ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО 20242-2— 2012

Системы промышленной автоматизации и интеграция

СЛУЖЕБНЫЙ ИНТЕРФЕЙС ДЛЯ ИСПЫТАТЕЛЬНЫХ ПРИКЛАДНЫХ ПРОГРАММ

Часть 2

Служебный интерфейс управления ресурсами

ISO 20242-2:2010
Industrial automation systems and integration — Service interface for testing applications — Part 2: Resource management service interface (IDT)

Издание официальное

Москва
Стандартинформ
2014

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН АНО «Международная академия менеджмента и качества бизнеса» на основе собственного аутентичного перевода на русский язык международного стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 100 «Стратегический и инновационный менеджмент»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 29 ноября 2012 г. № 1710-ст

4 Настоящий стандарт идентичен международному стандарту ИСО 20242-2:2010 «Системы промышленной автоматизации и интеграция. Служебный интерфейс для испытательных прикладных программ. Часть 2. Служебный интерфейс управления ресурсами (ISO 20242-2:2010 «Industrial automation systems and integration — Service interface for testing applications — Part 2: Resource management service interface»)

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в ГОСТ Р 1.0 — 2012 (раздел 8). 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 информационном 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 (gost.ru)

© Стандартинформ, 2014

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 и определения . . . 1
4 Сокращения . . . 1
5 Определения сервисов и сервисных процедур . . . 2
5.1 Общие положения . . . 2
5.2 Параметры . . . 2
5.3 Сервисные процедуры . . . 2
5.4 Сервисные примитивы и диаграммы состояния . . . 3
6 Сервисы управления ресурсами . . . 3
6.1 Обзор . . . 3
6.2 Перечень сервисов . . . 4
6.3 Сервисы поддержки управления . . . 7
6.4 Сервисы входа/выхода . . . 9
6.5 Расширенные сервисы . . . 18
6.6 Сервис поддержки обслуживания . . . 21
6.7 Устройство управления RMS-состоянием . . . 37
Приложение А (справочное) Практическое руководство пользователя интерфейса RMSI. Описание сервисов на языке программирования C/C++ . . . 41
Приложение В (справочное) Каскадирование драйверов устройств с помощью интерфейса RMSI 52
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации . . . 53

Введение

Настоящий стандарт разработан для облегчения интеграции измерительных и автоматических, а также других периферийных устройств в различных компьютеризированных областях применения. В стандарте определены принципы создания драйверов устройств и режимы их работы в области использования измерительных автоматических средств.

Основная цель комплекса стандартов ИСО 20242 — обеспечение:

- независимости пользователя от операционной системы;
- независимости пользователя от технологии соединения (интерфейс устройства/сеть);
- независимости пользователя от поставщиков устройств;
- возможности сертификации драйверов устройств с подсоединенными к ним устройствами и выбранными режимами работы (с учетом применяемой компьютерной платформы);
- независимости пользователя от последующих технологических усовершенствований устройств.

Стандарты комплекса ИСО 20242 не распространяются на разработку новых семейств устройств или использование специальных технологий для интерфейсов (сетей). В стандартах приведены общие описания сетей существующих устройств и их коммуникационных интерфейсов, обеспечивающих совместимость интерфейсов с другими устройствами аналогичного типа и назначения.

Комплекс стандартов ИСО 20242 включает требования, распространяющиеся на:

- служебный интерфейс для управления ресурсами;
- служебный интерфейс виртуального устройства;
- шаблон функциональных характеристик устройства;
- служебный интерфейс прикладных программ;
- методы проверки на совместимость, критерии и отчеты о проведенных проверках.

Комплекс стандартов ИСО 20242 состоит из следующих частей:

- часть 1. Общий обзор;
- часть 2. Служебный интерфейс управления ресурсами.

Следующие стандарты находятся в стадии разработки:

- часть 3. Служебный интерфейс виртуального устройства;
- часть 4. Шаблон профиля функциональных возможностей устройства.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Системы промышленной автоматизации и интеграция

СЛУЖЕБНЫЙ ИНТЕРФЕЙС ДЛЯ ИСПЫТАТЕЛЬНЫХ ПРИКЛАДНЫХ ПРОГРАММ

Часть 2

Служебный интерфейс управления ресурсами

Industrial automation systems and integration.
Service interface for testing applications. Part 2. Resource management service interface

Дата введения — 2014 — 01 – 01

1 Область применения

Настоящий стандарт определяет сервисный интерфейс, предоставляющий групповую точку доступа к сервису управления ресурсами и их обслуживания. Указанный интерфейс поддерживается операционной системой компьютера и его периферийными устройствами, включая специальные аппаратные средства на схемных платах, используемых при испытаниях приложений с компьютерной поддержкой. Сервисный интерфейс управления ресурсами предполагает групповое применение компьютерного адаптера платформы независимо от операционной системы и имеющихся интерфейсов связи.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты, которые необходимо учитывать при применении настоящего стандарта. При ссылках на документы с указанной датой утверждения необходимо пользоваться только данной редакцией, если эта дата не приведена, — последней редакцией ссылочных документов, включая любые поправки и изменения к ним.

ИСО 20242-1 Системы промышленной автоматизации и интеграция. Служебный интерфейс для испытательных прикладных программ. Часть 1. Обзор (ISO 20242-1 Industrial automation systems and integration — Service interface for testing applications — Part 1: Overview)

3 Термины и определения

В настоящем стандарте использованы следующие термины с соответствующими определениями:

3.1 **драйвер устройства** (device driver): Компьютерный программный модуль, обеспечивающий интерфейс с сервисными (служебными) функциями (в соответствии с ИСО 20242) и открывающий доступ адаптера платформы к физическим устройствам.

3.2 **адаптер платформы** (platform adapter): Компьютерный программный модуль, обеспечивающий работу сервисного (служебного) интерфейса управления ресурсами (в соответствии с ИСО 20242), включающего компьютерную платформу, операционную систему, аппаратные средства и периферийные устройства.

4 Сокращения

В настоящем стандарте использованы следующие сокращения:

CNF, Cnf — Подтверждение (сервисный примитив);
IND, Ind — Индикация (сервисный примитив);
REQ, Req — Запрос (сервисный примитив);
RMS — Сервисное управление ресурсами;

Издание официальное

RMSI — Сервисный интерфейс управления ресурсами;
RSP, Rsp — Отклик (сервисный примитив);
SAP — Точка доступа к севису.

5 Определения сервисов и сервисных процедур

5.1 Общие положения

Настоящий стандарт использует описательные определения в соответствии с ИСО/МЭК 10731.

Интерфейс между RMS-пользователем и RMS-провайдером описывается сервисными примитивами, передающими значения параметров.

Аспекты передачи данных находятся вне области применения ИСО 20242, поэтому для описания событий, происходящих на RMS-провайдере, используются только примитивы запроса и примитивы подтверждения. Примитивы индикации и примитивы отклика применяются для анализа событий, происходящих на RMS-провайдере. Сервисная модель, сервисный примитив и диаграмма последовательности — абстрактные описания, не дающие конкретной базы для практической реализации.

Правила практической реализации приведены в приложении А.

5.2 Параметры

Сервисные примитивы, представляющие взаимодействия между сервисным пользователем и сервисным провайдером (см. ИСО/МЭК 10731), передают значения параметров, т. е. информацию, используемую и обмениваемую в рамках этих взаимодействий.

Настоящий стандарт устанавливает табличный формат описания компонентов параметров RMS-примитивов (см. таблицу 1). Параметры каждой группы RMS-примитивов приведены в таблицах ниже, каждая их которых имеет три столбца. Первый содержит названия сервисных параметров, второй — входные значения параметров либо примитива запроса, либо примитива индикации, третий — выходные значения параметров либо примитива подтверждения, либо примитива отклика.

Одно значение параметра (или его часть) внесено в каждую строку каждой таблицы. Под соответствующим столбцом сервисного примитива размещены представленные ниже коды. Они указывают тип используемого параметра примитива, а также область применения параметра данного столбца:

a) M: параметр является обязательным для данного примитива;
b) C: использование данного параметра зависит от других параметров или от RMS-возможностей;
c) S: параметр является выбранным элементом списка;
d) (пробел): значение параметра не передается RMS-пользователями или RMS-провайдерами.

Таблица 1 — Формат для параметров сервисных примитивов

Название параметра	REQ или IND	CNF или RSP
Аргумент	M	
Параметр 1	M	
Параметр 2	C	
Результат (+)		S
Параметр 3		M
Параметр 4		C
Результат (–)		S
Параметр 5		M

5.3 Сервисные процедуры

5.3.1 Сервисные услуги с RMS-подтверждением

RMS-пользователь представляет примитив запроса в RMSI, что указывает на существование точки доступа к сервису (SAP). Соответствующая сервисная преобразующая сущность доставляет примитив подтверждения пользователю либо после завершения всех необходимых взаимодействий, либо после ошибки.

5.3.2 Обработка RMS-событий

Пользователь создает точку доступа к сервису (SAP) на RMSI для обработки события, на наличие которого указывает примитив индикации в данном месте. Пользователь RMSI отправляет примитив отклика либо после завершения всех необходимых взаимодействий, либо после ошибки (см. рисунок 1).

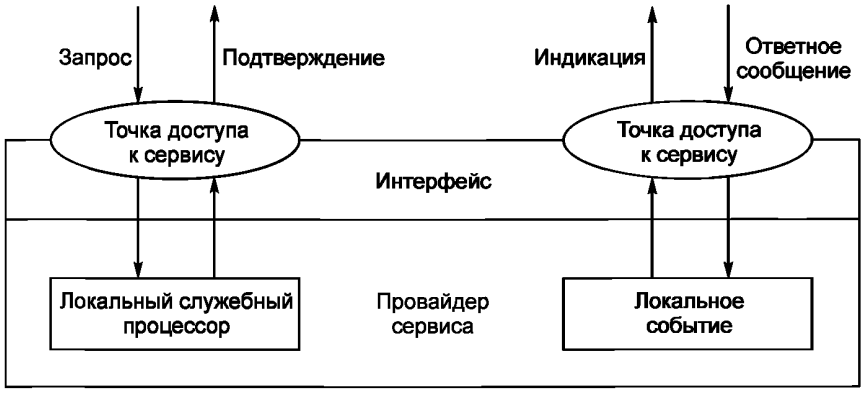
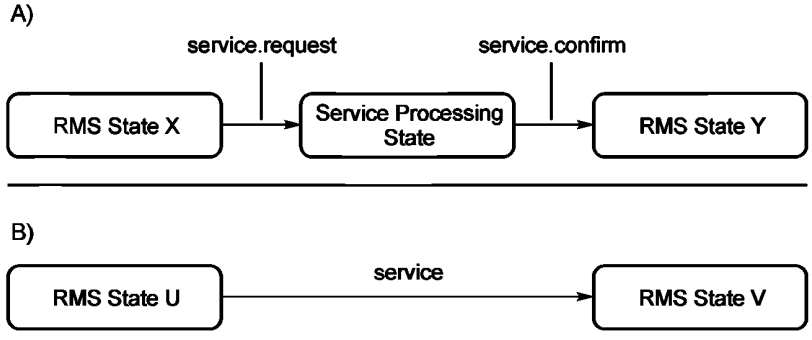


Рисунок 1 — Обработка локального RMS-события

5.4 Сервисные примитивы и диаграммы состояния

При необходимости UML-диаграммы (диаграммы на унифицированном языке моделирования) состояния используются для описания поведения RMS. На таких диаграммах только сервисные названия можно применять для описания изменения состояний. При этом нет необходимости явно указывать переходное состояние между примитивом запроса и примитивом подтверждения (см. рисунок 2B). В противном случае указывается переходное состояние, вызванное сервисом (см. рисунок 2A).



Service request — Запрос сервиса; Service confirm — Подтверждение получения сервиса; RMS State X — Состояние X сервисного управления ресурсами; Service processing state — Переходное состояние; RMS State Y — Состояние Y сервисного управления ресурсами; Service — Сервис; RMS State U — Состояние U сервисного управления ресурсами; RMS State V — Состояние V сервисного управления ресурсами

Рисунок 2 — Изменение состояния, вызванное сервисом

6 Сервисы управления ресурсами

6.1 Обзор

Интерфейс RMSI обеспечивает групповые сервисы поддержки управления, обслуживания и входа/выхода.

Сервисы входа/выхода обеспечивают доступ к нижележащему уровню иерархии, обеспечивающему расширенные сервисы, которые рассматриваются для описания структуры подгружаемых ресурсов для различных видов периферийных интерфейсов (см. рисунок 3).

Примечание 1 — ИСО 20242 не определяет методы интеграции сущности с расширенными сервисами в RMS-провайдере, которые зависят от операционной системы компьютера и языка программирования, используемого при практической реализации сервисных провайдеров. При этом расширенные сервисы необходимо описывать при расширении сервисов входа/выхода для различных периферийных интерфейсов без замены RMS-провайдера (см. пример практической реализации в приложении А).

Примечание 2 — Существуют особые каскадные методы (см. приложение Б) применения интерфейса RMSI в более сложных структурах интеграции устройств и оборудования.

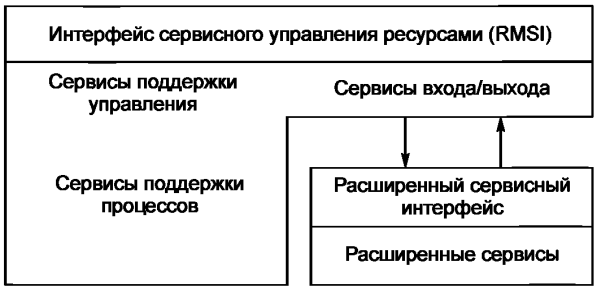


Рисунок 3 — Сервисные пользователи и сервисные провайдеры для интерфейса RMSI

6.2 Перечень сервисов

6.2.1 Групповые сервисы поддержки управления

Групповые сервисы поддержки управления используются для получения доступа к другим сервисам и для инициализации (и загрузки, при необходимости) расширенных сервисных провайдеров. Таблица 2 дает обзор указанных сервисов.

Таблица 2 — Групповые сервисы поддержки управления

Сервис	Название	Примечания
Получение сервисной ссылки	GetFuncAddress	Получение ссылки на сервис по идентификатору (название и/или количество) и номер версии
Инициализация типа периферийного интерфейса	io_initiate	Получение идентификатора указанного типа интерфейса и загрузка расширенного сервисного провайдера данного типа (при необходимости)
Принятие решения о типе периферийного интерфейса	io_conclude	Применение идентификатора типа и закрытие расширенного сервисного провайдера данного типа (при наличии)

6.2.2 Групповые сервисы входа/выхода

Групповые сервисы входа/выхода используются для установления связи с реальными устройствами, для конфигурирования периферийного интерфейса и управления им. Таблица 3 представляет обзор указанных сервисов.

Таблица 3 — Групповые сервисы входа/выхода

Сервис	Название	Примечания
Открыть канал периферийного интерфейса	io_open	Открытие периферийного интерфейса передачи данных и конфигурирование данного интерфейса
Реконфигурировать канал периферийного интерфейса	io_config	Изменение конфигурации интерфейса без его закрытия, например без изменения параметров передачи
Считать данные	io_read	Извлечение полученных данных на периферийном интерфейсе
Ввести данные	io_write	Введение данных в периферийный интерфейс для передачи
Выполнить операцию	io_execute	Выполнение операции, принадлежащей каналу периферийного интерфейса, что сравнимо с обработкой сразу и считанных и введенных данных с помощью одного сервиса (обмен данными)
Отменить установленную связь	io_cancel	Отмена сервиса «Считать данные», «Ввести данные» или «Выполнить операцию» и подготовка интерфейса к новым запросам

Окончание таблицы 3

Сервис	Название	Примечания
Получить статус канала периферийного интерфейса	io_stat	Получение информации о статусе периферийного интерфейса
Очистить буфер считывания	io_clear	Стирание содержимого входного буфера периферийного интерфейса
Закрыть канал периферийного интерфейса	io_close	Закрытие периферийного интерфейса
Выдать сигнал о наступлении события	io_event	Индикация локального события, выдача отклика на источник события

Групповые сервисы входа/выхода преобразуются в соответствующие расширенные (см. таблицу 4), если загрузка расширенного сервисного провайдера соответствует указанному типу интерфейса.

6.2.3 Расширенные сервисы

Расширенные сервисы не видны пользователю интерфейса RMSI. Они определены в настоящем стандарте для практической реализации иерархической модульной структуры RMS с помощью расширенных сервисных провайдеров. Указанные сервисы в значительной степени совпадают с групповыми сервисами входа/выхода RMS.

Таблица 4 — Расширенные сервисы для периферийных интерфейсов

Сервис	Название	Примечания
Инициировать тип расширенного интерфейса	ext_initiate	Задание идентификатора указанного типа периферийного интерфейса
Принять решение о типе расширенного интерфейса	ext_conclude	Задание типа идентификатора для сервиса ext_initiate
Открыть расширенный интерфейс	ext_open	Открытие периферийного интерфейса для передачи данных и его конфигурации
Реконфигурировать расширенный интерфейс	ext_config	Изменение конфигурации периферийного интерфейса без его закрытия, например без изменения параметров передачи
Считать данные расширенного интерфейса	ext_read	Считывание данных с периферийного интерфейса
Ввести данные в расширенный интерфейс	ext_write	Ввод данных в периферийный интерфейс для передачи
Выполнить операцию на расширенном интерфейсе	ext_execute	Выполнение операции на соответствующем периферийном интерфейсе, что сравнимо с обработкой считываемых и вводимых данных одного сервиса (обмен данными)
Отменить расширенную связь	ext_cancel	Отмена сервиса «Считать данные расширенного интерфейса», «Ввести данные в расширенный интерфейс» и «Выполнить операцию на расширенном интерфейсе», а также подготовка интерфейса к новым запросам
Получить статус расширенного интерфейса	ext_stat	Получение информации о статусе интерфейса
Очистить буфер считывания расширенного интерфейса	ext_clear	Стирание содержимого входного буфера периферийного интерфейса
Закрыть периферийный интерфейс	ext_close	Закрытие периферийного интерфейса
Получить сигнал о расширенном событии	ext_event	Индикация расширенного события и ожидание отклика

6.2.4 Сервисы поддержки обслуживания

Сервисы поддержки обслуживания (см. таблицу 5) обеспечивают доступ к памяти, управление таймером, семафорами и другими ресурсами операционной системы компьютера.

Таблица 5 — Сервисы поддержки обслуживания

Сервис	Название	Примечания
Выделить место в памяти	os_allocate	Выделение согласованного места в памяти заданного размера для хранения данных
Перевыделить место в памяти	os_reallocate	Изменение размера места в памяти, выделенного для хранения данных
Освободить память	os_free	Очистка выделенного в памяти места для хранения данных
Получить время	os_time	Получение информации о местном времени
Получить время обработки данных	os_clock	Получение информации о времени работы центрального процессора (CPU-time)
Ожидать	os_delay	Ожидание в течение указанного времени
Создать таймер	os_settimer	Создание и запуск таймера
Обеспечить срабатывание таймера	os_timerEvent	Выдача сигнала о срабатывании и ожидании отклика таймера
Удалить таймер	os_killtimer	Остановка и удаление таймера
Создать световой таймер	os_setLPTimer	Создание и запуск таймера со световой индикацией, разрешение и точность которого зависят от типа процесса световой индикации
Сигнализировать о событии с помощью таймера со световой индикацией	os_LPTimerEvent	Выдача сигнала световым таймером об истечении времени и ожидание отклика
Удалить таймер со световой индикацией	os_killLPTimer	Остановка и стирание таймера со световой индикацией
Идентифицировать процесс световой индикации	os_getLPnumber	Идентификация реального процесса световой индикации
Создать вычислительные семафоры	os_createSem	Создание вычислительных семафоров для управления одновременно используемыми ресурсами
Ожидать вычислительный семафор	os_waitSem	Ожидание свободного доступа к защищенному ресурсу
Открыть вычислительный семафор	os_releaseSem	Открытие доступа к защищенному ресурсу
Стереть вычислительный семафор	os_deleteSem	Стирание вычислительного семафора
Создать частный семафор	os_createMutex	Создание частного семафора для управления доступом к ресурсам с помощью различных процессов световой индикации, исключающих взаимное влияние
Ожидать частный семафор	os_waitMutex	Ожидание свободного доступа к защищенному ресурсу
Открыть частный семафор	os_releaseMutex	Открытие доступа к защищенному ресурсу
Стереть частный семафор	os_deleteMutex	Стирание частного семафора
Открыть журнал отладки	os_openDebug	Открытие текстового журнала для сообщений отладки
Записать сообщение отладки	os_writeDebug	Посылка сообщения в текстовый журнал отладки
Закрыть журнал отладки	os_closeDebug	Закрытие текстового журнала отладки

6.3 Сервисы поддержки управления

6.3.1 Услуга получения сервиса

6.3.1.1 Содержание сервисной услуги

Такой сервис необходим для получения ссылки на другие сервисы управления ресурсами, зависящие от версии программного обеспечения. Данные сервисы запрашиваются RMS-пользователем для каждого сервиса управления ресурсами, требуемого для практической реализации приложения.

6.3.1.2 Структура сервисного параметра

Параметры сервиса получения сервисной ссылки приведены в таблице 6.

Таблица 6 — Структура параметра получения сервисной ссылки

Название параметра	Req (запрос)	Cnf (подтверждение)
Аргумент	M	
Сервисный идентификатор (название)	M	
Требуемый номер версии	M	
Результат (+)		S
Сервисная ссылка		M
Результат (–)		S

6.3.1.3 Сервисные параметры

6.3.1.3.1 Аргумент

Аргумент содержит параметры сервисных запросов.

6.3.1.3.2 Сервисный идентификатор

Данный параметр идентифицирует сервис, для которого запрошена ссылка.

6.3.1.3.3 Требуемый номер версии

Данный параметр указывает версию программного обеспечения, которую RMS-пользователь запрашивает для сервиса.

6.3.1.3.4 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.3.1.3.5 Сервисная ссылка

Данный параметр содержит ссылку, идентифицирующую сервис с требуемым номером версии.

6.3.1.3.6 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.3.1.4 Сервисная процедура

Если сервис с указанным названием и номером версии доступен, то ссылка на него создана и предоставлена по запросу.

6.3.2 Сервис инициализации периферийного интерфейса

6.3.2.1 Содержание сервиса

Данный сервис запрашивает доступ к интерфейсу с указанным названием типа. Если название расширенного сервисного провайдера дано в запросе, то он также будет загружен и доступ к указанному интерфейсу будет запрошен для провайдера.

6.3.2.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 7.

Таблица 7 — Структура параметра инициализации периферийного интерфейса

Название параметра	Req	Cnf
Аргумент	M	
Название типа интерфейса	M	
Название провайдера расширенных сервисов	C	
Результат (+)		S
Идентификатор типа интерфейса		M
Результат (–)		S
Ошибка		M

6.3.2.3 Сервисные параметры

6.3.2.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.3.2.3.2 Название типа интерфейса

Данный параметр содержит название типа интерфейса.

6.3.2.3.3 Название провайдера расширенных сервисов

Данный условный параметр (при его наличии) содержит название провайдера расширенных сервисов, обеспечивающего сервисы входа/выхода для указанного типа интерфейса.

6.3.2.3.4 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.3.2.3.5 Идентификатор типа интерфейса

Данный параметр содержит номер, идентифицирующий рассматриваемый тип интерфейса для других сервисных запросов.

6.3.2.3.6 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.3.2.3.7 Ошибка (error)

Данный параметр указывает, что выполнено одно из следующих условий:

- тип интерфейса неизвестен или недоступен;
- расширенный сервисный провайдер неизвестен, не подгружается или не используется;
- тип интерфейса инициализирован ранее;
- при загрузке расширенного сервисного провайдера произошла ошибка памяти;
- выявлен сбой аппаратных средств.

6.3.2.4 Сервисная процедура

Данный сервис проверяет доступность интерфейса указанного типа. Если дано название расширенного сервисного провайдера, то он загружается. Если названия нет, то провайдер не загружается, и выполняется поиск указанного типа интерфейса.

6.3.3 Сервис принятия решения о типе периферийного интерфейса

6.3.3.1 Содержание сервиса

Данный сервис принимает решение о выборе периферийного интерфейса, инициированного ранее.

6.3.3.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 8.

Таблица 8 — Структура параметра принятия решения о выборе типа периферийного интерфейса

Название параметра	Req	Cnf
Аргумент	M	
Идентификатор типа интерфейса	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.3.3.3 Сервисные параметры

6.3.3.3.1 Аргумент

Аргумент содержит значение параметра запроса на данный сервис.

6.3.3.3.2 Идентификатор типа интерфейса

Данный параметр содержит идентификатор типа, который является результатом сервиса инициализации типа периферийного интерфейса.

6.3.3.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.3.3.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.3.3.3.5 Ошибка

Данный параметр указывает, что выполнено одно из следующих условий:

- идентификатор типа интерфейса неизвестен или недоступен;
- провайдер расширенных сервисов не может быть открыт;
- произошла ошибка памяти при открытии расширенного сервисного провайдера;
- выявлен сбой аппаратных средств.

6.3.3.4 Сервисная процедура

Данный сервис проверяет возможность открытия периферийного интерфейса указанного типа. Если он уже обработан соответствующим расширенным сервисным провайдером, то может быть открыт при условии, что все доступные периферийные интерфейсы всех типов уже открыты.

6.4 Сервисы входа/выхода

6.4.1 Сервисы открытия канала периферийного интерфейса

6.4.1.1 Содержание сервиса

Данный сервис открывает канал связи с периферийным интерфейсом указанного типа и задает конфигурацию канала по запросу пользователя.

6.4.1.2 Структура сервисного параметра

Сервисные параметры данного сервиса приведены в таблице 9.

Таблица 9 — Структура параметра открытия канала периферийного интерфейса

Название параметра	Req	Cnf
Аргумент	M	
Идентификатор типа интерфейса	M	
Название канала интерфейса	C	
Перечень параметров конфигурации	M	
Подтвержденная ссылка на точку доступа к сервису	M	
Ссылка на событие точки доступа к сервису	M	
Результат (+)		S
Идентификатор канала интерфейса		M
Результат (–)		S
Ошибка		M

6.4.1.3 Сервисные параметры

6.4.1.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.4.1.3.2 Идентификатор типа интерфейса

Данный параметр идентифицирует тип периферийного интерфейса.

6.4.1.3.3 Название канала интерфейса

Данный параметр идентифицирует установленный канал связи указанного типа. Он может быть опущен, если каналы реализованы сервисным управлением ресурсами RMS, а требуемые уникальные периферийные коннекторы (например, магистральной системы) не назначены.

6.4.1.3.4 Перечень параметров конфигурации

Перечень параметров конфигурации (не рассматриваемый в стандарте ИСО 20242) определяется типом выбранного периферийного интерфейса, который их задает.

Если существуют параметры конфигурации, годные для всех каналов указанного типа, то только первое применение данного сервиса (после инициализации типа интерфейса) выявляет их.

6.4.1.3.5 Подтвержденная ссылка на точку доступа к сервису

Данный параметр дает ссылку на точку доступа к специальному сервису — используется всеми подтвержденными сервисами в указанном канале.

6.4.1.3.6 Ссылка на событие в точке доступа к сервису

Данный параметр дает ссылку на точку доступа к специальному сервису — используется при обработке всех событий в канале.

6.4.1.3.7 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.4.1.3.8 Идентификатор канала интерфейса

Данный параметр — идентификатор, который применяется по особому запросу на сервис в канале.

6.4.1.3.9 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.4.1.3.10 Ошибка

Данный параметр указывает, что выполнено одно из следующих условий:

- интерфейс указанного типа недоступен;
- канал с таким названием недоступен;
- канал с указанным названием уже открыт;
- название канала отсутствует или неверно;
- ссылка на подтвержденный сервис в точке доступа отсутствует или неверна;
- ссылка на сервис обработки события в точке доступа отсутствует или неверна;
- произошла ошибка памяти;
- выявлен сбой аппаратных средств;
- произошла блокировка конфигурации канала;
- для открытия данного канала недостаточно ресурсов;
- не указан параметр ошибки;
- указан параметр ошибки.

6.4.1.4 Сервисная процедура

Данный сервис проверяет все параметры и при отсутствии ошибки делает попытку открыть указанный канал связи. Если канал готов к работе, то его идентификатор возвращается. В противном случае приходит сообщение об ошибке.

6.4.2 Сервис реконфигурации канала периферийного интерфейса

6.4.2.1 Содержание сервиса

Данный сервис используется для реконфигурации открытого канала связи периферийного интерфейса.

6.4.2.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 10.

Таблица 10 — Структура параметра реконфигурации канала периферийного интерфейса

Название параметра	Req	Cnf
Аргумент	M	
Идентификатор канала интерфейса	M	
Перечень параметров конфигурации	M	
Подтвержденная ссылка на точку доступа к сервису	M	
Ссылка на событие точки доступа к сервису	M	
Результат (+)		S
Идентификатор канала интерфейса		M
Результат (–)		S
Ошибка		M

6.4.2.3 Сервисные параметры

6.4.2.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.4.2.3.2 Идентификатор канала интерфейса

Данный параметр идентифицирует канал периферийного интерфейса.

6.4.2.3.3 Перечень параметров конфигурации

Перечень параметров конфигурации (находящийся вне области применения стандарта ИСО 20242) зависит от типа выбранного периферийного интерфейса, который их задает.

Если существуют параметры конфигурации, годные для всех каналов интерфейса данного типа, то они устанавливаются только в том случае, если идентификатор взят из первого открытого канала после инициализации его типа.

6.4.2.3.4 Подтвержденная ссылка на точку доступа к сервису

Данный параметр — ссылка на точку доступа к специальному сервису — используется для всех подтвержденных сервисов, принадлежащих каналу.

6.4.2.3.5 Ссылка на событие точки доступа к сервису

Данный параметр — ссылка на точку доступа к специальному сервису — применяется для всех сервисов обработки событий, принадлежащих каналу.

6.4.2.3.6 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.4.2.3.7 Идентификатор канала интерфейса

Данный параметр — идентификатор, который используется по особому запросу на сервис в канале.

6.4.2.3.8 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.4.2.3.9 Ошибка

Данный параметр указывает, что выполнено одно из следующих условий:

- канал недоступен;
- канал занят;
- отсутствует или неверна ссылка на точку доступа к подтвержденному сервису;
- отсутствует или неверна ссылка на точку доступа к сервису обработки события;
- произошла ошибка памяти;
- выявлен сбой аппаратных средств;
- произошла блокировка конфигурации канала;
- недостаточно ресурсов для конфигурации канала;
- не указана ошибка параметра;
- указана ошибка параметра.

6.4.2.4 Сервисная процедура

Сервис проверяет, задействован ли канал в передаче данных. Если задействован, то получается «Результат (–)» с сообщением об ошибке **"Channel busy** (Канал занят)". В противном случае все параметры проверяются, и (при отсутствии ошибки) указанный канал связи переконфигурируется. Если канал готов к работе, то получается «Результат (+)», в противном случае — «Результат (–)» с сообщением об ошибке.

6.4.3 Сервис считывания данных

6.4.3.1 Содержание сервиса

Сервис используется для получения данных по каналу периферийного интерфейса.

6.4.3.2 Структура сервисного параметра

Параметры данных сервисов приведены в таблице 11.

Таблица 11 — Структура параметров считывания данных

Название параметра	Req	Cnf
Аргумент	M	
Идентификатор канала интерфейса	M	
Получаемые данные управления процессом	M	
Максимальная длина полученных данных	M	
Максимальная продолжительность процесса	M	
Результат (+)		S
Данные управления процессом		M
Длина данных		M
Полученные данные		M
Результат (–)		S
Данные управления процессом		M
Длина данных		M
Полученные данные		M
Ошибка		M

6.4.3.3 Сервисные параметры

6.4.3.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.4.3.3.2 Идентификатор канала интерфейса

Указанный параметр идентифицирует канал периферийного интерфейса, ожидающего получения данных.

6.4.3.3.3 Получаемые данные управления процессом

Указанный параметр — идентификатор, определяемый пользователем для получения данных о процессе.

6.4.3.3.4 Максимальная длина полученных данных

Максимальная длина полученных данных — максимальное число восьмибитовых байтов, содержащихся в блоке полученного протокола.

6.4.3.3.5 Максимальная продолжительность процесса

Параметр содержит максимальную продолжительность получения данных в миллисекундах.

6.4.3.3.6 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.4.3.3.7 Получаемые данные управления процессом

Данный параметр — копия определенного пользователем порядка управления процессом, предоставленная по запросу на сервис.

6.4.3.3.8 Полученная длина данных

Число восьмибитовых байтов, содержащихся в блоке данных полученного протокола.

6.4.3.3.9 Полученные данные

Указанный параметр возвращает блок данных протокола, состоящий из одного или более восьмибитовых байтов. Их число всегда меньше максимальной длины данных, полученных по запросу, или равно ей.

6.4.3.3.10 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.4.3.3.11 Получаемые данные управления процессом

Данный параметр — копия определенного пользователем порядка управления процессом, предоставленная по запросу на сервис.

6.4.3.3.12 Полученная длина данных

Число восьмибитовых байтов, полученных до возникновения ошибки и прекращения получения данных о процессе.

6.4.3.3.13 Полученные данные

Указанный параметр возвращает полученные данные, состоящие из нуля и более восьмибитовых байтов.

6.4.3.3.14 Ошибка

Данный параметр указывает, что выполнено одно из следующих условий:

- канал недоступен;
- канал получения данных о процессе занят;
- произошла ошибка памяти;
- выявлен сбой аппаратных средств;
- буфер связи не размещен;
- неверно указана максимальная длина данных;
- произошла блокировка получения данных;
- получение данных о процессе отменено пользователем.

6.4.3.4 Сервисная процедура

Если канал доступен и не занят обработкой информации, то указанный сервис запускает получение данных о процессе. Работа заканчивается, когда получен полный блок данных протокола, достигнута их максимальная длина, произошла блокировка или возникла ошибка. При блокировке или ошибке входной буфер содержит все восьмибитовые байты, полученные за указанное время.

6.4.4 Сервис ввода данных

6.4.4.1 Содержание сервиса

Указанный сервис используется для передачи данных по каналу периферийного интерфейса.

6.4.4.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 12.

Таблица 12 — Структура параметра ввода данных

Название параметра	Req	Cnf
Аргумент	M	
Идентификатор канала интерфейса	M	
Передача данных управления процессом	M	
Длина передаваемых данных	M	
Передаваемые данные	M	
Максимальная продолжительность процесса	M	
Результат (+)		S
Передача данных управления процессом		M
Результат (–)		S
Передача данных управления процессом		M
Длина передаваемых данных		M
Ошибка		M

6.4.4.3 Сервисные параметры

6.4.4.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.4.4.3.2 Идентификатор канала интерфейса

Указанный параметр идентифицирует канал периферийного интерфейса для передачи данных.

6.4.4.3.3 Передача данных управления процессом

Указанный параметр — идентификатор, определяемый пользователем для передачи данных о процессе.

6.4.4.3.4 Длина передаваемых данных

Данный параметр указывает число передаваемых восьмибитовых байтов.

6.4.4.3.5 Передаваемые данные

Данный параметр содержит один или более передаваемых восьмибитовых байтов.

6.4.4.3.6 Максимальная продолжительность процесса

Указанный параметр содержит максимальную продолжительность передачи данных в миллисекундах.

6.4.4.3.7 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.4.4.3.8 Передача данных управления процессом

Данный параметр — копия определенного пользователем порядка управления процессом, предоставленная по запросу на сервис.

6.4.4.3.9 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.4.4.3.10 Передача данных управления процессом

Данный параметр — копия определенного пользователем порядка управления процессом, предоставленная по запросу на сервис.

6.4.4.3.11 Длина передаваемых данных

Число восьмибитовых байтов, переданных до возникновения ошибки или до остановки передачи данных о процессе.

6.4.4.3.12 Ошибка

Данный параметр указывает, что выполнено одно из следующих условий:

- канал недоступен;
- канал занят передачей данных о процессе;
- произошла ошибка памяти;
- выявлен сбой аппаратных средств;
- размещение в памяти буфера связи не выполнено;
- неверно указана максимальная длина данных;
- произошла блокировка передачи данных;
- передача данных о процессе отменена пользователем.

6.4.4.4 Сервисная процедура

Если канал доступен и не занят передачей информации, то сервис запускает передачу данных о процессе, которая заканчивается, когда передано заданное число восьмибитовых байтов, произошла блокировка или возникла ошибка. При блокировке или ошибке параметр "**Transmitted data length** (Длина передаваемых данных)" возвращает число переданных восьмибитовых байтов на момент блокировки или возникновения ошибки.

6.4.5 Сервис выполнения операции

6.4.5.1 Содержание сервиса

Указанный сервис используется для выполнения операции в данном канале периферийного интерфейса.

6.4.5.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 13.

Т а б л и ц а 13 — Структура параметра выполнения операции

Название параметра	Req	Cnf
Аргумент	M	
Идентификатор канала интерфейса	M	
Выполнение операции управления процессом	M	
Идентификатор операции	M	
Перечень входных параметров операции	M	
Максимальная продолжительность процесса	M	
Результат (+)		S
Выполнение операции управления процессом		M
Перечень выходных параметров операции		M
Результат (–)		S
Выполнение операции управления процессом		M
Ошибка		M

6.4.5.3 Сервисные параметры

6.4.5.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.4.5.3.2 Идентификатор канала интерфейса

Данный параметр идентифицирует канал периферийного интерфейса, включающий указанную операцию.

6.4.5.3.3 Выполнение операции управления процессом

Данный параметр — идентификатор, определяемый пользователем для выполняемого процесса.

6.4.5.3.4 Идентификатор операции

Данный параметр — идентификатор выполняемой операции.

6.4.5.3.5 Перечень входных параметров операции

Данный перечень зависит от вида выполняемой операции. Определения указанных операций и их аргументы в настоящем стандарте не рассматриваются.

6.4.5.3.6 Максимальная продолжительность процесса

Данный параметр содержит максимальную продолжительность выполнения операции в миллисекундах.

6.4.5.3.7 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.4.5.3.8 Выполнение операции управления процессом

Данный параметр — копия определенного пользователем порядка управления процессом, предоставленная по запросу на сервис.

6.4.5.3.9 Перечень выходных параметров операции

Данный перечень зависит от вида выполняемой операции. Определения операций и их аргументов в настоящем стандарте не рассматриваются.

6.4.5.3.10 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.4.5.3.11 Выполнение операции управления процессом

Данный параметр — копия определенного пользователем порядка управления процессом, доставленная по запросу на сервис.

6.4.5.3.12 Ошибка

Данный параметр указывает, что выполнено одно из следующих условий:

- канал недоступен;
- операция не найдена, идентификатор неправильный;
- операция находится в процессе выполнения;
- произошла ошибка памяти;
- выявлен сбой аппаратных средств;
- перечень входных параметров неправильный;

- произошла блокировка операции;
- выполнение операции отменено пользователем.

6.4.5.4 Сервисная процедура

Если канал доступен, и указанная операция еще не выполняется, то она будет запущена. Если произошла ошибка или операция отменена RMS-пользователем, то она будет остановлена.

6.4.6 Сервис отмены установления связи с процессом

6.4.6.1 Содержание сервиса

Данный сервис используется для отмены обработки получаемых или передаваемых данных о процессе или о выполнении операции.

6.4.6.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 14.

Таблица 14 — Структура параметра отмены установления связи с процессом

Название параметра	Req	Cnf
Аргумент	M	
Идентификатор канала интерфейса	M	
Установление связи с процессом	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.4.6.3 Сервисные параметры

6.4.6.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.4.6.3.2 Идентификатор канала интерфейса

Данный параметр идентифицирует канал связи периферийного интерфейса.

6.4.6.3.3 Установление связи с процессом

Данный параметр — идентификатор, устанавливаемый для отмены связи с процессом и определяемый пользователем при запуске сервиса, который затем отменяется.

6.4.6.3.4 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.4.6.3.5 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.4.6.3.6 Ошибка

Данный параметр указывает, что выполнено одно из следующих условий:

- канал не найден;
- связь с процессом не установлена;
- отмена установленной ранее связи невозможна;
- произошла ошибка памяти;
- выявлен сбой аппаратных средств.

6.4.6.4 Сервисная процедура

Сервис применяется для отмены связи с процессом. Если он выполнен успешно, то может быть начат новый процесс установления связи. Отмененный процесс дает «Результат (–)» и сообщение об ошибке **"Cancelled by user** (Отменено пользователем)".

6.4.7 Сервис получения статуса канала периферийного интерфейса

6.4.7.1 Содержание сервиса

Данный сервис используется для получения информации о статусе получаемых или передаваемых данных о процессе или выполнении операции.

6.4.7.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 15.

Таблица 15 — Структура параметра получения статуса канала периферийного интерфейса

Название параметра	Req	Cnf
Аргумент	M	
Идентификатор канала интерфейса	M	
Установление связи с процессом	M	
Результат (+)		S
Статус		M
Продолжение в настоящий момент		M
Результат (–)		S
Ошибка		M

6.4.7.3 Сервисные параметры

6.4.7.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.4.7.3.2 Идентификатор канала интерфейса

Данный параметр идентифицирует канал связи периферийного интерфейса.

6.4.7.3.3 Установление связи с процессом

Данный параметр — идентификатор проверки установления связи, определяемый пользователем при запуске надлежащего сервиса.

6.4.7.3.4 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.4.7.3.5 Статус

Данный параметр идентифицирует статус канала периферийного интерфейса.

6.4.7.3.6 Продолжение в настоящий момент

Данный параметр сигнализирует, продолжается ли установление связи в настоящий момент. Для процессов считывания или ввода данных он содержит ряд фактически полученных или переданных символов.

6.4.7.3.7 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.4.7.3.8 Ошибка

Данный параметр указывает, что выполнено одно из следующих условий:

- канал не найден;
- процесс установления связи не завершен;
- произошла ошибка памяти;
- выявлен сбой аппаратных средств.

6.4.7.4 Сервисная процедура

Данный сервис проверяет готовность канала к началу процесса установления связи и подтверждает его статус.

6.4.8 Сервис очистки буфера считывания

6.4.8.1 Содержание сервиса

Данный сервис используется для очистки буфера считывания канала периферийного интерфейса.

6.4.8.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 16.

Таблица 16 — Структура параметра очистки буфера считывания

Название параметра	Req	Cnf
Аргумент	M	
Индикатор канала интерфейса	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.4.8.3 Сервисные параметры

6.4.8.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.4.8.3.2 Идентификатор канала интерфейса

Данный параметр идентифицирует канал связи периферийного интерфейса.

6.4.8.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.4.8.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.4.8.3.5 Ошибка

Данный параметр указывает, что выполнено одно из следующих условий:

- канал не найден;
- произошла ошибка памяти;
- выявлен сбой аппаратных средств;
- канал занят получением данных.

6.4.8.4 Сервисная процедура

Данный сервис проверяет, занят ли канал получением данных. Если он не занят, то проводится очистка буфера считывания.

6.4.9 Сервис закрытия канала периферийного интерфейса

6.4.9.1 Содержание сервиса

Данный сервис используется для закрытия канала периферийного интерфейса.

6.4.9.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 17.

Таблица 17 — Структура параметра закрытия канала периферийного интерфейса

Название параметра	Req	Cnf
Аргумент	M	
Индикатор канала интерфейса	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.4.9.3 Сервисные параметры

6.4.9.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.4.9.3.2 Идентификатор канала интерфейса

Данный параметр идентифицирует канал связи периферийного интерфейса.

6.4.9.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.4.9.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.4.9.3.5 Ошибка

Данный параметр указывает, что выполнено одно из следующих условий:

- канал не найден;
- произошла ошибка памяти;
- выявлен сбой аппаратных средств;
- канал связи занят.

6.4.9.4 Сервисная процедура

Данный сервис проверяет, открыт ли канал связи. Если он открыт, то сервис закрывает канал периферийного интерфейса.

6.4.10 Сервис выдачи сигнала о наступлении события

6.4.10.1 Содержание сервиса

Данный сервис используется для выдачи сигнала о наступлении локального события.

6.4.10.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 18.

Таблица 18 — Структура параметра выдачи сигнала о наступлении события

Название параметра	Ind	Rsp
Аргумент	M	
Идентификатор канала интерфейса	M	
Идентификатор события	M	
Сообщение о событии	M	
Результат (+)		S
Статус		M
Результат (–)		S
Ошибка		M

6.4.10.3 Сервисные параметры

6.4.10.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.4.10.3.2 Идентификатор канала интерфейса

Данный параметр идентифицирует канал связи периферийного интерфейса. Если его значение равно нулю, то событие является особым для одного канала.

6.4.10.3.3 Идентификатор события

Данный параметр — идентификатор события.

6.4.10.3.4 Сообщение о событии

Указанный параметр возвращает сообщение, содержащее данные о структуре особого события.

Примечание — Определение идентификаторов, а также передача данных о событии в настоящем стандарте не рассматриваются. Рассмотрение таких вопросов зависит от текущего приложения.

6.4.10.3.5 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.4.10.3.6 Статус

Данный параметр идентифицирует статус обработчика события:

- обработка события закончена;
- обработка события продолжается в настоящий момент.

6.4.10.3.7 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.4.10.3.8 Ошибка

Данный параметр указывает на сообщение об ошибке:

- обработка события временно невозможна;
- обработка события с указанным идентификатором продолжается в настоящий момент;
- обработка события невозможна.

6.4.10.4 Сервисная процедура

Прежде всего данный сервис проверяет возможность обработки указанного события. Потом соответственно запускается обработчик, и возвращается статус "Обработка события продолжается в настоящий момент". Другой вариант: обработчик вызывается, заканчивает выполнение задания, и возвращается статус "Обработка события закончена". Если событие не может быть обработано, то приходит сообщение об ошибке.

6.5 Расширенные сервисы

6.5.1 Инициирование интерфейса расширенного типа

6.5.1.1 Содержание сервиса

Запрос на сервис продвигается инициированием интерфейса периферийного типа (см. 6.3.2) после инсталляции расширенного сервисного провайдера.

6.5.1.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 19.

Таблица 19 — Структура параметра инициирования интерфейса расширенного типа

Название параметра	Req	Cnf
Аргумент	M	
Название типа интерфейса	M	
Идентификатор типа интерфейса	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.5.1.3 Сервисные параметры

6.5.1.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.5.1.3.2 Название типа интерфейса

Данный параметр содержит название типа интерфейса.

6.5.1.3.3 Идентификатор типа интерфейса

Данный параметр содержит число, идентифицирующее тип интерфейса для продвижения других запросов на указанный сервис.

6.5.1.3.4 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.5.1.3.5 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.5.1.3.6 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- тип интерфейса неизвестен или недоступен;
- интерфейс указанного типа уже инициализирован;
- выявлен сбой аппаратных средств.

6.5.1.4 Сервисная процедура

Данный сервис проверяет доступность интерфейса с указанным названием типа. Если интерфейс доступен, то он готовится к применению.

6.5.2 Принятие решения об использовании интерфейса расширенного типа

Процедура и параметры данного сервиса идентичны процедуре и параметрам сервиса принятия решения об использовании интерфейса периферийного типа (см. 6.3.3).

6.5.3 Открытие канала расширенного интерфейса

6.5.3.1 Содержание сервиса

Данный сервис применяется для открытия канала связи расширенного интерфейса указанного типа и задает конфигурацию по запросу пользователя.

6.5.3.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 20.

Таблица 20 — Структура параметра открытия канала расширенного интерфейса

Название параметра	Req	Cnf
Аргумент	M	
Идентификатор типа интерфейса	M	
Идентификатор канала интерфейса	M	
Название канала интерфейса	M	
Перечень параметров конфигурации	M	
Подтвержденная ссылка на точку доступа к сервису	M	
Ссылка на событие точки доступа к сервису	M	
Результат (+)		S
Результат (-)		S
Ошибка		M

6.5.3.3 Сервисные параметры

6.5.3.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.5.3.3.2 Идентификатор типа интерфейса

Данный параметр идентифицирует тип периферийного интерфейса.

6.5.3.3.3 Идентификатор канала интерфейса

Данный параметр — назначенный идентификатор канала периферийного интерфейса.

6.5.3.3.4 Название канала интерфейса

Данный параметр идентифицирует канал связи указанного типа.

6.5.3.3.5 Перечень параметров конфигурации

Перечень параметров конфигурации зависит от типа выбранного периферийного интерфейса. Спецификация перечня в настоящем стандарте не рассматривается и описана в руководстве для указанного периферийного интерфейса.

6.5.3.3.6 Подтвержденная ссылка на точку доступа к сервису

Данный параметр — ссылка на точку доступа к специальному сервису, используемому для всех подтвержденных сервисов, принадлежащих данному каналу.

6.5.3.3.7 Ссылка на событие в точке доступа к сервису

Данный параметр — ссылка на точку доступа к специальному сервису, используемая всеми сервисами обработки событий, принадлежащих данному каналу.

6.5.3.3.8 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.5.3.3.9 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.5.3.3.10 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- интерфейс указанного типа недоступен;
- канал с таким названием недоступен;
- канал уже открыт;
- отсутствует или неверно название канала;
- отсутствует или неверна ссылка на подтвержденную точку доступа к сервису;
- отсутствует или неверна ссылка на точку доступа к сервису обработки событий;
- произошла ошибка памяти;
- выявлен сбой аппаратных средств;
- произошла блокировка конфигурации канала;
- недостаточно ресурсов для открытия данного канала;
- не указан параметр ошибки;
- указан параметр ошибки;
- выявлена любая другая ошибка.

6.5.3.4 Сервисная процедура

Данный сервис проверяет все параметры и при отсутствии ошибки делает попытку открыть указанный канал связи. Если он не готов к работе, то приходит сообщение об ошибке.

6.5.4 Реконфигурация канала расширенного интерфейса

Процедура и параметры данного сервиса идентичны процедуре и параметрам сервиса реконфигурации канала периферийного интерфейса (см. 6.4.2).

6.5.5 Считывание данных расширенного интерфейса

Процедура и параметры данного сервиса идентичны процедуре и параметрам сервиса считывания данных (см. 6.4.3).

6.5.6 Ввод данных расширенного интерфейса

Процедура и параметры данного сервиса идентичны процедуре и параметрам сервиса ввода данных (см. 6.4.4).

6.5.7 Выполнение операции расширенного интерфейса

Процедура и параметры данного сервиса идентичны процедуре и параметрам сервиса выполнения операции (см. 6.4.5).

6.5.8 Отмена установления расширенной связи с процессом

Процедура и параметры данного сервиса идентичны процедуре и параметрам сервиса отмены установления связи с процессом (см. 6.4.6).

6.5.9 Получение статуса канала расширенного интерфейса

Процедура и параметры данного сервиса идентичны процедуре и параметрам сервиса получения статуса канала периферийного интерфейса (см. 6.4.7).

6.5.10 Очистка буфера считывания расширенного интерфейса

Процедура и параметры данного сервиса идентичны процедуре и параметрам сервиса очистки буфера считывания (см. 6.4.8).

6.5.11 Закрытие канала расширенного интерфейса

Процедура и параметры данного сервиса идентичны процедуре и параметрам сервиса закрытия буфера периферийного интерфейса (см. 6.4.9).

6.5.12 Выдача сигнала о расширенном событии

Процедура и параметры данного сервиса идентичны процедуре и параметрам сервиса выдачи сигнала о наступлении события (см. 6.4.10).

6.6 Сервис поддержки обслуживания

6.6.1 Выделение памяти

6.6.1.1 Содержание сервиса

Сервис используется для получения ссылки на согласованный объем данных указанного размера.

6.6.1.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 21.

Таблица 21 — Структура параметра выделения памяти

Название параметра	Req	Cnf
Аргумент	M	
Число восьмибитовых байтов в объеме данных	M	
Результат (+)		S
Ссылка на выделенный объем данных		M
Результат (–)		S

6.6.1.3 Сервисные параметры

6.6.1.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.1.3.2 Число восьмибитовых байтов в объеме данных

Данный параметр указывает размер запрошенного объема данных в восьмибитовых байтах.

6.6.1.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.6.1.3.4 Ссылка на выделенный объем данных

Данный параметр содержит ссылку на идентифицированный запрошенный объем данных.

6.6.1.3.5 Результат (–)

Указанный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.1.4 Сервисная процедура

Если согласованная область данных указанного размера доступна, то создается ссылка на нее. В противном случае запрос снимается.

6.6.2 Перевыделение памяти

6.6.2.1 Содержание сервиса

Данный сервис используется для изменения размера предварительно выделенного объема памяти.

6.6.2.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 22.

Таблица 22 — Структура параметра перевыделения памяти

Название параметра	Req	Cnf
Аргумент	M	
Ссылка на выделенный объем памяти	M	
Число восьмибитовых байтов в перевыделенном объеме памяти	M	
Результат (+)		S
Ссылка на перевыделенный объем памяти		M
Результат (–)		S

6.6.2.3 Сервисные параметры

6.6.2.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.2.3.2 Ссылка на выделенный объем памяти

Указанный параметр — ссылка на объем данных, размер которых необходимо изменить. Это должен быть результат сервиса либо выделения, либо перевыделения памяти.

6.6.2.3.3 Число восьмибитовых байтов в перевыделенном объеме данных

Указанный параметр указывает новый размер запрошенного объема данных в восьмибитовых байтах.

6.6.2.3.4 Результат (+)

Параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.6.2.3.5 Сервисная ссылка

Указанный параметр содержит ссылку на идентифицированный запрошенный объем данных.

6.6.2.3.6 Результат (–)

Параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.2.4 Сервисная процедура

Если общий объем памяти позволяет изменить указанный объем данных, то создается ссылка на новый объем. Содержимое наименьшей общей части старого и нового объемов памяти должно быть неизменным. Если их изменить невозможно, то запрос снимается, и старый объем сохраняется.

6.6.3 Освобождение памяти

6.6.3.1 Содержание сервиса

Сервис используется для получения ссылки на согласованный объем данных указанного размера.

6.6.3.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 23.

Таблица 23 — Структура параметра освобождения памяти

Название параметра	Req	Cnf
Аргумент	M	
Ссылка на выделенный объем данных	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.6.3.3 Сервисные параметры

6.6.3.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.3.3.2 Ссылка на выделенный объем данных

Указанный параметр — ссылка на объем данных, который необходимо открыть. Это должен быть результат сервиса либо выделения, либо перевыделения памяти.

6.6.3.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.6.3.3.4 Сервисная ссылка

Данный параметр содержит ссылку на идентифицированный запрошенный объем данных.

6.6.3.3.5 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.3.3.6 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- неверная ссылка на выделенный объем данных;
- любая другая ошибка.

6.6.3.4 Сервисная процедура

Если указанная ссылка для выделенного объема данных правильна, то область памяти открывается и освобождается для ее любого другого выделения. В противном случае сохраняется старый объем данных, и приходит сообщение об ошибке.

6.6.4 Определение времени

6.6.4.1 Содержание сервиса

Данный сервис используется для получения информации о времени в различных форматах по отношению к Всемирному координированному времени (по Гринвичу).

6.6.4.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 24.

Таблица 24 — Структура параметра определения времени

Название параметра	Req	Cnf
Аргумент	M	
Требуемый формат времени	M	
Результат (+)		S
Перечень характерных элементов времени		M
Результат (–)		S
Ошибка		M

6.6.4.3 Сервисные параметры

6.6.4.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.4.3.2 Требуемый формат времени

Данный параметр указывает требуемый формат времени в следующем перечне:

- время UNIX: количество секунд, прошедших с 00:00 часов 1 января 1970 года, единицы времени — секунды и микросекунды по Гринвичу, начиная с последней секунды;
- абсолютное время: единицы времени для лет, месяцев, дня месяца, часа, минуты, секунд, миллисекунд, микросекунд и наносекунд по Всемирному координированному времени с дополнительными элементами, учитывающими отличие от местного времени в секундах.

6.6.4.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.6.4.3.4 Перечень характерных элементов времени

Данный параметр содержит элементы структуры времени в соответствии с его требуемым форматом.

6.6.4.3.5 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.4.3.6 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- информация о времени не может быть получена;
- выявлена любая другая ошибка.

6.6.4.4 Сервисная процедура

Всемирное координированное время уточняется в ходе RMS. Если уточнение невозможно, то приходит сообщение об ошибке.

6.6.5 Определение времени обработки

6.6.5.1 Содержание сервиса

Данный сервис дает показание счетчика микросекунд RMS. Время его запуска не определяется.

6.6.5.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 25.

Таблица 25 — Структура параметра определения времени обработки

Название параметра	Req	Cnf
Результат (+)		S
Локальное время обработки		M
Результат (–)		S
Ошибка		M

6.6.5.3 Сервисные параметры

6.6.5.3.1 Аргумент

Данный сервис не имеет параметров.

6.6.5.3.2 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.6.5.3.3 Локальное время обработки

Данный параметр содержит значение счетчика микросекунд RMS.

6.6.5.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.5.3.5 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- информация о локальном времени обработки не может быть получена;
- выявлена любая другая ошибка.

6.6.5.4 Сервисная процедура

Система сервисного управления ресурсами RMS должна иметь счетчик микросекунд, который запускается когда она доступна. Начальное показание не определено. Данный сервис предоставляет информацию о фактическом показании счетчика.

6.6.6 Ожидание

6.6.6.1 Содержание сервиса

Данный сервис создает задержку по времени заданной продолжительности.

6.6.6.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 26.

Таблица 26 — Структура параметра ожидания

Название параметра	Req	Cnf
Аргумент	M	
Время ожидания	M	
Результат (+)		S
Результат (-)		S
Ошибка		M

6.6.6.3 Сервисные параметры

6.6.6.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.6.3.2 Время ожидания

Данный параметр указывает задержку по времени в миллисекундах.

6.6.6.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.6.6.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.6.3.5 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- замедление не выполнено, сервисная услуга закончена немедленно;
- выявлена любая другая ошибка.

6.6.6.4 Сервисная процедура

Если ресурсы определения времени доступны, то данный сервис ожидает в течение указанного времени. В противном случае приходит сообщение об ошибке.

6.6.7 Создание таймера

6.6.7.1 Содержание сервиса

Данный сервис создает и запускает таймер RMS, а также инсталлирует точку доступа к сервису для обработки событий таймера.

6.6.7.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 27.

Таблица 27 — Структура параметра создания таймера

Название параметра	Req	Cnf
Аргумент	M	
Ссылка на точку доступа таймера	M	
Таймер управления процессом	M	
Продолжительность	M	
Количество событий	M	
Результат (+)		S
Идентификатор таймера		M
Результат (–)		S
Ошибка		M

6.6.7.3 Сервисные параметры

6.6.7.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.7.3.2 Ссылка на точку доступа таймера

Данный параметр — ссылка на точку доступа специального сервиса, используемого для обработки событий, принадлежащих указанному таймеру.

6.6.7.3.3 Таймер управления процессом

Данный параметр — порядок обработки, определяемый пользователем для идентификации событий указанного таймера. Данные события подвергаются обработке.

6.6.7.3.4 Продолжительность

Данный параметр — время, протекшее до выдачи таймером сигнала о наступлении события.

6.6.7.3.5 Количество событий

Таймеры могут работать в режиме повторений. Данный параметр назначает их количество. Нулевое значение указывает на повторяющийся таймер, который останавливается только после его удаления.

6.6.7.3.6 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.6.7.3.7 Идентификатор таймера

Данный параметр идентифицирует созданный таймер. Идентификатор необходим для его удаления.

6.6.7.3.8 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.7.3.9 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- ресурс для таймеров исчерпан;
- выявлена любая другая ошибка.

6.6.7.4 Сервисная процедура

Если соответствующие ресурсы доступны, то таймер создается и запускается. Если время ожидания исчерпано, то он выдает сигнал о наступлении события. Если указанное количество событий больше одного, то таймер запускается снова. Если их зафиксированное число совпадает с указанным количеством, то он удаляется, и имеющиеся ресурсы становятся доступными для другого таймера. Если количество событий равно нулю, то таймер работает в режиме повторения и может быть остановлен только с помощью сервиса его удаления.

Точность таймера зависит от ресурсов RMS и должна быть максимально возможной.

6.6.8 Выдача таймером сигнала о наступлении события

6.6.8.1 Содержание сервиса

Данный сервис выдает сигнал об окончании интервала времени и формирует примитив индикации.

6.6.8.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 28.

Т а б л и ц а 28 — Структура параметра выдачи таймером сигнала о наступлении события

Название параметра	Ind	Rsp
Аргумент	M	
Таймер управления процессом	M	
Статус таймера	M	
Результат (+)		S
Результат (–)		S
Код ошибки		C

6.6.8.3 Сервисные параметры

6.6.8.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.8.3.2 Таймер управления процессом

Данный параметр определяет порядок обработки информации, устанавливаемый пользователем для идентификации таймера, выдающего сигнал о наступлении указанного события. Это — параметр сервиса создания таймера.

6.6.8.3.3 Статус таймера

Данный параметр указывает, что предшествующее событие, зафиксированное таймером, все еще ждет обработки до получения примитива отклика.

6.6.8.3.4 Результат (+)

Данный параметр указывает, что сервисная индикация выполнена успешно, и событие больше не ждет обработки.

6.6.8.3.5 Результат (–)

Данный параметр указывает, что сервисная индикация не может быть получена (обработана), и информации о событии не ожидается.

6.6.8.3.6 Код ошибки

Данный параметр указывает причину ошибки.

6.6.8.4 Сервисная процедура

Если время ожидания таймера истекло, то сигнал о наступлении события подается пользователю в виде примитива индикации. Проверяется факт ожидания обработки предшествующего события и статус таймера.

6.6.9 Удаление таймера

6.6.9.1 Содержание сервиса

Данный сервис останавливает работу таймера в RMS, удаляет его и освобождает все связанные ресурсы.

6.6.9.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 29.

Т а б л и ц а 29 — Структура параметра удаление таймера

Название параметра	Req	Cnf
Аргумент	M	
Идентификатор таймера	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.6.9.3 Сервисные параметры

6.6.9.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.9.3.2 Идентификатор таймера

Данный параметр — идентификатор таймера, возвращаемый сервисом его создания.

6.6.9.3.3 Результат (+)

Данный параметр указывает, что запрос на сервис выполнен успешно и таймер удален.

6.6.9.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.9.3.5 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- идентификатор таймера неправильный;
- операция не выполнена, таймер занят обработкой события;
- выявлена любая другая ошибка.

6.6.9.4 Сервисная процедура

Если у таймера нет событий, ждущих обработки, то он останавливается и удаляется. В противном случае приходит сообщение об ошибке.

6.6.10 Создание таймера со световой индикацией

6.6.10.1 Содержание сервиса

Данный сервис создает и запускает таймер со световой индикацией в RMS, а также инсталлирует точку доступа к сервису для обработки событий таймера.

6.6.10.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 30.

Таблица 30 — Структура параметра создания таймера со световой индикацией

Название параметра	Req	Cnf
Аргумент	M	
Ссылка на точку доступа таймера	M	
Таймер управления процессом	M	
Продолжительность	M	
Количество событий	M	
Результат (+)		S
Идентификатор таймера		M
Результат (–)		S
Ошибка		M

6.6.10.3 Сервисные параметры

6.6.10.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.10.3.2 Ссылка на точку доступа к таймеру

Данный параметр — ссылка на точку доступа специального сервиса. Используется для обработки событий, принадлежащих данному таймеру.

6.6.10.3.3 Таймер управления процессом

Данный параметр задает порядок обработки информации, определяемый пользователем. Служит для идентификации событий таймера и их обработки.

6.6.10.3.4 Продолжительность

Данный параметр — время, протекшее до выдачи таймером сигнала о наступлении события.

6.6.10.3.5 Количество событий

Таймеры могут работать в режиме повторений. Данный параметр назначает их количество. Нулевое значение указывает, что таймер повторяющийся и может быть остановлен только его удалением.

6.6.10.3.6 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.6.10.3.7 Идентификатор таймера

Данный параметр идентифицирует созданный таймер. Идентификатор необходим для его удаления.

6.6.10.3.8 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.10.3.9 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- ресурс для процессов световой индикации исчерпан;
- выявлена любая другая ошибка.

6.6.10.4 Сервисная процедура

Если соответствующие ресурсы доступны, то создается новый процесс световой индикации, и запускается ассоциированный таймер. По истечении заданного интервала времени он выдает сигнал о наступлении события. Если указанное количество событий больше одного, то таймер запускается снова. Если число сигналов совпадает с указанным количеством событий, то таймер удаляется, и процесс световой индикации прекращается. Если указанное количество событий равно нулю, то таймер работает в режиме повторения и может быть остановлен только с помощью сервиса его удаления.

Точность таймера зависит от диспетчеризации процесса индикации.

6.6.11 Сигнал светового таймера о наступлении события

6.6.11.1 Содержание сервиса

Интервал ожидания исчерпан, и процесс индикации, ассоциированный с данным таймером, выдает ее примитив.

6.6.11.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 31.

Таблица 31 — Структура параметра сигнала светового таймера о наступлении события

Название параметра	Ind	Rsp
Аргумент	M	
Таймер управления процессом	M	
Статус таймера	C	
Результат (+)		M

6.6.11.3 Сервисные параметры

6.6.11.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.11.3.2 Таймер управления процессом

Данный параметр задает порядок обработки информации, определяемый пользователем для идентификации таймера, выдающего сигнал о наступлении указанного события. Это параметр сервиса создания таймера со световой индикацией.

6.6.11.3.3 Статус таймера

Данный параметр применяется, если процесс световой индикации может быть использован для управления временем в абсолютных единицах. Таймер фиксирует, что время обработки предшествующего события превысило установленный предел.

6.6.11.3.4 Результат (+)

Данный параметр указывает, что сервисная индикация выполнена успешно, и рассматриваемое событие не требует обработки.

6.6.11.4 Сервисная процедура

Если время ожидания таймера исчерпано, то пользователю выдается сигнал о наступлении события с помощью примитива индикации. По возможности проверяется факт превышения допустимого времени обработки предшествующего события.

6.6.12 Удаление таймера со световой индикацией

6.6.12.1 Содержание сервиса

Данный сервис останавливает отслеживание параметров световой индикации, ассоциированных с указанным таймером, и освобождает все связанные ресурсы.

6.6.12.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 32.

Таблица 32 — Структура параметра удаления таймера со световой индикацией

Название параметра	Req	Cnf
Аргумент	M	
Идентификатор таймера	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.6.12.3 Сервисные параметры

6.6.12.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.12.3.2 Идентификатор таймера

Данный параметр — идентификатор таймера, который формируется сервисом его создания.

6.6.12.3.3 Результат (+)

Данный параметр указывает, что запрос на сервис выполнен успешно, и таймер удален.

6.6.12.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.12.3.5 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- идентификатор таймера неправильный;
- операция не выполнена, таймер занят обработкой события;
- выявлена любая другая ошибка.

6.6.12.4 Сервисная процедура

Если таймер не занят обработкой события, то процесс световой индикации останавливается, и связанный ресурс освобождается. В противном случае приходит сообщение об ошибке.

6.6.13 Идентификация процесса световой индикации

6.6.13.1 Содержание сервиса

Данный сервис идентифицирует процесс световой индикации по поступившему запросу.

6.6.13.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 33.

Таблица 33 — Структура параметра идентификации процесса световой индикации

Название параметра	Req	Cnf
Результат (+)		S
Идентификатор процесса световой индикации		M
Результат (–)		S
Ошибка		M

6.6.13.3 Сервисные параметры

6.6.13.3.1 Аргумент

Запрос на данный сервис не имеет параметров.

6.6.13.3.2 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.6.13.3.3 Идентификатор процесса световой индикации

Данный параметр содержит уникальный идентификатор процесса световой индикации.

6.6.13.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.13.3.5 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- процесс световой индикации не может быть идентифицирован;
- выялена любая другая ошибка.

6.6.13.4 Сервисная процедура

Данный сервис дает информацию об идентификаторе текущего процесса световой индикации по запросу пользователя. Если это невозможно, то приходит сообщение об ошибке.

6.6.14 Создание вычислительного семафора

6.6.14.1 Содержание сервиса

Данный сервис задает порядок доступа к ресурсам. Число доступов к ним в одно и то же время ограничено. Количество использования ресурса в одно и то же время определяется запросом на сервис.

6.6.14.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 34.

Таблица 34 — Структура параметра создания вычислительного семафора

Название параметра	Req	Cnf
Аргумент	M	
Количество одновременного использования ресурса	M	
Результат (+)		S
Обработка сигнала семафора		M
Результат (–)		S
Ошибка		M

6.6.14.3 Сервисные параметры

6.6.14.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.14.3.2 Количество одновременного использования ресурса

Данный параметр показывает, какое количество пользователей имеют одновременный доступ к ресурсу.

6.6.14.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.6.14.3.4 Обработка сигнала семафора

Данный параметр задает порядок доступа к защищенному ресурсу.

6.6.14.3.5 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.14.3.6 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- количество пользователей, одновременно обращающихся к ресурсу, указано неверно;
- семафор не работает;
- выявлена любая другая ошибка.

6.6.14.4 Сервисная процедура

Проверяется количество пользователей, одновременно обращающихся к данному ресурсу, и доступ к нему. Создается объект управления, загружается ассоциированный счетчик с заданным максимальным количеством пользователей, одновременно обращающихся к ресурсу, возвращается результат обработки информации. В противном случае приходит сообщение об ошибке.

6.6.15 Ожидание вычислительного семафора

6.6.15.1 Содержание сервиса

Данный сервис регулирует доступ к ресурсу, защищенному семафором, по запросу. Если доступ не разрешен, то он откладывается на некоторое время.

6.6.15.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 35.

Таблица 35 — Структура параметра ожидания вычислительного семафора

Название параметра	Req	Cnf
Аргумент	M	
Обработка сигнала семафора	M	
Максимальное время ожидания	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.6.15.3 Сервисные параметры

6.6.15.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.15.3.2 Обработка сигнала семафора

Данный параметр характеризует вычислительный семафор, защищающий ресурс. Результат обработки информации возвращается сервисом создания вычислительного семафора.

6.6.15.3.3 Максимальное время ожидания

Это максимальное время отсрочки доступа к защищенному ресурсу.

6.6.15.3.4 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно, и защищенный ресурс доступен.

6.6.15.3.5 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.15.3.6 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- сигнал семафора обработан неверно;
- максимальное время ожидания превысило допуск;
- выялена любая другая ошибка.

6.6.15.4 Сервисная процедура

Если сигнал семафора обработан верно, и показание ассоциированного счетчика больше нуля, то оно уменьшается на единицу, и предоставляется доступ к защищенному ресурсу. В противном случае доступ откладывается до момента, когда показание счетчика больше нуля, или установленное время ожидания превышает допуск, тогда приходит сообщение об ошибке.

6.6.16 Открытие вычислительного семафора

6.6.16.1 Содержание сервиса

С помощью данного сервиса открывается доступ к ресурсу, защищенному семафором.

6.6.16.2 Структура сервисного параметра

Сервисные параметры данного сервиса приведены в таблице 36.

Таблица 36 — Структура параметра открытия вычислительного семафора

Название параметра	Req	Cnf
Аргумент	M	
Обработка сигнала семафора	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.6.16.3 Сервисные параметры

6.6.16.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.16.3.2 Обработка сигнала семафора

Данный параметр задает порядок обработки информации вычислительным семафором, защищающим ресурс. Указанный порядок возвращается сервисом создания такого семафора.

6.6.16.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно. Об открытии защищенного ресурса присылается уведомление.

6.6.16.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.16.3.5 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- неправильно обработан сигнал семафора;
- новое открытие ресурса невозможно;
- выявлена любая другая ошибка.

6.6.16.4 Сервисная процедура

Если сигнал семафора обработан верно, и текущее показание ассоциированного счетчика меньше начального, то оно уменьшается на единицу. В противном случае приходит сообщение об ошибке.

6.6.17 Стирание вычислительного семафора

6.6.17.1 Содержание сервиса

Данный сервис стирает вычислительный семафор и освобождает все ресурсы, связанные с ним.

6.6.17.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 37.

Таблица 37 — Структура параметра стирания вычислительного семафора

Название параметра	Req	Cnf
Аргумент	M	
Обработка сигнала семафора	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.6.17.3 Сервисные параметры

6.6.17.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.17.3.2 Обработка сигнала семафора

Данный параметр задает порядок обработки сигнала вычислительного семафора, защищающего ресурс. Указанный порядок возвращается сервисом создания такого семафора.

6.6.17.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно, и семафор стерт.

6.6.17.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.17.3.5 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- неправильно обработан сигнал семафора;
- защищенный ресурс все еще доступен;
- выявлена любая другая ошибка.

6.6.17.4 Сервисная процедура

Если сигнал семафора обработан верно, и защищенный ресурс недоступен, то семафор стирается. В противном случае приходит сообщение об ошибке.

6.6.18 Создание частного семафора

6.6.18.1 Содержание сервиса

Данный сервис определяет порядок управления ресурсом, который может быть использован неограниченно с помощью процесса световой индикации, овладевающего ресурсом, при получении доступа. Никакой другой процесс световой индикации не может получить доступ к ресурсу до тех пор, пока он занят указанным процессом.

6.6.18.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 38.

Таблица 38 — Структура параметра создания частного семафора

Название параметра	Req	Cnf
Результат (+)		S
Обработка сигнала семафора		M
Результат (–)		S
Ошибка		M

6.6.18.3 Сервисные параметры

6.6.18.3.1 Аргумент

Данный запрос на сервис не имеет параметров.

6.6.18.3.2 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.6.18.3.3 Обработка сигнала семафора

Данный параметр управляет доступом к защищенному ресурсу.

6.6.18.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.18.3.5 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- управление семафором недоступно;
- выявлена любая другая ошибка.

6.6.18.4 Сервисная процедура

Если имеются ресурсы для управления доступом, то создается объект управления, и возвращается порядок обработки информации. В противном случае приходит сообщение об ошибке.

6.6.19 Ожидание частного семафора

6.6.19.1 Содержание сервиса

С помощью данного сервиса можно запросить доступ к ресурсу, защищенному указанным семафором. Доступ всегда разрешен для процесса световой индикации, приватизировавшего ресурс. Если доступ запрашивается другим процессом световой индикации, то он откладывается до момента превышения указанного времени ожидания или открытия ресурса приватизирующим процессом.

6.6.19.2 Структура сервисного параметра

Сервисные параметры данного сервиса приведены в таблице 39.

Таблица 39 — Структура параметра ожидания частного семафора

Название параметра	Req	Cnf
Аргумент	M	
Обработка сигнала семафора	M	
Максимальное время ожидания	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.6.19.3 Сервисные параметры

6.6.19.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.19.3.2 Обработка сигнала семафора

Данный параметр задает порядок обработки информации частным семафором, защищающим ресурс, и возвращается сервисом его создания.

6.6.19.3.3 Максимальное время ожидания

Это максимальное время отсрочки доступа к защищенному ресурсу.

6.6.19.3.4 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно, и может быть получен доступ к защищенному ресурсу.

6.6.19.3.5 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.19.3.6 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- неправильно обработан сигнал семафора;
- максимальное время ожидания превысило допуск;
- выявлена любая другая ошибка.

6.6.19.4 Сервисная процедура

Если порядок обработки сигнала семафора неверен, и ресурс приватизирован данным процессом световой индикации или не приватизирован никаким световым процессом, то доступ к защищенному ресурсу гарантирован. Если ресурс приватизирован другим процессом световой индикации, то доступ к нему откладывается до момента его полного освобождения или превышения указанного времени ожидания. Если оно превысило допустимое значение, то приходит сообщение об ошибке.

Если рассматриваемый доступ является первым для данного процесса световой индикации, то он приватизирует ресурс.

6.6.20 Открытие частного семафора

6.6.20.1 Содержание сервиса

С помощью данного сервиса можно открыть доступ к ресурсу, защищенному указанным семафором. Право собственности процесса световой индикации на ресурс заканчивается, если количество открытий ресурса указанным сервисом равно количеству отказов в доступе, выданных сервисом ожидания частного семафора.

6.6.20.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 40.

Таблица 40 — Структура параметра открытия частного семафора

Название параметра	Req	Cnf
Аргумент	M	
Обработка сигнала семафора	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.6.20.3 Сервисные параметры

6.6.20.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.20.3.2 Обработка сигнала семафора

Данный параметр задает порядок обработки сигнала вычислительного семафора, защищающего ресурс, и возвращается сервисом создания частного семафора.

6.6.20.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно, и получено уведомление об открытии защищенного ресурса.

6.6.20.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.20.3.5 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- неправильно обработан сигнал семафора;
- новое открытие доступа невозможно;
- выявлена любая другая ошибка.

6.6.20.4 Сервисная процедура

Если сигнал семафора обработан правильно, то приходит уведомление об открытии ресурса, в противном случае — сообщение об ошибке.

Проверяется соответствие сервиса ожидания частного семафора и сервиса его открытия. Доступ к защищенному ресурсу может быть получен другим процессом световой индикации, если число этих двух запросов совпадает.

6.6.21 Стирание частного семафора

6.6.21.1 Содержание сервиса

Данный сервис стирает частный семафор и освобождает все выделенные для него ресурсы.

6.6.21.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 41.

Таблица 41 — Структура параметра стирания частного семафора

Название параметра	Req	Cnf
Аргумент	M	
Обработка сигнала семафора	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.6.21.3 Сервисные параметры

6.6.21.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.21.3.2 Обработка сигнала семафора

Данный параметр задает порядок обработки сигнала частного семафора, защищающего ресурс, и возвращается сервисом его создания.

6.6.21.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно, и семафор стерт.

6.6.21.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.21.3.5 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- неправильно обработан сигнал семафора;
- защищенный ресурс все еще приватизирован;
- выявлена любая другая ошибка.

6.6.21.4 Сервисная процедура

Если сигнал семафора обработан верно, и защищенный ресурс не приватизирован процессом световой индикации, то семафор стирается. В противном случае приходит сообщение об ошибке.

6.6.22 Открытие журнала отладки

6.6.22.1 Содержание сервиса

Данный сервис открывает канал регистрации сообщений отладки.

6.6.22.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 42.

Таблица 42 — Структура параметра открытия журнала отладки

Название параметра	Req	Cnf
Аргумент	M	
Отладка идентификатора канала	M	
Результат (+)		S
Порядок регистрации		M
Результат (–)		S
Ошибка		M

6.6.22.3 Сервисные параметры

6.6.22.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.22.3.2 Отладка идентификатора канала

Данный параметр идентифицирует канал отладки, определяемый RMS-пользователем. Ресурсы канала назначаются RMS-провайдером.

6.6.22.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно.

6.6.22.3.4 Порядок регистрации

Данный параметр задает порядок обработки последующего доступа к каналу отладки.

6.6.22.3.5 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.22.3.6 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- новые каналы отладки недоступны;
- неправильно выбран идентификатор;
- выявлена любая другая ошибка

6.6.22.4 Сервисная процедура

Если канал отладки доступен, и идентификатор правильный (не задан), то создается новый канал для формирования потока текстовых сообщений. В противном случае приходит сообщение об ошибке.

6.6.23 Ввод сообщения об отладке

6.6.23.1 Содержание сервиса

Данный сервис оформляет сообщение в канале отладки.

6.6.23.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 43.

Таблица 43 — Структура параметра ввода сообщения отладки

Название параметра	Req	Cnf
Аргумент	M	
Порядок регистрации	M	
Сообщение об отладке	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.6.23.3 Сервисные параметры

6.6.23.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.23.3.2 Порядок регистрации

Данный параметр задает порядок обработки сообщения в канале отладки. Значение параметра возвращается сервисом открытия журнала отладки.

6.6.23.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно, и сообщение принято.

6.6.23.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.23.3.5 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- неправильно установлен порядок регистрации;
- канал отладки временно недоступен;
- возможности канала отладки исчерпаны;
- выявлена любая другая ошибка.

6.6.23.4 Сервисная процедура

Если регистрация выполнена верно, и канал отладки готов принять сообщение, то оно регистрируется в журнале отладки. В противном случае приходит сообщение об ошибке.

6.6.24 Закрытие журнала отладки

6.6.24.1 Содержание сервиса

Данный сервис закрывает канал отладки.

6.6.24.2 Структура сервисного параметра

Параметры данного сервиса приведены в таблице 44.

Таблица 44 — Структура параметра закрытия журнала отладки

Название параметра	Req	Cnf
Аргумент	M	
Порядок регистрации	M	
Результат (+)		S
Результат (–)		S
Ошибка		M

6.6.24.3 Сервисные параметры

6.6.24.3.1 Аргумент

Аргумент содержит значение параметра запроса на сервис.

6.6.24.3.2 Порядок регистрации

Данный параметр задает порядок обработки канала отладки. Значение параметра возвращается сервисом открытия журнала отладки.

6.6.24.3.3 Результат (+)

Данный параметр селекционного типа указывает, что запрос на сервис выполнен успешно, и канал отладки закрыт.

6.6.24.3.4 Результат (–)

Данный параметр селекционного типа указывает, что запрос на сервис не выполнен.

6.6.24.3.5 Ошибка

Данный параметр указывает на сообщение об ошибке из следующего перечня:

- неправильно установлен порядок регистрации;

- канал отладки занят;
- выявлена любая другая ошибка.

6.6.24.4 Сервисная процедура

Если регистрация верна, и канал отладки не занят, то он закрывается. В противном случае приходит сообщение об ошибке.

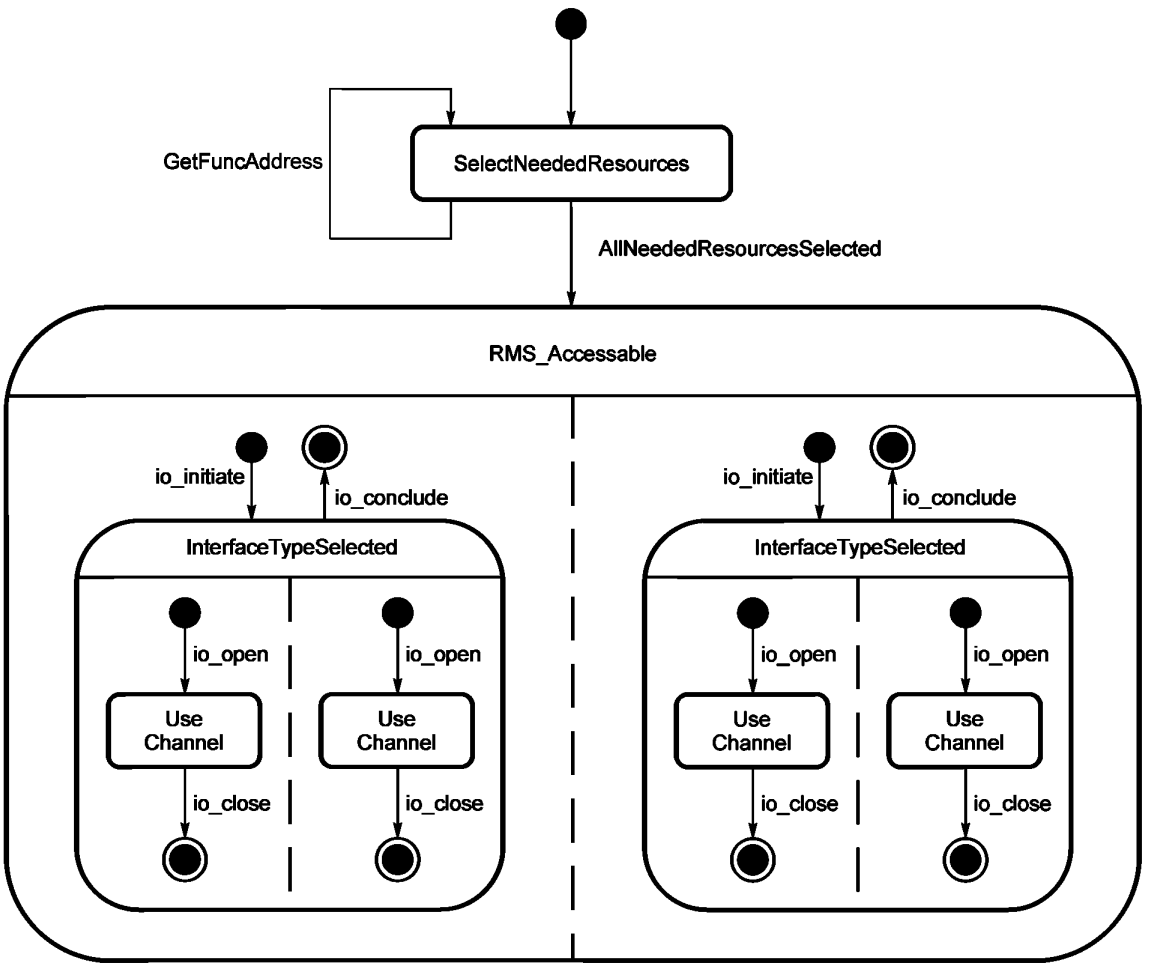
6.7 Устройство управления RMS-состоянием

6.7.1 Изменение состояния

Переход от предшествующего состояния к последующему выполняется с помощью сервисов, блокировок и процессов завершения выполнения задания. Изменению состояния препятствуют сервисные ошибки, которые в нижеследующей диаграмме состояний не показаны.

6.7.2 Диаграмма состояний

Интерфейс сервисного управления ресурсами RMSI может поддерживать различные версии одинаковых сервисов. Перед запуском сервиса необходимо установить номер ее версии. Пользователь должен выбрать сервисы, необходимые для рассматриваемого приложения, которое, в свою очередь, может быть только частью всего процесса управления ресурсами RMS (см. рисунок 4).



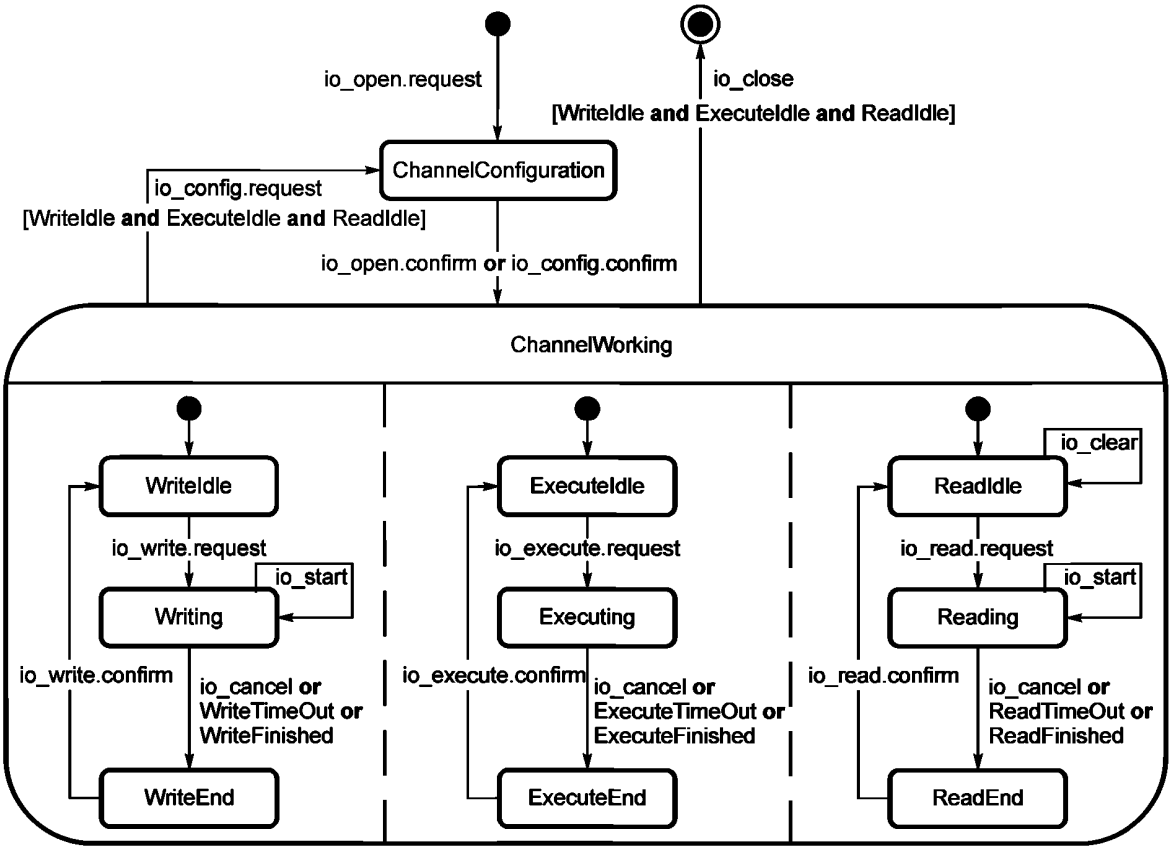
GetFuncAddress — состояние получения функционального адреса; SelectNeededResources — состояние выбора необходимого ресурса; AllNeededResourcesSelected — состояние, когда все необходимые ресурсы выбраны; RMS Accessable — состояние доступности сервисного управления ресурсами; io_initiate — сервис создания объекта для интерфейса; io_conclude — сервис подтверждения о завершении операции ввода/вывода; InterfaceTypeSelected — состояние выбора типа интерфейса; io_open — сервис создания объекта для канала связи; Use Channel — состояние использования канала; io_close — сервис закрытия доступа к каналу.

Рисунок 4 — Диаграмма RMS-состояний с несколькими периферийными интерфейсами

Интерфейс RMSI может поддерживать любое количество разных типов периферийных интерфейсов, которые, в свою очередь, могут иметь любое количество каналов связи. Сервис **io_initiate** создает объект для интерфейса заданного типа, а **io_open** — для канала связи. Следовательно, объекты интерфейсов различного типа могут находиться в разных состояниях, так же как и объекты различных каналов интерфейса одного типа. На рисунке 4 показан пример с двумя областями и произвольным количеством объектов для интерфейса указанного типа и канала.

6.7.3 Диаграмма состояний канала

Для обеспечения обмена данными с периферийными интерфейсами (выполнения рабочих процедур с периферийными компонентами) необходимо учитывать время выполнения запроса, наличие блокировки и возможность отмены процесса переработки информации. На рисунке 5 дана диаграмма состояний на унифицированном языке моделирования UML, содержащая подробности указанных процедур.



io_open request — Запрос на открытие канала; io_close — Закрытие канала; WriteIdle and ExecuteIdle and ReadIdle — Ввод, выполнение операции и считывание на свободном канале; Channel configuration — Выбор конфигурации канала; io_config request — Запрос на конфигурирование; io_open confirm or io_config confirm — Подтверждение открытия или конфигурирования; Channel working — Рабочее состояние канала; io_clear — Очистка; WriteIdle — Выполнение ввода в свободном канале; ExecuteIdle — Выполнение операции в свободном канале; ReadIdle — Считывание в свободном канале; io_write request — Запрос на ввод; io_execute request — Запрос на выполнение операции; io_read request — Запрос на считывание; io_start — Сервис подтверждения объема переданной информации; Writing — Состояние ввода; Executing — Состояние выполнения; Reading — Состояние считывания; io_write confirm — Подтверждение ввода; io_execute confirm — Подтверждение выполнения; io_read confirm — Подтверждение считывания; io_cancel or WritetimeOut or WriteFinished — Отмена ввода/вывода, или блокировка ввода, или завершение ввода; io_cancel or ExecuteTimeOut orExecuteFinished — Отмена ввода/вывода, или блокировка выполнения, или завершение выполнения; io_cancel or ReadTimeOut or ReadFinished — Отмена ввода/вывода, или блокировка считывания, или завершение считывания; WriteEnd — Окончание ввода; ExecuteEnd — Окончание выполнения; ReadEnd — Окончание считывания.

Рисунок 5 — Диаграмма состояний RMS-канала и сервис ввода/вывода

Расширенные сервисы ввода/вывода и их сопряжение приводят к появлению особых состояний соответствующих процедур RMS. Эти сервисы непосредственно связаны с расширенными сервисами. Процесс изменения состояния может быть описан только вместе с сервисом ввода/вывода.

6.7.4 Состояние интерфейса RMSI

6.7.4.1 Состояние выбора необходимого ресурса

Интерфейс RMSI находится в исходном состоянии, доступен только сервис получения функционального адреса, который выбирает другие необходимые сервисы по названию и номеру версии. По окончании выбора такой сервис доступен RMS-пользователю.

6.7.4.2 Состояние RMS_accessible

Данное состояние вводится пользователем интерфейса RMSI, если выбраны все необходимые ресурсы. Изменение состояния происходит при первом доступе к сервису, отличному от сервиса получения функционального адреса.

6.7.4.3 Состояние выбора типа интерфейса

Данное состояние вводится сервисом **io_initiate**. Указанный тип периферийного интерфейса (компонента) выбран и может использоваться. Число его типов не ограничено. Любое количество состояний выбора одних типов может существовать одновременно с выбором других.

6.7.4.4 Состояние конфигурации канала

Данное состояние принадлежит одному каналу интерфейса выбранного типа и вводится сервисом **io_open.request** из состояния выбора типа или сервисом **io_config.request** из рабочего состояния канала. Если конфигурация периферийного интерфейса (компонента) с сервисом **io_open** неверна, то состояние выбора типа интерфейса возвращается. Если реконфигурация сервисной процедурой **io_config** снова неудачна, то рабочее состояние канала возвращается вместе со старой конфигурацией.

6.7.4.5 Рабочее состояние канала

Данное состояние вводится, если конфигурация канала периферийного интерфейса (компонента) успешно закончена. В противном случае оно вводится снова, если реконфигурация не проходит и сохраняется старая конфигурация.

6.7.4.6 Состояние ввода на свободном канале

Канал готов к передаче данных и ожидает команды на выполнение сервиса ввода по запросу **io_write.request**. Если области считывания и выполнения команды находятся в состоянии «считывание в свободном канале» или «выполнение операции в свободном канале» соответственно, то рабочее состояние канала может быть оставлено сервисом **io_close**.

6.7.4.7 Состояние ввода

Канал находится в состоянии передачи данных, которая может быть остановлена с помощью сервиса **io_cancel**. Изменение данного состояния происходит, если передача данных закончена или ее время больше допуска. С помощью сервиса **io_stat** может быть подтверждено фактическое количество переданных байтов.

6.7.4.8 Состояние окончания ввода

Ввод данных закончен или прерван, и получено сообщение о подтверждении для повторного входа в состояние ввода в свободном канале.

6.7.4.9 Состояние считывания в свободном канале

Канал готов к получению данных и ожидает запроса на считывание **io_read.request**. Если области ввода и выполнения операций находятся в состоянии «ввод данных» или «выполнение операции» соответственно, то состояние рабочего канала может быть оставлено с помощью сервиса **io_close**.

6.7.4.10 Состояние считывания

Канал получает данные и ожидает их периферийного ввода. Режим получения данных может быть прерван сервисом **io_cancel**. Состояние изменяется, если этот процесс завершен или его продолжительность превышает допуск. Сервис **io_stat** подтверждает фактическое количество полученных байтов.

6.7.4.11 Состояние окончания считывания

Считывание закончено или прервано. Подтверждение создается для повторного входа в состояние считывания в свободном канале.

6.7.4.12 Состояние выполнения операции в свободном канале

Канал готов к выполнению периферийного процесса и ожидает запроса на выполнение операции **io_execute.request**. Если области считывания и ввода информации в данном канале находятся в состоянии «считывание в свободном канале» или «ввод данных в свободном канале» соответственно, то рабочее состояние канала может быть изменено сервисом **io_close**.

6.7.4.13 Состояние выполнения

Периферийный процесс выполняется, что может быть прервано сервисом **io_cancel**. Переход из данного состояния происходит, если выполнение закончено или его продолжительность превышает допуск.

6.7.4.14 Состояние завершения выполнения

Выполнение закончено или прервано и подтверждение создано для повторного входа в состояние выполнения в свободном канале.

Приложение А
(справочное)

Практическое руководство пользователя интерфейса RMSI. Описание сервисов на языке программирования C/C++

А.1 Вопросы соответствия

Использование данного руководства повышает уверенность в том, что различные варианты практической реализации интерфейса RMSI в одной операционной системе совместимы с другими интерфейсами и сервисными процедурами, а также с адаптерами платформы ASAM GDI версии 4.3 [6].

А.2 Использование описаний

Указанные описания недостаточны для практической реализации RMSI. Необходимо также принимать во внимание описания сервисов, данные в разделе 6.

А.3 Стандарты языков программирования С и С++

Язык программирования С установлен стандартом ИСО/МЭК 9899, язык программирования С++ — стандартом ИСО/МЭК 14882.

А.4 Условные обозначения простого типа данных

Если не оговорено иное, то простой тип данных C/C++, применяемых в приложении, представляется в 32-битовой среде. Поэтому в большинстве случаев эти данные не зависят от выбора операционной системы. При использовании другой среды (например, 64-битовой) некоторые простые типы данных могут зависеть от выбора операционной системы. Для указанных случаев практическая реализация документируется по отношению к соответствующим спецификациям ASAM GDI.

Информация в строках представлена восьмибитовыми байтами без значения «0», означающего конец строки.

А.5 Специальные простые типы данных

А.5.1 Типы данных, не зависящие от операционной системы

Для примитивных параметров сервисов определения специальных простых типов данных, не зависящих от операционной системы, применяются в соответствии с таблицей А.1.

Таблица А.1 — Типы данных для параметров, не зависящих от операционной системы OS

Тип данных	Определения языка C/C++	Примечания
APICHAR	signed char (характеристика со знаком)	Диапазон от –128 до 127
APIBYTE	unsigned char (характеристика без знака)	Диапазон от 0 до 2exp8
APIRET	signed short (короткое представление со знаком)	Диапазон от –65536 до 65535
APIHND	unsigned long (длинное представление без знака)	Диапазон от 0 до 2exp32

А.5.2 Типы данных, зависящие от операционной системы

Для примитивов параметров сервисов определения специальных простых типов данных, зависящих от операционной системы, используются в соответствии с таблицей А.2.

Таблица А.2 — Вызываемые функции, зависящие от операционной системы OS

Тип данных	Определение на языке С в операционной системе	
	Win32	Linux
PA_CALL	Stdcall	Специальный тип функций отсутствует
PA_CB	Cdecl	Специальный тип функций отсутствует

А.6 Специальные сложные типы данных

А.6.1 Соглашения компиляции

Сложные типы данных компилируются и размещаются в памяти восьмибитовыми байтами.

А.6.2 Структуры данных на языке «С»

Для некоторых примитивных параметров сервисов структуры данных, не зависящие от операционной системы, используются в соответствии с таблицей А.3.

Таблица А.3 — Структура данных, не зависящая от операционной системы OS

Тип данных	Утверждения языка C/C++	Примечания

IO_STAT	struct
	{
	short errorCode;
	unsigned long nrChrs;
	};
		errorCode:
		статус (ошибка) канала периферийного интерфейса
		nrChr:
		количество полученных (переданных) символов

IO_CONFDAT	struct
	{
	char*name;
	short typeId;
	void *paramPtr;
	PA_CB *completePtr;
	PA_CB*eventPtr;
	};
		name:
		Название канала периферийного интерфейса
		typeId:
		Идентификатор типа интерфейса
		paramPtr:
		Адрес структуры особых данных конфигурации канала
		completePtr:
		Адрес вызываемой функции после завершения установки асинхронной связи
		eventPtr:
		Адрес вызываемой функции, если событие происходит внутри RMS-провайдера или расширенного провайдера

OS_UCT	struct
	{
	long seconds;
	unsigned long microSec;
	};
		seconds:
		количество секунд, прошедших с 00:00 часов 1 января 1970 г. (по Гринвичу)
		microSec:
		количество микросекунд, прошедших с последней полной секунды

A_time	struct
	{
	short year;
	char month;
	char mday;
	char hour;
	char minute;
	char second;
	short milliSec;
	short microSec;
	short nanoSec;
	long timeZoneDiff;
	};
		year:
		фактический год (например, 2007)
		month:
		фактический месяц (1-12), считая с января
		mday:
		фактический день (1-31), считая с первого дня месяца
		hour:
		количество часов (0-23), считая от полуночи
		minute:
		количество минут (0-59) с начала часа
		second:
		количество секунд (0-59) с начала минуты
		milliSec:
		количество миллисекунд (0-999) с начала секунды
		microSec:
		количество микросекунд (0-999) с начала миллисекунды
		nanoSec:
		количество наносекунд (0-999) с начала микросекунды
		
		timeZoneDiff:
		отличие между временем по Гринвичу и местным временем в секундах

А.7 Соглашение для предварительно определенных констант

Если константы устанавливаются по имени, то они должны быть определены в файлах заголовка, названия которых для платформы ASAM GDI указаны в соответствующих спецификациях. Содержимое файлов заголовка может расширяться при дальнейшей практической реализации. Комбинации различных версий драйверов устройств и адаптеров платформ могут также обрабатываться внутри файлов заголовка. Указанные вопросы в настоящем стандарте не рассматриваются.

Предварительно определенные константы, применяемые в настоящем приложении, приведены в таблице А.4.

Таблица А.4 — Предварительно определенные константы

Название	Значение	Описание
IOEXT_getFHECID	0	Идентификатор специальной операции, необходимый для ее поиска по названию

А.8 Соглашения для прототипов функций

Прототипы функций:

returnType callType functionName (перечень типов аргумента).

В следующих описаниях возвращенное значение типа **returnType** обозначается **return**. Аргументы обозначаются **arg** и нумеруются, начиная с единицы, например **arg1**, **arg2** и т. д.

А.9 Возвращение значений

Нумерация в таблице А.5 используется для возвращения значений нескольких функций.

Таблица А.5 — Нумерация возвращенных значений

Нумерация	Название	Описание
0	COM_FIN	Функция выполнена успешно в синхронном режиме
1	COM_BUSY	Функция запущена в асинхронном режиме

Если ожидаемым возвращаемым значением является указатель (описатель), то оно равно нулю (NULL) только в случае ошибки.

А.10 Номера ошибок

Большинство функций в случае ошибок выдают отрицательные значения. Их смысл поясняется в таблице А.6.

Таблица А.6 — Номер ошибки

№ ошибки	Описание
–1	Тип интерфейса неизвестен или не открыт
–2	Расширенный провайдер сервисов недоступен, не ограничен или не открыт
–3	Тип интерфейса уже инициализирован
–4	Ошибка памяти
–5	Сбой аппаратных средств
–6	Доступ временно невозможен, процесс занят
–7 ... –9	Номер зарезервирован для последующего использования
–10	Канал неизвестен или не открыт (название канала не найдено)
–11	Канал уже открыт
–12	Отсутствует название канала в перечне аргументов
–13	Отсутствует адрес обратного вызова для асинхронной связи
–14	Отсутствует адрес обратного вызова для обработки события

Окончание таблицы А.6

№ ошибки	Описание
–15	Адрес протокола неверен или не найден
–16	Адрес порта неверен или не найден
–17	Скорость передачи данных не отрегулирована
–18	Длина передаваемых данных не отрегулирована
–19	Длина характеристики неправильна
–20	Ошибка буфера передачи данных, сбой при выделении (перевыделении) памяти
–21 ... –24	Номер зарезервирован для последующего использования
–25	Функция не поддержана расширенными сервисами
–26	Процесс передачи данных занят
–27	Процесс получения данных занят
–28, –29	Номер зарезервирован для последующего использования
–30	Сервисный описатель неизвестен или уже применен данным устройством
–31 ... –34	Номер зарезервирован для последующего использования
–35	Установленная связь не может быть отменена
–36 ...–39	Номер зарезервирован для последующего применения
–40	Установление связи или выполнение операции заблокировано
–41	Ресурс недоступен
–42	Связь отменена внутри процесса или самим пользователем
–43	Очередь на отправку данных перегружена
–44	Очередь на получение данных перегружена
–45 ...–49	Номер зарезервирован для последующего использования
–50	Название функции неизвестно
–51	Идентификатор функции неизвестен
–52 ...–89	Номер зарезервирован для последующего применения
–90	Идентификатор операции неизвестен
–91	Операция не выполнена
–92 ...–99	Номер зарезервирован для последующего использования
–100	Не указана ошибка параметра или отсутствует структура параметра
<–100	Спецификация ошибки параметра: абсолютное значение — 100 равно индексу первого неверного параметра в его структуре, элементы которой нумеруются, начиная с единицы, в сторону возрастания

А.11 Сервисные функции поддержки управления

Функции сервисов поддержки управления описаны в таблице А.7.

Таблица А.7 — Описание функций сервисов поддержки управления

Сервисная услуга	Прототип функции	Связь с сервисными параметрами
Получение сервисной ссылки	void * PA_CALL getFuncAddress (short, APICHAR *)	arg1: № версии (см. ниже) arg2: сервисный идентификатор, т. е. название сервиса, используемого для идентификации как строка return: адрес соответствующей функции (значение NULL, если функция не найдена)

Окончание таблицы А.7

Сервисная услуга	Прототип функции	Связь с сервисными параметрами
Инициирование типа периферийного интерфейса	short PA_CALL io_initiate (APICHAR * , APICHAR *)	arg1: строка, содержащая название провайдера расширенных сервисов, или пустая строка, если его наличие не предполагается arg2: строка, содержащая название типа подсоединенного интерфейса return: положительное число для идентификации типа подсоединенного интерфейса (отрицательное число для индикации ошибки по таблице А.6)
Принятие решения о типе периферийного интерфейса	short PA_CALL_io_conclude (short)	arg1: идентификатор типа, возвращаемый сервисом io_initiate return: значение COM_FIN, если интерфейс закрыт правильно (отрицательное значение для индикации ошибки по таблице А.6)

Номер версии запрошенного сервиса для функции **getFuncAddress** закодирован двумя числами в верхнем байте и нижнем аргументе **arg1**.

Для перечня возможных ошибок, возвращаемого функцией, см. соответствующий параметр ошибки в описании сервиса, приведенного в 6.3.

А.12 Функция сервиса входа/выхода

А.12.1 Функции, связанные с описаниями сервисов в 6.4

Функции сервисов входа/выхода описаны в таблице А.8.

Таблица А.8 — Групповое описание функций входа/выхода

Сервис	Прототип функции	Связь с сервисными параметрами
Открытие канала периферийного интерфейса	short PA_CALL io_open (IO_CONFDAT *)	arg1: структура данных (дана в таблице А.3) return: положительное число для идентификации открытого канала для последующего доступа (отрицательное число для индикации ошибки по таблице А.6)
Реконфигурация канала периферийного интерфейса	short PA_CALL io_config (short, IO_CONFDAT *)	arg1: идентификатор открытого канала arg2: структура данных (дана в таблице А.3) return: значение COM_FIN, если канал реконфигурирован правильно (отрицательное значение для индикации ошибки по таблице А.6)
Считывание данных	short PA_CALL io_read (short, APIBYTE *, unsigned long, IO_STAT*, APIHND, unsigned long)	arg1: идентификатор открытого канала arg2: адрес области памяти, выделенной пользователем, для хранения полученных данных arg3: максимальное количество полученных байтов arg4: структура данных (приведена в таблице А.3) arg5: положительный идентификатор процесса связи, введенный RMS-пользователем и отосланный назад RMS-провайдером с помощью сервиса io_complete; нулевое значение указывает на синхронную связь arg6: максимальное время протекания процесса связи в миллисекундах return: значение COM_FIN, если запрошена синхронная связь, и процесс закончен правильно (значение COM_BUSY, если процесс запущен для асинхронного получения данных или отрицательное значение для индикации ошибки по таблице А.6)
Ввод данных	short PA_CALL io_write (short, APIBYTE *, unsigned long, IO_STAT*, APIHND, unsigned long)	arg1: идентификатор открытого канала arg2: адрес области памяти, выделенной пользователем для хранения передаваемых данных arg3: количество передаваемых байтов arg4: структура данных (приведена в таблице А.3) arg5: положительный идентификатор процесса связи, заданный RMS-пользователем и отосланный назад RMS провайдером с помощью сервиса io_complete; нулевое значение указывает на синхронную связь

Окончание таблицы А.8

Сервис	Прототип функции	Связь с сервисными параметрами
		arg6: максимальное время процесса передачи данных в миллисекундах return: значение COM_FIN, если запрошена синхронная связь, и процесс закончился правильно (значение COM_BUSY, если процесс запущен для асинхронной передачи данных, или отрицательное значение для индикации ошибки по таблице А.6)
Выполнение операции	short PA_CALL io_execute (short, APIHND, void *, void *, void *, APIHND, unsigned long)	arg1: идентификатор открытого канала arg2: идентификатор операции arg3: адрес области памяти, выделенной пользователем для хранения входных данных операции arg4: адрес области памяти, выделенной пользователем для хранения выходных данных операции arg5: адрес области памяти, выделенной пользователем для хранения возвращаемых данных операции arg6: положительный идентификатор процесса связи, задаваемый RMS-пользователем и отосланный назад RMS-провайдером с помощью сервиса io_complete; нулевое значение указывает на использование синхронной связи arg7: максимальная продолжительность процесса выполнения в миллисекундах return: значение COM_FIN, если запрошена синхронная связь, и процесс закончился правильно (значение COM_BUSY, если запущен асинхронный процесс, или отрицательное значение для индикации ошибки по таблице А.6)
Отмена связи	short PA_CALL io_cancel (short, APIHND)	arg1: идентификатор открытого канала arg2: идентификатор процесса связи, переданный с помощью запуска ожидающего сервиса return: значение COM_FIN, если указанный процесс связи отменен (отрицательное значение для индикации ошибки по таблице А.6)
Получение статуса канала периферийного интерфейса	short PA_CALL io_stat (short, APIHND, IO_STAT *)	arg1: идентификатор открытого канала arg2: идентификатор процесса связи, переданный с помощью запуска проверяемого сервиса arg3: структура данных (приведена в таблице А.3) return: значение COM_FIN, если статус указанного процесса связи подтвержден (отрицательное значение для индикации ошибки по таблице А.6)
Очистка буфера считывания	short PA_CALL io_clear (short)	arg1: идентификатор открытого канала return: значение COM_FIN, если буфер считывания указанного канала очищен (отрицательное значение индикации ошибки по таблице А.6)
Закрытие канала периферийного интерфейса	short PA_CALL io_close (short)	arg1: идентификатор открытого канала return: значение COM_FIN, если указанный канал закрыт (отрицательное значение для индикации ошибки по таблице А.6)
Выдача сигнала о наступлении события	short PA_CB io_event (short, APIHND, void *)	arg1: идентификатор канала, которому принадлежат данные события (нулевое значение, если событие не принадлежит каналу или относится ко всем каналам в режиме широковещания) arg2: идентификатор события arg3: структура данных, соответствующих указателю или событию return: значение COM_FIN, если событие обработано (значение COM_BUSY, если обработка запущена и продолжается, а также отрицательное значение индикации ошибки, когда событие не обрабатывается)

Функция **io_execute** может также использоваться для получения идентификатора операции с помощью ее названия. Для этого операция с предварительно определенным идентификатором **IOEXT_getFHECID** вызывается с именем требуемой операции в качестве входного параметра. Выходным параметром является идентификатор операции.

Функция **io_event** описывается пользователем интерфейса RMSI и вызывается RMS-провайдером при наступлении локального события. В среде C/C++ для обработки указанных событий применяются функции обратного вызова, адрес которых находится у RMS-провайдера. Функция вызывается сервисом **io_open** и изменяется сервисом **io_config**.

A.12.2 Особые функции асинхронной связи

Описание сервисного управления ресурсами в разделе 6 нейтрально по отношению к специальным сценариям типа синхронной и асинхронной связи. В среде C/C++ используют функции асинхронной связи обратного вызова.

Пользователь интерфейса RMSI применяет функции, описанные в таблице A.9. Они вызываются RMS-провайдером после установления связи сервисами **io_read, io_write** или **io_execute**. Адрес функции предоставляется RMS-провайдером с помощью сервиса **io_open** и может быть изменен сервисом **io_config**.

Таблица A.9 — Особая функция асинхронной связи

Сервисная услуга	Прототип функции	Связь с сервисными параметрами
Никакие заданные сервисы асинхронной связи не указаны	short PA_CB io_complete (APIHND, IO_STAT*)	arg1: идентификатор процесса связи, установленный RMS-пользователем с помощью сервисов io_read, io_write или io_execute и отосланный назад RMS-провайдером с помощью сервиса io_complete arg2: структура данных (приведена в таблице A.3) return: значение COM_FIN

A.13 Расширенные сервисные функции

A.13.1 Функции, связанные с описанием сервисов в 6.5

Функции расширенных сервисов представлены в таблице A. 10.

Таблица A.10 — Функции расширенных сервисов

Сервис	Прототип функции	Связь с сервисными параметрами
Инициирование интерфейса расширенного типа	short PA CALL ext_initiate (APICHAR *, short)	arg1: название интерфейса (см. сервис io_initiate) arg2: идентификатор типа интерфейса, определенный RMS внутренне return: значение COM_FIN, если интерфейсом можно пользоваться (значение ошибки по таблице A.6)
Принятие решения об использовании интерфейса расширенного типа	short PA CALL ext_conclude (short)	сервисная io_conclude (см.таблицу A.3)
Открытие расширенного интерфейса	short PA CALL ext_open (IO_CONFDAT *, short)	arg1: структура данных (см. таблицу A.3) arg2: идентификатор канала интерфейса определяется RMS внутренне return: значение COM_FIN, если канал свободен для последующего доступа (отрицательное значение индикации ошибки по таблице A.6)
Реконфигурация расширенного интерфейса	short PA_CALL ext_config (short, IO_CONFDAT *)	сервис io_config (*)
Считывание данных расширенного интерфейса	short PA_CALL ext_read (short, APIBYTE *, unsigned long, IO_STAT*, APIHND, unsigned long)	сервис io_read (*)
Ввод данных расширенного интерфейса	short PA_CALL ext_write (short, APIBYTE *, unsigned long, IO_STAT*, APIHND, unsigned long)	сервис io_write (*)
Выполнение операции расширенного интерфейса	short PA CALL ext_execute (short, APIHND, void*, void*, void*, APIHND, unsigned long)	сервис io_execute (*)

Окончание таблицы А.10

Сервис	Прототип функции	Связь с сервисными параметрами
Отмена расширенной связи	short PA CALL ext_cancel (short, APIHND)	сервис io_cancel (*)
Получение статуса расширенного интерфейса	short PA CALL ext_stat (short, APIHND, IO_STAT *)	сервис io_stat (*)
Очистка буфера считывания расширенного интерфейса	short PA CALL ext_clear (short)	сервис io_clear (*)
Закрытие расширенного интерфейса	short PA CALL ext_close (short)	сервис io_close (*)
Выдача сигнала о расширенном событии	short PA CB ext_event (short, APIHND, void *)	сервис io_event (*)

(*) См. таблицу А.8

Функция **ext_event** устанавливается в процессе управления ресурсами и вызывается провайдером расширенных сервисов при локальном событии. В среде C/C++ для обработки указанных событий используются функции обратного вызова адрес которых предоставляется провайдером расширенных сервисов с помощью сервиса **ext_open** и изменяется сервисом **ext_config**.

А.13.2 Особая функция асинхронной связи

Описание сервисов управления ресурсами в разделе 6 нейтрально по отношению к специальным сценариям типа синхронной и асинхронной связи. В среде C/C++ применяются функции обратного вызова для обработки асинхронной связи.

Сервисное управление ресурсами использует функцию (см. таблицу А.11), вызываемую провайдером расширенных сервисов о завершении процесса установления связи, запущенного сервисами **ext_read, ext_write** или **ext_execute**. Адрес данной функции доставляется провайдером расширенных сервисов с помощью сервиса **ext_open** и может быть изменен сервисом **ext_config**.

Таблица А.11 — Особая функция асинхронной связи

Сервис	Прототип функции	Связь с сервисными параметрами
Никакие явно заданные сервисы не указаны для обработки данных асинхронной связи	short PA_CB ext_complete (APIHND, IO_STAT *)	arg1: идентификатор процесса связи, установленный RMS методом вызова сервисов услуг ext_read, ext_write или ext_execute, и отосланный назад провайдером расширенных сервисов с помощью сервиса io_complete arg2: структура данных (приведена в таблице А.3) return: COM_FIN

А.14 Сервисные функции поддержки обслуживания

А.14.1 Функции, связанные с описанием сервисов в 6.6

Функции расширенных сервисов приведены в таблице А.12.

Таблица А.12 — Сервисные функции поддержки обслуживания

Сервис	Прототип функции	Связь с сервисными параметрами
Выделение памяти	APIBYTE * PA_CALL os_allocate (unsigned long)	arg1: количество элементов типа APIBYTE в соответствующей области памяти return: указатель выделенной области памяти либо NULL, если память не выделена
Перевыделение памяти	APIBYTE * PA_CALL os_reallocate (APIBYTE *, unsigned long)	arg1: указатель перевыделенной области памяти arg2: количество элементов типа APIBYTE в соответствующей области памяти return: указатель перевыделенной области памяти либо NULL, если перевыделение памяти не выполнено

Продолжение таблицы А.12

Сервис	Прототип функции	Связь с сервисными параметрами
Освобождение памяти	short PA_CALL os_free (APIBYTE *)	arg1: указатель освобождаемой области памяти return: значение COM_FIN, если память освобождена (отрицательное значение для индикации ошибки по таблице А.6)
Определение времени Версия, представленная структурой A_time	void PA_CALL os_time a (A_time *)	arg1: указатель структуры, содержащей результат A_time return: невозвращаемое значение
Определение времени Версия, представляемая структурой OS_UCT	void PA_CALL os_time (OS_UCT *)	arg1: указатель структуры, содержащей результат OS_UCT return: невозвращаемое значение
Определение времени обработки	unsigned long PA_CALL os_clock (void)	return: время работы центрального процессора в микросекундах
Ожидание	void PA_CALL os_delay (unsigned long)	arg1: время ожидания в миллисекундах return: невозвращаемое значение
Создание таймера	APIHND PA_CALL os_settimer (pTimerCB, unsigned long, APIHND, unsigned long)	arg1: указатель функции обратного вызова по истечении времени ожидания таймера (см. подробности в А.14.2) arg2: запрошенная продолжительность до истечения времени ожидания таймера arg3: идентификатор данного таймера, являющийся аргументом функции обратного вызова arg4: количество повторений таймера, значение 0 определяет периодически запускаемый таймер, который можно остановить сервисом os_killtimer return: идентификатор данного таймера 0: таймер не может быть создан
Выдача таймером сигнала о событии		Для данного сервиса необходима инсталляция указателя обратного вызова функции. (см. arg1 прототипа функции os_settimer, детали см. в А.14.2).
Удаление таймера	short PA CALL os_killtimer (APIHND)	arg1: идентификатор таймера, который должен быть остановлен и/или удален; возвращаемое значение функции os_settimer return: значение COM_FIN, если таймер удален (отрицательное значение для индикации ошибки по таблице А.6)
Создание таймера со световой индикацией	APIHND PA_CALL os_setLPTimer (PtimerCB, unsigned long, APIHND, unsigned long)	arg1: указатель функции обратного вызова, когда время ожидания таймера истекает (подробности см. в А.14.2) arg2: запрошенная продолжительность до окончания времени ожидания таймера arg3: идентификатор таймера, являющийся аргументом функции обратного вызова arg4: количество повторений таймера, значение 0 задает периодически запускаемый таймер, который можно остановить функцией os_killtimer return: идентификатор данного таймера 0: таймер не может быть создан
Сигнал светового таймера о наступлении события		Инсталяция указателя обратного вызова функции (см. arg1 прототипа os_setLPTimer, подробности в разделе А. 14.2)

Продолжение таблицы А.12

Сервис	Прототип функции	Связь с сервисными параметрами
Удаление таймера со световой индикацией	short PA CALL os_killLPTimer (APIHND)	arg1: идентификатор таймера, подлежащего остановке и/или удалению, — возвращаемое значение функции os_setLPTimer return: значение COM_FIN, если таймер удален (отрицательное значение для индикации ошибки по таблице А.6)
Идентификация процесса световой индикации	APIHND PA_CALL os_getLPnumber (void)	return: идентификатор текущего процесса световой индикации для вызывающего оператора данной функции 0: текущий процесс световой индикации неизвестен
Создание вычислительного семафора	APIHND PA_CALL os_createSem (unsigned long)	arg1: счетчик семафора, показывающий количество одновременного использования ресурса return: порядок обработки сигнала семафора
Ожидание вычислительного семафора	short PA_CALL os_waitSem (APIHND, unsigned long)	arg1: порядок обработки сигнала семафора, возвращаемый функцией os_createSem arg2: максимальное время (миллисекунды) ожидания до получения доступа к ресурсу return: значение COM_FIN, если доступ гарантирован (отрицательное значение для индикации ошибки по таблице А.6)
Открытие вычислительного семафора	short PA CALL os_releaseSem (APIHND)	arg1: порядок обработки сигнала семафора, возвращаемый функцией os_createSem return: значение COM_FIN, если семафор открыт (отрицательное значение для индикации ошибки по таблице А.6)
Стирание вычислительного семафора	short PA CALL os_deleteSem (APIHND)	arg1: порядок обработки сигнала семафора, возвращаемый функцией os_createSem return: значение COM_FIN, если семафор стерт (отрицательное значение для индикации ошибки по таблице А.6)
Создание частного семафора	APIHND PA_CALL os_createMutex (void)	return: порядок обработки сигнала семафора
Ожидание частного семафора	short PA_CALL os_waitMutex (APIHND, unsigned long)	arg1: порядок обработки сигнала семафора, возвращаемый функцией os_createMutex arg2: максимальное время (миллисекунды) ожидания для получения доступа к ресурсу return: значение COM_FIN, если доступ гарантирован (отрицательное значение для индикации ошибки по таблице А.6)
Открытие частного семафора	short PA CALL os_releaseMutex (APIHND)	arg1: порядок обработки сигнала семафора, возвращаемый функцией os_createMutex return: значение COM_FIN, если семафор открыт (отрицательное значение для индикации ошибки по таблице А.6)
Стирание частного семафора	short PA CALL os_deleteMutex (APIHND)	arg1: порядок обработки сигнала семафора, возвращаемый функцией os_createMutex return: значение COM_FIN, если семафор стерт (отрицательное значение для индикации ошибки по таблице А.6)

Окончание таблицы А.12

Сервис	Прототип функции	Связь с сервисными параметрами
Открытие журнала отладки	APIHND PA CALL os_openDebug (APICHAR *)	arg1: название канала отладки в виде строки return: данные для последующего доступа к каналу отладки (значение NULL, если канал не открыт)
Ввод сообщения отладки	short PA CALL os_writeDebug (APIHND, APICHAR *)	arg1: данные канала отладки, возвращаемые функцией os_openDebug arg2: указатель данных в строке, направляемых в канал отладки return: значение COM_FIN, если сообщение введено (отрицательное значение для индикации ошибки по таблице А.6)
Закрытие журнала отладки	short PA CALL os_closeDebug (APIHND)	arg1: данные канала отладки , возвращаемые функцией os_openDebug return: значение COM_FIN, если канал отладки закрыт (отрицательное значение для индикации ошибки по таблице А.6)

А.14.2 Подача сигнала о событии таймера

Для подачи сигнала о событии таймера пользователь интерфейса RMSI устанавливает функцию обратного вызова RMS-провайдером, когда истекает время ожидания. Адрес данной функции предоставляется RMS-провайдером созданием таймеров с помощью функций **os_settimer** и **os_setLPTimer**. Прототипы данных функций обратного вызова определяются в среде C/C++ по таблице А. 13.

Т а б л и ц а А.13 — Тип функции обратного вызова для событий таймера

Определение прототипа функции	Связь с сервисными параметрами
typedef short (PA_CB *pTimerCB) (APIHND, IO_STAT *)	arg1: идентификатор процесса связи, установленный RMS-пользователем с помощью сервиса os_settimer или os_setLPTimer и отосланный назад таймером по истечении времени ожидания arg2: структура данных (см. таблицу А.3) return: значение COM_FIN

Приложение В
(справочное)

Каскадирование драйверов устройств с помощью интерфейса RMSI

В.1 Принцип каскадирования

В ИСО 20242-1:2005 (приложение С) каскадирование описывается как ситуация, когда драйверы используют непосредственно сервисный интерфейс виртуального устройства (VDSI – Virtual Driver Service Interface) драйверов других устройств вместо интерфейса RMSI адаптера платформы. Если каскадирование необходимо для стандартных драйверов устройств, то интерфейс RMSI требуется для покрытия драйвера устройства нижележащей системы.

Такая система может быть полным стеком, соответствующим ИСО 20242 и включающим координатор, драйвер устройства и адаптер платформы (или только другой драйвер). Так как только один адаптер платформы может существовать в одном контексте, соответствующем ИСО 20242, то он должен быть подготовлен для каскадирования.

В.2 Каскадирование с помощью сервисного интерфейса прикладной программы APSI

Если каскадирование драйверов выполняется с помощью прикладного интерфейса APSI, то расширение адаптера платформы необходимо для установления соответствия сервисов входа/выхода интерфейса RMSI (с помощью расширенных сервисов) сервисам интерфейса APSI. Затем особый координатор получает доступ к нижнему драйверу устройства в соответствии с ИСО 20242-5 (пунктирные нештрихованные стрелки на рисунке В.1).

В.3 Каскадирование без интерфейса APSI

Если каскадирование проводится без координатора, то необходимо расширение адаптера платформы для приведения в соответствие сервисов входа/выхода интерфейса RMSI (с помощью расширенных сервисов) сервисам виртуального интерфейса VDSI. Это позволяет получить прямой доступ к нижнему драйверу устройства (пунктирные штрихованные стрелки на рисунке В.1).



Рисунок В.1 — Каскадирование драйвера с помощью интерфейса сервисного управления ресурсами RMSI

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО 20242-1	IDT	ГОСТ Р ИСО 20242-1—2010 «Системы промышленной автоматизации и интеграция. Служебный интерфейс для испытательных прикладных программ. Часть 1. Общие положения»

Примечание — В настоящей таблице использовано следующее условное обозначение степени соответствия стандарта:

- IDT — идентичный стандарт.

Редактор *Е. Г. Кузнецова*
Технический редактор *В. Н. Прусакова*
Корректор *Л. Я. Митрофанова*
Компьютерная верстка *В. Н. Романовой*

Сдано в набор 03.09.2014. Подписано в печать 08.12.2014. Формат $60\times84^{1}/_{8}$. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 6,51. Уч.-изд. л. 6,00. Тираж 44 экз. Зак. 1484.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru
Набрано и отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256.

gost.e-ecolog.ru