxxxxx		ISDURequest_2, прием
ISDUWait_3, запуск таймера ISDU	6	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx		ISDUWait_3, приложение занято
ISDUWait_3, + таймер ISDU	7	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx		ISDUWait_3, приложение занято
ISDUWait_3, + таймер ISDU	8	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx		ISDUWait_3, приложение занято
ISDUWait_3, + таймер ISDU	9	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx		ISDUWait_3, приложение занято
ISDUWait_3, + таймер ISDU	10	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx		ISDUWait_3, приложение занято
ISDUResponse_4, прием Стоп таймера ISDU	11	1111 0000	10 xxxxxx	xxxxxxxx		1101 0001	xxxxxxxx	0 0 xxxxxx		ISDUResponse_4. передача
ISDUResponse_4, прием	12	1110 0001	10 xxxxxx	xxxxxxxx		0001 0011	xxxxxxxx	0 0 xxxxxx		ISDUResponse_4, передача
ISDUResponse_4, прием	13	1110 0010	10 xxxxxx	xxxxxxxx		Data 1	xxxxxxxx	0 0 xxxxxx		ISDUResponse_4, передача
ISDUResponse_4, прием	14	1110 0011	10 xxxxxx	xxxxxxxx		Data 2	xxxxxxxx	0 0 xxxxxx		ISDUResponse_4, передача
ISDUResponse_4, прием	15	1110 0100	10 xxxxxx	xxxxxxxx		Data 3	xxxxxxxx	0 0 xxxxxx		ISDUResponse_4. передача
ISDUResponse_4, прием	16	1110 0101	10 xxxxxx	xxxxxxxx		Data 4	xxxxxxxx	0 0 xxxxxx		ISDUResponse_4. передача
ISDUResponse_4, прием	17	1110 0110	10 xxxxxx	xxxxxxxx		Data 5	xxxxxxxx	0 0 xxxxxx		ISDUResponse_4, передача
ISDUResponse_4, прием	18	1110 0111	10 xxxxxx	xxxxxxxx		Data 6	xxxxxxxx	0 0 xxxxxx		ISDUResponse_4, передача
ISDUResponse_4, прием	19	1110 1000	10 xxxxxx	xxxxxxxx		Data 7	xxxxxxxx	0 0 xxxxxx		ISDUResponse_4. передача

Рисунок I.1 — Пример обмена сообщениями ISDU, лист 1

Ведущий узел						Устройство			
		FC	CKT	PD	OD	OD	PD	CKS	
Комментарий (состояние, действие) (см. таблицу 46)	№ цикла	R Com Flow W Chan. CTRL 1 бит 2 бита 5 бит	Frame CHK Typ 2 бита 6 бит	PD 8 бит	Данные запроса Ведущий узел 8 бит	Устройство 8 бит	PD	CHD E PD	Комментарий (состояние, действие)
ISDUResponse_4, нет ответа, повтор в следующем цикле	20	1110 1001	10 Err	xxxxxxxx				xxxxxx	ISDUResponse_4, нарушение контрольной суммы, не посылать ответ
ISDUResponse_4, нет ответа, повтор в следующем цикле	21	1110 1001	10 Err	xxxxxxxx				xxxxxx	ISDUResponse_4, нарушение CHK, не посылать ответ
ISDUResponse_4, прием	22	1110 1001	10 xxxxxx	xxxxxxxx		Data 8	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4 передача
ISDUResponse_4, прием	34	1110 1010	10 xxxxxx	xxxxxxxx		Data 9	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4 передача
ISDUResponse_4, прием, пуск обработчика События	35	1110 1011	10 xxxxxx	xxxxxxxx		Data 10	xxxxxxxx	1 0 xxxxxx	ISDUResponse_4 передача заморозить событие
Read_Event_2, прием	36	1100 0000	10 xxxxxx	xxxxxxxx		Diag State with detail	xxxxxxxx	1 0 xxxxxx	Read_Event_2, передача
Read_Event_2, прием	37	110x xxxx	10 xxxxxx	xxxxxxxx		Event qualifier	xxxxxxxx	1 0 xxxxxx	Read_Event_2, передача
CommandHandler_2, передача, установить состояние PDOutdata в «недопустимое»	38	0010 0000	10 xxxxxx	xxxxxxxx	1001 1001		xxxxxxxx	1 0 xxxxxx	CommandHandler_2, прием, установить состояние PDOutdata в «недопустимое»
Read_Event_2, прием	39	110x xxxx	10 xxxxxx	xxxxxxxx		ErrorCode msb	xxxxxxxx	1 0 xxxxxx	Read_Event_2, передача
Read_Event_2, прием	40	110x xxxx	10 xxxxxx	xxxxxxxx		ErrorCode lsb	xxxxxxxx	1 0 xxxxxx	Read_Event_2, передача
EventConfirmation_4 передача простои обработчика событий	41	0100 0000	10 xxxxxx	xxxxxxxx	xxxxxxxx		xxxxxxxx	0 0 xxxxxx	Read_Event_2, передача
ISDUResponse_4, прием	42	1110 1100	10 xxxxxx	xxxxxxxx		Data 11	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, передача
ISDUResponse_4, прием	43	1110 1101	10 xxxxxx	xxxxxxxx		Data 12	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, передача
ISDUResponse_4, прием	44	1110 1110	10 xxxxxx	xxxxxxxx		Data 13	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, передача
ISDUResponse_4, прием	45	1110 1111	10 xxxxxx	xxxxxxxx		Data 14	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, передача
ISDUResponse_4, прием	46	1110 0000	10 xxxxxx	xxxxxxxx		Data 15	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, передача
ISDUResponse_4, прием	47	1110 0001	10 xxxxxx	xxxxxxxx		Data 16	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, передача
ISDUResponse_4, прием	48	1110 0010	10 xxxxxx	xxxxxxxx		CHKPDU	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, передача
Idle 1	49	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle 1
Idle 1	50	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle 1
Idle 1	51	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle 1
Idle_1	52	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Write Parameter, передача	53	0011 0000	10 xxxxxx	xxxxxxxx	xxxxxxxx		xxxxxxxx	0 0 xxxxxx	Write Parameter, прием

Рисунок I.1, лист 2

Ведущий узел						Устройство			
		FC	CKT	PD	OD	OD	PD	CKS	
Комментарий (состояние, действие) (см. таблицу 46)	№ цикла	R Com Flow W Chan. CTRL 1 бит 2 бита 5 бит	Frame CHK Typ 2 бита 6 бит	PD 8 бит	Данные запроса Ведущий узел 8 бит	Устройство 8 бит	PD	CHD E PD	Комментарий (состояние, действие)
Read Parameter, прием	54	1011 0000	10 xxxxxx	xxxxxxxx		xxxxxxxx	xxxxxxxx	0 0 xxxxxx	Read Parameter, передача
Idle 1	55	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle 1
Idle 1	56	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle 1
Idle_1	57	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
ISDURequest_2, передача	58	0111 0000	10 xxxxxx	xxxxxxxx	0001 1011				
ISDURequest_2, передача	59	0110 0001	10 xxxxxx	xxxxxxxx	Index				
ISDURequest_2, передача	60	0110 0010	10 xxxxxx	xxxxxxxx	Data 1				
ISDURequest_2, передача	61	0110 0011	10 xxxxxx	xxxxxxxx	Data 2				
ISDURequest_2, передача	62	0110 0100	10 xxxxxx	xxxxxxxx	Data 3				
ISDURequest_2, передача	63	0110 0101	10 xxxxxx	xxxxxxxx	Data 4				
ISDURequest_2, передача	64	0110 0110	10 xxxxxx	xxxxxxxx	Data 5				
ISDURequest_2, передача	65	0110 0111	10 xxxxxx	xxxxxxxx	Data 6				
ISDURequest_2, передача	66	0110 1000	10 xxxxxx	xxxxxxxx	Data 7				
ISDURequest_2, передача	67	0110 1001	10 xxxxxx	xxxxxxxx	Data 8				
ISDURequest_2, передача	68	0110 1010	10 xxxxxx	xxxxxxxx	CHKPDU				
ISDUWait_3, пуск таймер ISDU	69	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001			
ISDUResponse_4, прием, стоп таймера ISDU	70	1111 0000	10 xxxxxx	xxxxxxxx		0101 0010			
ISDUResponse_4, прием	71	1110 0001	10 xxxxxx	xxxxxxxx		CHKPDU			
Idle 1	72	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000			
Idle_1	73	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000			
ISDURequest_2, передача,	74	0111 0000	10 xxxxxx	xxxxxxxx	1011 0101		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, прием
ISDURequest_2, передача	75	0110 0001	10 xxxxxx	xxxxxxxx	Index(hi)		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, прием
ISDURequest_2, передача	76	0110 0010	10 xxxxxx	xxxxxxxx	Index(lo)		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, прием
ISDURequest_2, передача	77	0110 0011	10 xxxxxx	xxxxxxxx	Subindex		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, прием
ISDURequest_2, передача	78	1111 0000	10 xxxxxx	xxxxxxxx	CHKPDU		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, прием
ISDUWait_3, пуск таймера ISDU	79	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, приложение занято
ISDUWait_3, * таймер ISDU	80	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, приложение занято
ISDUWait_3, * таймер ISDU	81	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, приложение занято

Рисунок I.1, лист 3

Ведущий узел							Устройство			
		FC		CKT	PD	OD	OD	PD	CKS	
Комментарий (состояние, действие) (см. таблицу 46)	№ цикла	R W	Com Chan.	Flow CTRL	Frame Typ	CHK		Данные запроса	PD	CHD E PD	Комментарий (состояние, действие)
							Ведущий узел	Устройство			
		1 бит	2 бита	5 бит	2 бита	6 бит	8 бит	8 бит	8 бит		
ISDUWait_3, * таймер ISDU	82	1111	0000		10	xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, приложение занятоI
ISDUWait_3, * таймер ISDU	83	1111	0000		10	xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	SDUWait_3, приложение занято
ISDUResponse_4, прием Остановить таймер ISDU	84	1111	0000		10	xxxxxx	xxxxxxxx		1101 0001	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, передача
ISDUResponse_4, прием	85	1110	0001		10	xxxxxx	xxxxxxxx		0001 1110	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, передача
ISDUResponse_4, прием	86	1110	0010		10	xxxxxx	xxxxxxxx		Data 1	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, передача
ISDUResponse_4, ПРЕРВАТЬ	87	1111	1111		10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4 ПРЕРВАТЬ
Idle 1	88	1111	0001		10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle 1
Idle_1	89	1111	0001		10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1

Рисунок I.1, лист 4

Приложение J
(справочное)

Рекомендуемые методы определения изменения параметров

J.1 Сигнатура CRC

Циклический избыточный контроль (CRC) принадлежит к семейству функций хеширования HASH. Сигнатура CRC по всем изменяемым параметрам может вычисляться Устройством с помощью так называемого правильного образующего многочлена. Вычисление приводит к различным значениям сигнатуры при любом изменении набора параметров. Следует отметить, что сигнатура обеспечивает также упорядоченность октетов в наборе параметров. Любое изменение в порядке при вычислении сигнатуры приводит к отличному значению. Качество защиты (необнаруженные изменения) серьезно зависит как от образующего многочлена CR, так и от длины (числа октетов) набора параметров. Начальное значение должно быть больше нуля. Один метод вычисления прямо применяет формулу, другой применяет перемещение октетов и таблицы поиска. Первый метод требует меньше программной памяти и работает медленнее. Второй метод требует памяти для таблицы поиска ($1 \cdot 2^{10}$ октетов для 32-битовой сигнатуры) и является быстрым. Сравнение наборов данных параметров выполняется в состоянии «Checksum_9» конечной машины DS на рисунке 100. В таблице J.1 перечисляется несколько возможных образующих многочленов и их уровень определения изменений.

Таблица J.1 — Правильные образующие многочлены CRC

Образующий многочлен	Сигнатура	Длина данных	Необнаруженные изменения
0x9B	8-битовая	1 октет	$< 2^{-8}$ (не рекомендуется)
0x4EAB	16-битовая	1 < число октетов < 3	$< 2^{-16}$ (не рекомендуется)
0x5D6DCB	24-битовая	1 < число октетов < 4	$< 2^{-24}$ (не рекомендуется)
0xF4ACFB13	32-битовая	1 < число октетов < 2^{32}	$< 2^{-32}$ (рекомендуется)

J.2 Счетчик изменений

Для подсчета любых изменений набора параметров может быть реализован 32-битовый счетчик. Устройство будет применять случайное начальное значение Счетчика изменений. Значение самого счетчика не будет сохраняться через Список индексов Устройства. После скачивания фактическое значение счетчика считывается из Устройства во избежание множественных записей, инициированных последовательностью скачивания. Сравнение наборов данных параметров выполняется в состоянии «Checksum_9» конечной машины DS на рисунке 100.

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
IEC 60947-5-2	—	*
IEC 61000-4-2	—	*
IEC 61000-4-3	—	*
IEC 61000-4-4	—	*
IEC 61000-4-5	—	*
IEC 61000-4-6	—	*
IEC 61000-4-11	—	*
IEC 61000-6-2	—	*
IEC 61000-6-4	—	*
IEC 61076-2-101	—	*
IEC 61131-1	—	*
IEC 61131-2	—	*
IEC/TR 62390	—	*
ISO/IEC 646:1991	—	*
ISO/IEC 646:1991	—	*
ISO/IEC 10646	—	*
ISO/IEC 10731	—	*
ISO/IEC 19505 (all parts)	—	*
ISO 1177	—	*
IEEE Std 754—2008	—	*
(IETF): RFC 5905	—	*
* Соответствующий национальный стандарт отсутствует.		

Библиография

[1] IEC 60050 (all parts), International Electrotechnical Vocabulary (available at <http://www.electropedia.org>)
[2] IEC 60870-5-1:1990, Telecontrol equipment and systems — Part 5: Transmission protocols — Section One: Transmission frame formats
[3] IEC 61158-2, Industrial communication networks — Fieldbus specifications — Part 2: Physical layer specification and service definition
[4] IEC/TR 62453-61, Field device tool interface specification — Part 61: Device Type Manager (DTM) Styleguide for common object model
[5] ISO/IEC 7498-1, Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model
[6] IO-Link Consortium, IO Device Description (IODD), V1.1, Order No. 10.012 (available at <http://www.io-link.com>)
[7] IO-Link Consortium, IO-Link Smart Sensor Profile, V1.1, Order No. 10.042 (available at <http://www.io-link.com>)
[8] IO-Link Consortium, IO-Link Communication, V1.0, January 2009, Order No. 10.002 (available at <gost.e-ecolog.ru
[9] IO-Link Consortium, IO-Link Test Specification, V1.1, Order No. 10.032 (available at <gost.e-ecolog.ru

УДК 681.58:681.3:006.354

ОКС 35.240.50
25.040.40

Ключевые слова: программируемые контроллеры, одноточечный интерфейс цифровой связи (SDCI), физический уровень, канальный уровень, прикладной уровень, модуль управления системой, данные запроса, данные процесса

БЗ 10—2017/158

Редактор *Р.Г. Говердовская*
Технический редактор *В.Н. Прусакова*
Корректор *Е.Р. Ароян*
Компьютерная верстка *Ю.В. Половой*

Сдано в набор 18.09.2017 Подписано в печать 19.10.2017. Формат 60 × 84 1/8. Гарнитура Ариал.
Усл. печ. л. 27.9. Уч.-изд. л. 27,15 Тираж 22 экз. Зак. 2289.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

ИД «Юриспруденция», 115419, Москва, ул. Орджоникидзе, 11
www.jurisizdat.ru gost.e-ecolog.ru

Издано и отпечатано во ФГУП «СТАНДАРТИНФОРМ». 123001, Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru