ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 55940— 2014

Телевидение вещательное цифровое

РАСШИРЕННЫЕ ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ К ОБЩЕМУ ИНТЕРФЕЙСУ В СИСТЕМАХ ОГРАНИЧЕНИЯ ДОСТУПА

Основные параметры

Часть 1

EN 50221:1997 (NEQ)

Издание официальное

Москва
Стандартинформ
2015

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Федеральным государственным унитарным предприятием ордена Трудового Красного Знамени научно-исследовательским институтом радио, Самарский филиал «Самарское отделение научно-исследовательского института радио»

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 480 «Связь»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 28 января 2014 г. № 7-ст

4 Настоящий стандарт разработан с учетом основных нормативных положений европейского регионального стандарта ЕН 50221:1997 «Спецификация общего интерфейса для условного доступа и других применений в цифровых видеовещательных декодерах» (EN 50221:1997 «Common interface specification for conditional access and other digital video broadcasting decoder applications», NEQ)

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, 2015

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

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины, определения, сокращения и символы . . . 2
4 Структура общего интерфейса . . . 3
4.1 Структура уровней общего интерфейса . . . 3
4.2 Физическая реализация . . . 3
4.3 Клиент-сервер . . . 3
4.4 Кодирование данных . . . 3
4.5 Расширяемость . . . 3
4.6 Использование существующих стандартов . . . 3
5 Описание архитектуры общего интерфейса . . . 4
5.1 Вводная часть . . . 4
5.2 Интерфейс транспортного потока . . . 4
5.3 Интерфейс команд . . . 4
5.4 Требования к физическому уровню . . . 5
6 Интерфейс транспортного потока (TSI) . . . 6
6.1 Физический и канальный уровни интерфейса транспортного потока . . . 6
6.2 Транспортный уровень интерфейса транспортного потока . . . 6
6.3 Верхние уровни интерфейса транспортного потока . . . 7
7 Интерфейс команд — уровни транспортные и сеансовые . . . 7
7.1 Основной транспортный уровень . . . 7
7.2 Уровень сеанса . . . 12
8 Интерфейс команд — уровень приложений . . . 13
8.1 Введение . . . 13
8.2 Ресурсы . . . 13
8.3 Модули данных протокола приложения . . . 15
8.4 Система управления ресурсами . . . 15
8.5 Ресурсы управления узлом и информацией . . . 16
8.6 Ресурс интерфейса человек — машина . . . 17
8.7 Ресурсы связи . . . 18
8.8 Идентификаторы ресурса и теги объекта приложения . . . 18
Приложение А (обязательное) Базовый физический уровень карты ПК . . . 21
Приложение Б (справочное) Дополнительные объекты . . . 24
Библиография . . . 27

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Телевидение вещательное цифровое

РАСШИРЕННЫЕ ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ К ОБЩЕМУ ИНТЕРФЕЙСУ В СИСТЕМАХ ОГРАНИЧЕНИЯ ДОСТУПА

Основные параметры

Часть 1

Digital video broadcasting LVD. Advanced technical requirements for a commoon interface of conditional access systems. Basic parameters. Part 1

Дата введения — 2014—09—01

1 Область применения

Настоящий стандарт распространяется на общий интерфейс модуля, входящего в состав приемника-декодера [узла, абонентского приемника] телевизионных программ и реализующего алгоритм ограниченного [условного] доступа к программам телевизионного вещания. Общий интерфейс включает в себя интерфейс транспортного потока и интерфейс команд. Стандарт устанавливает основные параметры общего интерфейса, обеспечивающего сопряжение модуля с приемником-декодером. Стандарт позволяет унифицировать параметры модуля при работе с приемниками-декодерами независимых производителей в различных системах условного доступа.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующий стандарт:

ГОСТ Р 52210—2004 Телевидение вещательное цифровое. Термины и определения

ГОСТ Р 53531—2009 Телевидение вещательное цифровое. Требования к защите информации от несанкционированного доступа в сетях кабельного и наземного телевизионного вещания. Основные параметры. Технические требования

ГОСТ Р ИСО/МЭК 8824-1—2001 Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации

ГОСТ Р ИСО/МЭК 8825—93 Информационная технология. Взаимосвязь открытых систем. Спецификация базовых правил кодирования для абстрактно-синтаксической нотации версии один (АСН.1)

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется принять в части, не затрагивающей эту ссылку.

Издание официальное

3 Термины, определения, сокращения и символы

3.1 В настоящем стандарте применены термины по ГОСТ Р 52210, ГОСТ Р 52591, ГОСТ Р 53528, а также следующие термины с определениями:

3.1.1 **букет программ**: Совокупность сервисов, предлагаемых абоненту как единый программный продукт.

3.1.2 **модуль** (module): Устройство, предназначенное для выполнения специализированных задач совместно с узлом, например: в качестве подсистемы ограниченного [условного] доступа, в качестве модуля приложения электронного гида (путеводителя) по программам или для предоставления ресурсов, необходимых приложению, но отсутствующих непосредственно в узле.

3.1.3 **ограниченный [условный] доступ** (conditional access; CA): Синоним системы ограниченного [условного] доступа, СОД.

3.1.4 **приложение** (application): Программное обеспечение, предоставляющее возможность решения определенной задачи. Приложение выполняется в модуле. Приложение обменивается данными с узлом и предоставляет пользователям средства, предоставляемые непосредственно узлом. Приложение может обрабатывать транспортный поток.

3.1.5 **ресурс** (resource): Единица функциональных возможностей, предоставляемых узлом для использования модулем. Ресурс определяется набором объектов, которыми обмениваются модуль и узел. Термин «ресурсы» используется вместо термина «услуги» или «службы» для того, чтобы избежать путаницы, поскольку эти термины часто встречаются как «службы» в вещании телевидения и радио.

3.1.6 **система ограниченного [условного] доступа**; СОД (conditional access system): Система, обеспечивающая защиту информации от несанкционированного доступа в сетях вещания в соответствии с ГОСТ Р 53531.

3.1.7 **служба** (service): Набор элементарных потоков, который предлагается пользователю в виде программы. Элементарные потоки связаны общей синхронизацией. Набор элементарных потоков формируется из различных данных, например видео, аудио, субтитров, других данных.

3.1.8 **сообщение, предоставляющее право доступа** (Entitlement Management Message; EMM): Сообщение, которое содержит данные, разрешающие конкретному дескремблеру открыть для просмотра конкретную программу на предусмотренный срок.

3.1.9 **сообщение, управляющее правом доступа** (Entitlement Control Message; ECM): Сообщение, которое передает дескремблеру слово управления (CW), передает информацию о критериях доступа к программам и букетам.

3.1.10 **узел** (host): Устройство, к которому могут быть подключены модуль или модули, например: интегрированый приемник-декодер, абонентский приемник, видеомагнитофон, компьютер.

3.1.11 **нерабочее время [простой]** (timeout): Состояние процесса обмена командами между узлом и модулем.

3.2 В настоящем стандарте применены следующие сокращения:

ИТС (Transport Connection Identifier, TCI) — идентификатор транспортного соединения;

МСЭ (International Telecommunication Union, ITU) — Международный союз электросвязи;

ПК — персональный компьютер;

CJL — система ограниченного [условного] доступа;

Телетекст EBU (European Broadcasting Teletext; EBT) — телетекст, нормированный EBU;

APDU (Application Protocol Data Unit) — модуль данных протокола приложения;

ASN.1 (Abstract Syntax Notation One) — абстрактная синтаксическая нотация версии 1;

CA (conditional access) — ограниченный [условный] доступ;

C_TPDU (Command Transport Protocol Data Unit) — блок данных команды транспортного протокола;

DRCS (Dynamically Redefinable Character Sets) — динамически переопределяемый набор символов;

ECM — Entitlement Control Message (сообщение, управляющее правом доступа);

EBU (European Broadcasting Union) — Европейский союз радиовещания;

EMM — Entitlement Management Message (сообщение, предоставляющее право доступа);

ITU (International Telecommunications Union) — Международный союз электросвязи; МСЭ;

ITU-T (International Telecommunications Union — Telecommunication Standardization Sector) — Сектор стандартизации электросвязи МСЭ;

LSCR (Low-Speed Communications Resource) — ресурс низкоскоростной связи;
MPEG (Motion Pictures Expert Group) — группа экспертов по движущимся изображениям;
MMI (Man-Machine Interface) — интерфейс человек — машина;
PDU (Protocol Data Unit) — модуль данных протокола;
SPDU (Session Protocol Data Unit) — модуль данных протокола сеанса;
R_TPDU (Response Transport Protocol Data Unit) — блок данных ответа транспортного протокола;
TDT (Time and Date Table) — таблица времени и даты;
TPDU (Transport Protocol Data Units) — модуль данных транспортного протокола.

4 Структура общего интерфейса

4.1 Структура уровней общего интерфейса

В настоящем стандарте общий интерфейс представлен в виде иерархической структуры, разделенной на уровни. Это обеспечивает гибкость его использования в других приложениях вне контекста приложений условного доступа.

Приложения и уровни сеансов определены для всех применений общего интерфейса. Параметры транспортного и канального уровней могут зависеть от физического уровня, используемого в конкретной реализации. В настоящем стандарте определены параметры физического интерфейса и содержится полная спецификация физического модуля.

Узел может установить транспортное соединение с несколькими модулями, которые могут быть связаны с узлом непосредственно или косвенно.

Каждое соединение поддерживается в присутствии модуля. Каждый модуль может управлять несколькими сеансами с узлом.

4.2 Физическая реализация

Базовая спецификация включает в себя спецификацию физического интерфейса, совместимого с картой ПК, выполненной в соответствии со стандартом, используемым в индустрии персональных компьютеров. Допускается использование других физических реализаций.

4.3 Клиент-сервер

Настоящий стандарт устанавливает параметры интерфейса, разработанного по принципу, в соответствии с которым приложения, как клиенты, могут использовать ресурсы, предоставляемые сервером. Приложения размещены на модуле, и ресурсы могут быть предоставлены узлом или другим модулем, управляемым узлом.

4.4 Кодирование данных

Параметры передачи данных через интерфейс команд определены в терминах объектов. Объекты кодируются кодированием всей совокупности «тег-длина-величина», полученной при использовании синтаксиса для кодирования согласно ГОСТ Р ИСО/МЭК 8824-1, ГОСТ Р ИСО/МЭК 8825. Настоящий стандарт допускает расширение этого правила.

В случае использования специфического кодирования для транспортного уровня некоторой реализации карты ПК, для другой физической реализации допускается использование других правил кодирования. Семантика в обоих случаях должна быть идентичной.

4.5 Расширяемость

Настоящий стандарт предусматривает возможность расширения при добавлении новых объектов и расширении существующих. Это обеспечивает отсутствие проблем дефицита пространства тегов кодирования или ограничений на значения длины. Менеджер ресурсов предоставляет механизм расширения спектра ресурсов, предоставляемых узлами, как для целей условного доступа, так и для других приложений базовых модулей.

4.6 Использование существующих стандартов

В настоящем стандарте предусматривается использование существующих стандартов индустрии персональных компьютеров, обеспечивающих применение уже стандартизованных структурных блоков, что устраняет необходимость разработки новых стандартов и дает возможность повторного использования программного обеспечения и аппаратных средств.

5 Описание архитектуры общего интерфейса

5.1 Вводная часть

Пример структурной схемы приемного оборудования СОД представлен на рисунке 1.

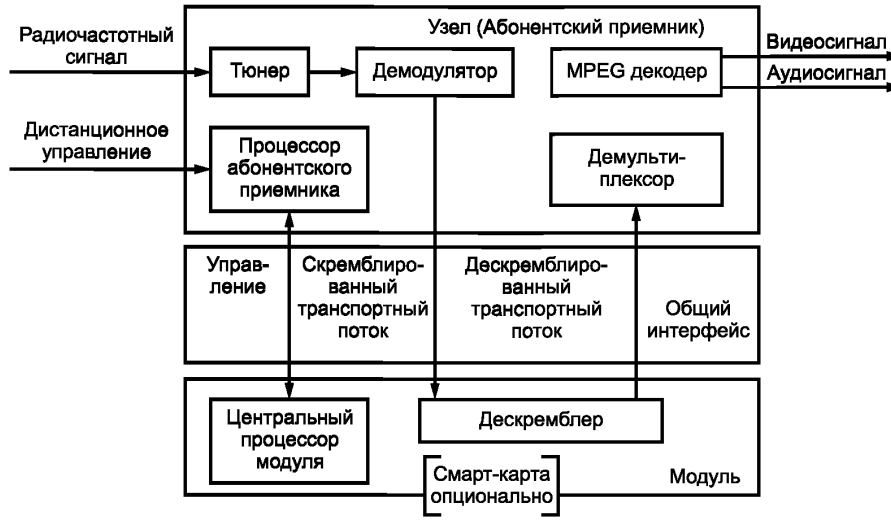

Рисунок 1 — Структурная схема приемного оборудования СОД

Общий интерфейс является составной частью СОД. Он входит в состав приемного оборудования СОД и обеспечивает сопряжение абонентского приемника с модулем. Архитектура общего интерфейса представлена в ГОСТ Р 53531 (п. 4.2.4).

В состав общего интерфейса в соответствии с ГОСТ Р 53531 (п. 4.2.4.1) входят многоуровневые интерфейсы транспортного потока и команд. Верхние уровни интерфейсов являются общими для всех реализаций. Для реализаций нижних уровней допускаются альтернативные решения. Настоящий стандарт базируется на стандартной карте ПК, требования к которой соответствуют спецификациям [1]—[3].

5.2 Интерфейс транспортного потока

Интерфейс транспортного потока переносит транспортные пакеты MPEG-2 между узлом и модулем в обоих направлениях. Если модуль разрешает доступ к службам в транспортном потоке и эти службы были отобраны узлом, то пакеты, несущие эти службы, будут направлены в модуль для дескремблирования, пакеты других служб не дескремблируются. Уровни интерфейса транспортного потока показаны на рисунке 2. Параметры транспортного уровня и всех верхних уровней определены в спецификации MPEG-2 согласно стандарту [4].

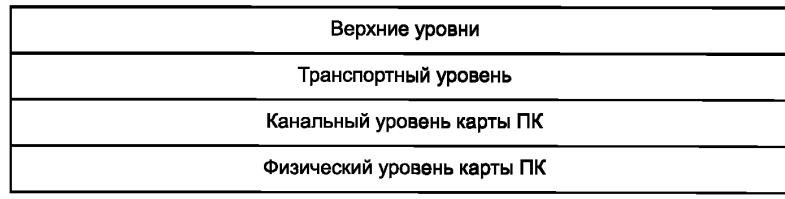

Рисунок 2 — Уровни интерфейса транспортного потока

5.3 Интерфейс команд

Интерфейс команд функционирует в модуле и в узле и обеспечивает все связи между приложениями.

Для обеспечения необходимой функциональности протоколы связи интерфейса команд определены на нескольких уровнях. Необходимая функциональность включает в себя:

- возможность поддержки нескольких модулей на одном узле;
- способность поддерживать сложные комбинации операций между модулем и узлом;
- набор функциональных примитивов (объектов), которые позволяют узлу предоставлять ресурсы модуля с возможностью их расширения. Уровни интерфейса команд показаны ниже на рисунке 3.

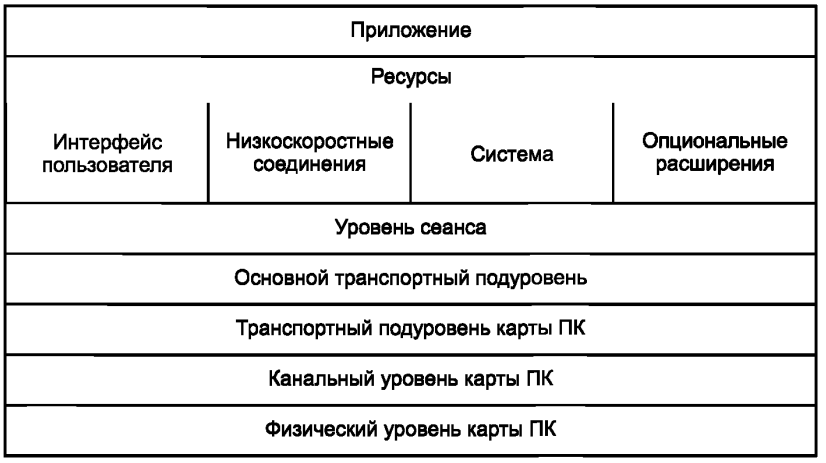

Рисунок 3 — Уровни интерфейса команд

Реализация карты ПК имеет свои физические и канальные уровни, а также собственный транспорт нижнего подуровня. Будущие физические реализации могут различаться на этих уровнях, а различия будут ограничены этими уровнями. Реализация конкретных особенностей транспорта на нижнем подуровне ограничена кодированием и конкретными деталями протокола обмена сообщениями, а также общим верхним подуровнем, определяющим идентификацию, установление и разрыв соединений транспортного уровня. Уровни сеанса, ресурсов и приложений являются общими для всех физических реализаций.

К интерфейсу на уровне приложений не предъявляются требования применения конкретной семантики. Такие ресурсы связи, как интерактивный интерфейс пользователя и передача данных с низкой скоростью, узел предоставляет приложениям, работающим на модуле.

5.4 Требования к физическому уровню

5.4.1 Введение

В данном разделе определены требования к физическому уровню интерфейса, которые определяют параметры механического и электрического соединения узла и модуля, то есть тип и размер сокета, количество контактов, напряжения, сопротивления, пределы мощности.

В данном разделе определены требования и ограничения на следующие характеристики:

- транспортный поток и команды логических связей;
- скорость передачи данных;
- режимы работы при подключении и отключении;
- низкий уровень инициализации;
- использование нескольких модулей физического уровня.

5.4.2 Соединения данных и логических команд

На физическом уровне должны поддерживаться независимые двухсторонние логические соединения для транспортного потока и для команд. Интерфейс транспортного потока принимает транспортный поток MPEG-2, состоящий из последовательности транспортных пакетов, поступающих либо непрерывно, либо разделенных нулевыми данными. Возвращенный транспортный поток может иметь часть входящих транспортных пакетов в дескремблированной форме. Ограничения, накладываемые на параметры интерфейса транспортного потока, должны быть в соответствии со стандартом [5] (5.4.2).

5.4.3 Режимы работы при подключении и отключении модуля к узлу

Физический уровень должен поддерживать подключение и отключение модуля в любое время, независимо от того, узел включен (питается) или нет.

Подключение или отключение модуля не должно вызывать электрических повреждений модуля или узла и не должно создавать ложные модификации данных, которые хранятся в энергонезависимой памяти модуля.

Если модуль не подключен, то интерфейс транспортного потока должен выполнить обход модуля и интерфейса команд неактивного модуля. На соединении с модулем узел:

- создает низкоуровневые последовательности, инициирующие модуль;
- выполняет все процедуры низкого уровня установления соединения, используемые на физическом уровне;
- устанавливает совместимость модуля с модулем DVB.

Если инициализация успешно завершена, то узел вводит модуль в тракт транспортного потока. Допускается потеря части данных транспортного потока во время этого процесса. Одновременно соединение транспортного уровня устанавливается на интерфейсе команд для инициализации уровня приложения.

Если физический уровень используется в других приложениях, не относящихся к соединению DVB-совместимых модулей, и если к узлу подключается несовместимый модуль, то повреждения модуля или узла не должны возникать и узел должен пытаться завершить инициализацию как в случае DVB-совместимых модулей.

Опционально узел может сигнализировать пользователю о подключении несовместимого модуля. При отключении модуля узел должен вывести модуль из тракта транспортного потока данных. Потеря части данных транспортного потока в этом процессе является допустимой. Кроме того, узел должен прекратить соединение интерфейса команд.

5.4.4 Подключение к узлу нескольких модулей

Прикладной уровень не накладывает никаких ограничений на количество модулей, которые могут быть подключены к узлу. На физическом уровне разрешается подключение к узлу одновременно нескольких модулей, предел допустимого количества подключаемых модулей установлен равным 15 модулям. В случаях подключениия более одного модуля соединение транспортных потоков должно быть последовательным через каждый модуль, как показано на рисунке 4.

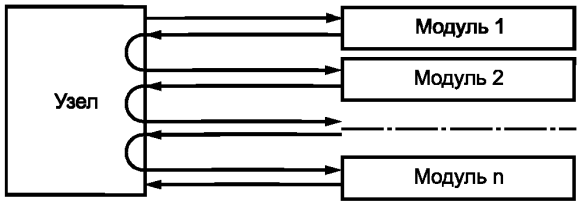

Рисунок 4 — Принцип подключения к узлу последовательно соединенных модулей

Узел должен поддерживать одновременное соединение интерфейса команд для каждого модуля. Операции между узлом и модулем должны обрабатываться независимо для каждого модуля. При отключении модуля подключение интерфейса команд транспортного уровня к любому другому модулю не должно нарушаться или прекращаться. При соединении нескольких модулей с узлом узел должен иметь возможность выбора модуля для дескремблирования выбранной службы (служб).

6 Интерфейс транспортного потока (TSI)

6.1 Физический и канальный уровни интерфейса транспортного потока

Физический и канальный уровни TSI определяются физической реализацией модуля.

6.2 Транспортный уровень интерфейса транспортного потока

На транспортном уровне интерфейса транспортного потока используется уровень транспортной системы MPEG-2. Данные через интерфейс транспортного потока передаются в формате транспортных пакетов MPEG-2. От приемника к модулю через интерфейс ТП передается мультиплекс MPEG-2, этот мультиплекс возвращается к приемнику частично или полностью дескремблированным. Если пакет не

скремблирован, то модуль возвращает его без изменений. Если пакет скремблирован и принадлежит к выбранной службе и модуль может дать доступ к этой службы, то модуль возвращает соответствующий дескремблированный пакет с флагом transport_scrambling_control, установленным в «00».

Если скремблирование выполняется на уровне пакетизированного элементарного потока (PES), то модуль реагирует аналогичным образом и при тех же условиях, что и для случая ТП, и возвращает соответствующий дескремблированный PES с флагом PES_scrambling_control, установленным в «00». Пакеты ТП и PES пакеты полностью определены в спецификации системы MPEG-2 согласно стандарту [4].

6.3 Верхние уровни интерфейса транспортного потока

Настоящий стандарт не оперирует со структурами MPEG-2, уровень которых выше уровня транспортного потока. Однако модуль выделяет из транспортного потока и обрабатывает данные, необходимые для его функционирования, — сообщения EMM и ECM.

7 Интерфейс команд — уровни транспортные и сеансовые

В соответствии с ГОСТ Р 53531 (п. 4.24.2) интерфейс команд выполняет обмен командами между абонентским приемником и модулем.

Параметры передачи данных через интерфейс команд нормируются в терминах объектов. Объекты кодируются в форме «тег-длина-величина» в соответствии с синтаксисом согласно ГОСТ Р ИСО/МЭК 8824-1, ГОСТ Р ИСО/МЭК 8825.

В таблице 1 представлены значения длины поля Length, используемые всеми блоками данных протокола на транспортном уровне, на сеансовом уровне и на уровне приложений

Таблица 1 — Поле Length, используемое всеми блоками данных протокола на транспортном уровне, сеансовом уровне и на уровне приложений

Синтаксис	Количество битов	Мнемоника
length_field() {		
size_indicator	**1**	**bslbf**
if (size_indicator = = 0)		
length_value	**7**	**uimsbf**
else if (size_indicator = = 1) {		
length_field_size	**7**	**uimsbf**
for (i=0; i<length_field_size; i++) {		
length_value_byte	**8**	**bslbf**
}		
}		
}		

В этом разделе описаны объекты ASN.1 для транспортного и сеансового уровней, которые перемещаются через интерфейс команд. Параметры всех этих объектов и для объектов уровня приложений приведены в разделе 8 настоящего стандарта, применяется кодирование поля Length, параметры которого показаны в таблице 1.

Поле **size_indicator** является первым битом поля length_field. Если size_indicator = 0, то длина поля данных кодируется в последующих 7 битах. Значение длины поля от 0 до 127 может быть закодировано в одном байте. Если длина поля превышает 127, то в поле **size_indicator** устанавливается «1». В этом случае в следующих 7 битах кодируется количество байтов в поле длины (Length). Эти следующие байты должны быть сцеплены. Любая величина поля длины (Length) до 65535 может быть закодирована в трех байтах.

Неопределенный формат длины, определенный основными правилами кодирования ASN.1 согласно ГОСТ Р ИСО/МЭК 8825, не используется.

7.1 Основной транспортный уровень

7.1.1 Введение

Транспортный уровень интерфейса команд работает поверх физического уровня — уровня канала, определяемого используемой физической реализацией. Для базовых физических реализаций карт стандарта ПК интерфейс — на уровне канала в соответствии с приложением А. Транспортный протокол

предполагает, что уровень канала надежен: данные передаются в правильном порядке, без удаления или повторения данных.

Транспортный протокол является протоколом типа «команда-ответ»: узел отправляет модулю команду, используя Блок данных команды транспортного протокола (Command Transport Protocol Data Unit; C_TPDU), и ожидает ответа от модуля в Блоке данных ответа транспортного протокола (Response Transport Protocol Data Unit; R_TPDU). Модуль не может инициировать обмен данными. Он должен ожидать запрос от узла или сначала отправить этому узлу данные. Протокол поддерживается одиннадцатью объектами транспортного уровня. Часть объектов появляется только в блоках C_TPDU от узла, некоторые только в блоках R_TPDU от модуля, и некоторые могут появиться в обоих перечисленных блоках. Функции объектов транспортного уровня:

- Create_T_C, C_T_C_Reply создают новые транспортные соединения;
- Delete_T_C, D_T_C_Reply удаляют транспортные соединения;
- Request_T_C, New_T_C позволяют модулю запрашивать узел о создании нового транспортного соединения;
- T_C_Error сигнализирует о состоянии ошибки;
- T_SB переносит информацию о состоянии модуля к узлу;
- T_RCV запрашивает ожидаемые данные от модуля;
- T_Data_More, T_Data_Last передают данные между узлом и модулем от более высоких уровней. T_Data_Last с пустым полем данных используется узлом для регулярного опроса данных от модуля, когда у узла нет данных для передачи.

C_TPDU от узла содержит только один объект транспортного протокола. R_TPDU может переносить от модуля один или два объекта транспортного протокола. Единственный объект или второй объект из двух объектов в блоке R_TPDU всегда является объектом блока T_SB.

7.1.2 Объекты транспортного протокола

Все объекты транспортного уровня содержат идентификатор транспортного соединения ИТС (transport connection identifier; TCI). ИТС представляет собой поле в 1 байт. ИТС позволяет установить одновременно до 255 активных соединений транспортного уровня на узел. Значение ИТС всегда устанавливает узел. Значение ИТС, равное 0, зарезервировано. В настоящем стандарте протокол описывается в подробностях, касающихся общей части для всех физических реализаций, параметры объектов описываются только в общих чертах. Подробности кодирования объектов определяются используемым физическим уровнем. Параметры кодирования для физической реализации стандартных карт ПК в соответствии с приложением А.

Узел должен обеспечивать создание не менее 16 транспортных соединений на поддерживаемый сокет модуля. Рекомендуется обеспечивать создание и распределение 255 соединений среди сокетов модуля.

Функции объектов транспортного уровня:

- Create_T_C создает транспортное соединение. Create_T_C выпускается узлом и переносит значение ИТС для соединения, которое будет установлено;
- C_T_C_Reply является ответом целевого модуля на Create_T_C, он переносит ИТС для создаваемого соединения;
- Delete_T_C удаляет существующее транспортное соединение. В качестве параметра используется значение ИТС соединения, которое будет удалено. Блок может быть выпущен узлом или модулем. Модуль выпускает Delete_T_C в ответ на запрос или в ответ на данные от узла;
- D_T_C_Reply является ответом на удаление. При некоторых обстоятельствах этот ответ может не достигнуть своего места назначения в связи с этим объект Delete_T_C связан с временем ожидания (тайм-аутом). Если ситуация тайм-аута назревает до получения ответа, то все меры, которые были бы предприняты для получения ответа, могут быть предприняты на интервале тайм-аута;
- Request_T_C модуль запрашивает узел о создании нового транспортного соединения. Он отправляется от модуля на существующем транспортном соединении в ответ на запрос или на данные от узла;
- New_T_C является ответом на Request_T_C. Он отправляется на том же самом транспортном соединении, как и объект Request_T_C, он переносит ИТС нового соединения. New_T_C сопровождается объектом Create_T_C для нового соединения, которое устанавливает надлежащее транспортное соединение;
- T_C_Error сигнализирует о состоянии ошибки и переносит код ошибки, определяющий ошибку, в поле размером 1 байт. В данной версии этот блок отправляется в ответ на Request_T_C;
- T_SB отправляется модулем как ответ на все объекты, полученные от узла, или добавляется к другим объектам протокола, или пересылается как собственное, в зависимости от обстоятельств. Переносится в 1 байте;

- T_RCV отправляется узлом для запроса тех данных, которые модуль хочет отправить (запрос содержится в предыдущем T_SB от модуля) к узлу;

- T_Data_More и T_Data_Last передают данные между узлом и модулем и могут быть в любом блоке C_TPDU или R_TPDU. От модуля они отправляются только в ответ на явный запрос T_RCV от узла. T_Data_More используется, если Модуль данных протокола (Protocol Data Unit, PDU) от более высокого уровня должен быть разделен на фрагменты для передачи из-за внешних ограничений на размер передаваемых данных. Он указывает на то, что после него будет отправлено не менее еще одного фрагмента PDU верхнего уровня. T_Data_Last указывает на последний или единственный фрагмент PDU верхнего уровня.

7.1.3 Транспортный протокол

На рисунках 5 и 6 показаны диаграммы изменения состояний для установления соединения и разрыва соединения на стороне узла и стороне модуля соответственно.

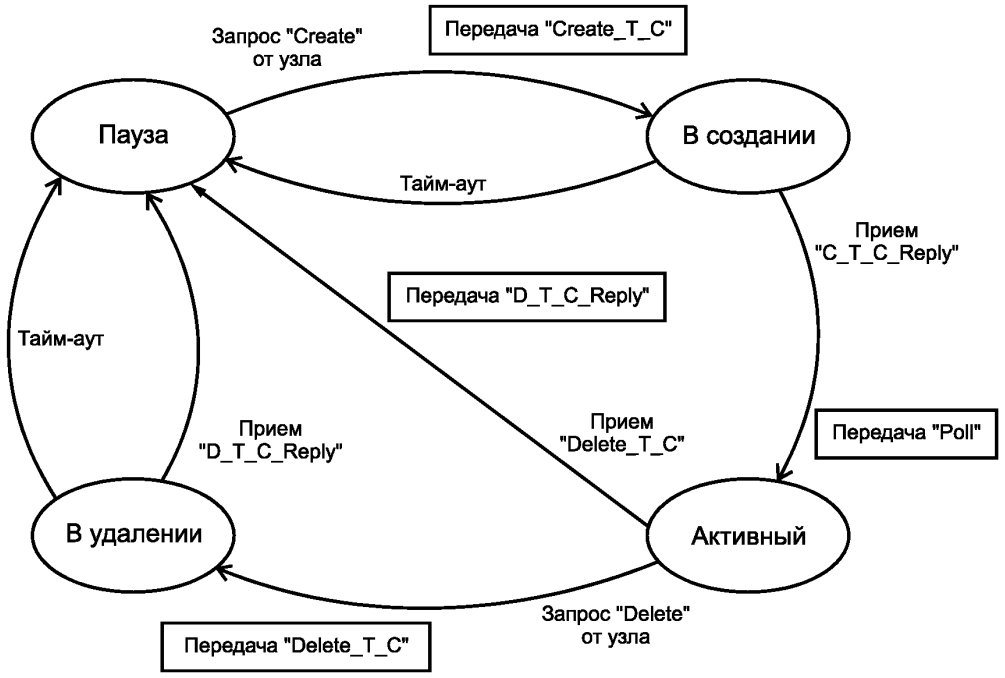

Рисунок 5 — Диаграмма состояний транспортного протокола на узле

В таблице 2 приведены ожидаемые объекты при приеме в различных состояниях транспортного соединения на стороне узла.

Таблица 2 — Ожидаемые объекты при приеме в различных состояниях транспортного соединения на стороне узла

Состояние	Ожидаемые объекты — Узел
Неактивный	Отсутствуют
В ожидании	C_T_C_Reply (+ T_SB)
Активный	T_Data_More, T_Data_Last, Request_T_C, Delete_T_C, T_SB
В удалении	D_T_C_Reply (+ T_SB)

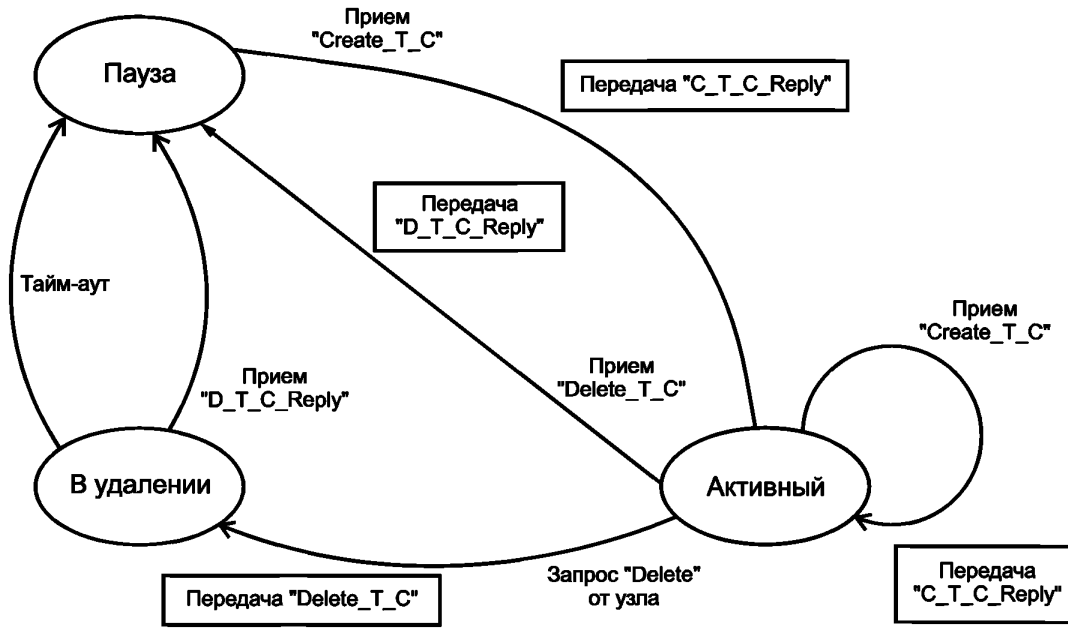

Рисунок 6 — Диаграмма состояний транспортного протокола на модуле

В таблице 3 приведены ожидаемые объекты, принимаемые модулем в различных состояниях транспортного соединения.

Таблица 3 — Ожидаемые объекты, принимаемые модулем в различных состояниях транспортного соединения

Состояние	Ожидаемые объекты — Узел
Состояние	Ожидаемые объекты — модуль
Неактивный	Create_T_C
Активный	Create_T_C, T_Data_More, T_Data_Last, New_T_C, Delete_T_C, T_RCV, T_C_Error
В удалении	D_T_C_Reply

Каждая дуга изменения состояния маркируется событием, вызвавшим этот переход. Если переход инициирует и отправление объекта, то эта причина обозначается текстом в рамке. Когда узел хочет установить транспортное соединение с модулем, он отправляет объект Create_T_C и перемещается в состояние «В создании». Модуль должен ответить непосредственно объектом C_T_C_Reply. Если после периода тайм-аута модуль не отвечает, то узел возвращается к состоянию ожидания (через дугу Тайм-аут). Узел не будет передавать или опрашивать модуль снова на этом конкретном транспортном соединении, последующие C_T_C_Reply будут проигнорированы. Если впоследствии узел снова использует тот же самый идентификатор транспортного соединения, тогда модуль снова получит Create_T_C, из этого он должен сделать вывод, что старое транспортное соединение завершено и что устанавливается новое.

Когда модуль отвечает C_T_C_Reply, узел перемещается в состояние соединения «Активный». Если узел имеет данные для передачи, то он может их передать.

Если узел хочет завершить транспортное соединение, он отправляет объект Delete_T_C и перемещается в состояние «В удалении». Тогда по получении объекта D_T_C_Reply или после тайм-аута он возвращается к состоянию «Неактивный», если ни один не принимается. Если узел получает объект Delete_T_C от модуля, он выпускает объект D_T_C_Reply и переходит непосредственно в состояние ожидания. За исключением состояния «Активный» игнорируется любой объект, принятый в любом состоянии, не относящийся к ожидаемым.

В состоянии «Активный» узел периодически выпускает запросы или отправляет данные (если у него есть PDU верхнего уровня для передачи). В ответ это получает объект T_SB, которому предшествует объект Request_T_C или Delete_T_C (если это хочет сделать модуль).

В состоянии «Активный» данные могут быть отправлены узлом в любое время. Если модуль хочет отправить данные, то он должен ожидать сообщение от узла - обычно данные или запрос - и затем указать в ответ, что у этого есть доступные данные в T_SB. Узел тогда в некоторый момент отправит модулю запрос T_RCV, на который модуль отвечает, отправляя данные в объекте T_Data. В случаях использования T_Data_More каждый последующий фрагмент должен ожидать получения другого T_RCV от узла.

Если модуль хочет установить другое транспортное соединение, то он должен передать объект Request_T_C либо в ответ на опрос или на данные. Если он может удовлетворить запрос узла, то он ответит New_T_C, содержащим идентификатор транспортного соединения для нового соединения, сразу после объекта Create_T_C для создания соединения. Если узел не может удовлетворить запрос, потому что используются все транспортные идентификаторы соединения, то он ответит T_C_Error, содержащим соответствующий код ошибки.

Пример передачи последовательности передачи объектов создания и использования транспортного соединения показан на рисунке 7.

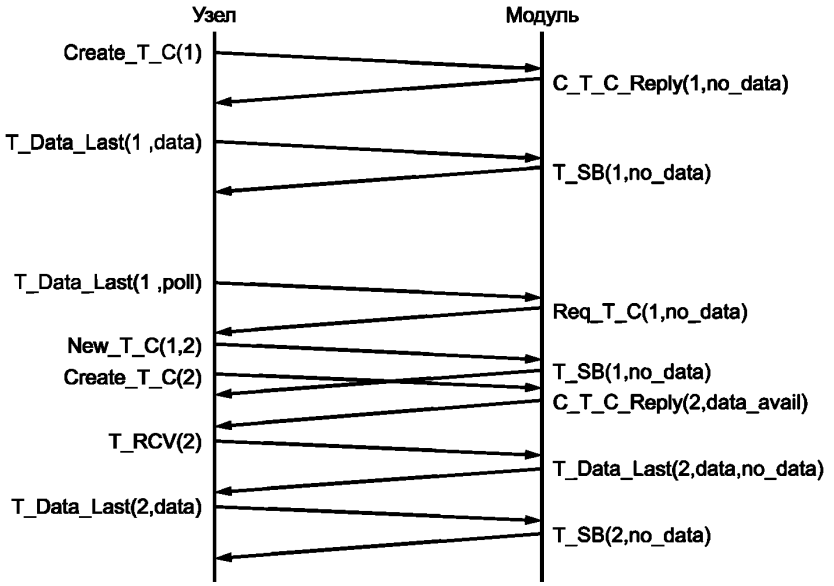

Рисунок 7 — Пример передачи объектов транспортного протокола

В этом примере предполагается, что модуль был только что включен и что было установлено физическое соединение (выполнена инициализация карты ПК). Узел выпускает объект Create_T_C для транспортного соединения номер 1. Модуль отвечает объектом C_T_C_Reply для транспортного соединения 1, указывая при этом на отсутствие данных для передачи. Узел отправляет данные с объектом T_Data_Last, и модуль отвечает объектом T_SB. Некоторое время спустя узел опрашивает модуль пустым объектом T_Data_Last, и модуль отвечает объектом Request_T_C, сообщая этим, что он хочет иметь новое транспортное соединение и указывая (в добавленном объекте T_SB), что у него нет данных для передачи на соединении 1. Узел отвечает объектом New_T_C, указывающим, что транспортное соединение 2 будет установлено, и объектом Create_T_C о транспортном соединении 2. Модуль отвечает объектом T_SB на первый и на второй объекты C_T_C_Reply, указывая, что он имеет данные для передачи

на этом соединении. Узел отвечает объектом T_RCV для получения данных, а модуль отвечает объектом T_Data_Last, содержащим данные. Узел отвечает объектом T_Data_Last с собственными данными, а модуль отвечает на это объектом T_SB об отсутствии данных для передачи.

7.2 Уровень сеанса

7.2.1 Введение

Уровень сеанса обеспечивает механизм, с помощью которого приложения взаимодействуют и используют ресурсы. Ресурс является механизмом инкапсуляции функциональности на уровне приложений. Параметры ресурса описаны полностью в 8.2 настоящего стандарта.

Ресурсы различаются по количеству сеансов, которые они могут одновременно поддерживать. Некоторые ресурсы поддерживают только один сеанс. Если второе приложение пытается запросить сеанс к такому ресурсу, уже находящемуся в режиме использования, тогда оно получит ответ «ресурс занят». Другие ресурсы могут поддерживать одновременно больше одного сеанса, тогда запросы ресурса будут поддерживаться до некоторого предела, определенного ресурсом. Примером последнего может быть ресурс дисплея, который в некоторых реализациях узла может поддерживать сеансы одновременного отображения в различных окнах.

7.2.2 Объекты протокола сеанса

В состав объектов протокола сеанса входят объекты:

- open_session_request;
- open_session_response;
- create_session;
- create_session_response;
- close_session_request;
- close_session_response;
- session_number.

Общее описание перечисленных объектов протокола сеанса должно быть в соответствии со стандартом [5] (7.2.2).

7.2.3 Протокол сеанса

На уровне сеанса диалог между модулем или узлом инициируется модулем или узлом. Примеры диалогов, когда модуль запрашивает ресурс, который обеспечивает узел, и когда модуль А запрашивает ресурс, предоставляемый модулем В, представлены в стандарте [5] (7.2.3, рисунки 9 и 10).

7.2.4 Структура модуля данных протокола сеанса (SPDU)

Для обмена данными на уровне сеанса между узлом и модулем или между модулем и узлом используется модуль данных SPDU.

Структура SPDU показана на рисунке 8. Параметры кодирования SPDU показаны в таблице 4.

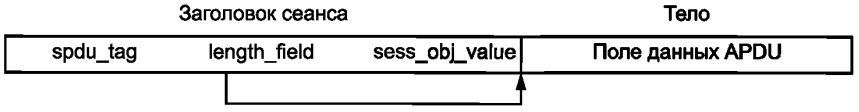

Рисунок 8 — Структура SPDU

Таблица 4 — Параметры кодирования SPDU

Синтаксис	Количество битов	Мнемоника

SPDU () {
spdu_tag
length_field ()
for (i=0; i<length_value; i ++) {
session_object_value byte
}
for (i=0; i<N; i ++) {
apdu ()
}
}
	8
	
	8
		Uimsbf
		
		uimsbf

SPDU состоит из двух частей:

- обязательного заголовка сеанса, состоящего из значения тега spdu_tag, поля length_field, в котором закодирована продолжительность сеанса, значения поля объекта и значения объекта сеанса. Объекты сеанса описываются в стандарте [5] (7.2.6). Поле длины не включает длину любого следующего APDU;

- условного тела переменной длины, которое содержит целое число APDU, принадлежащих тому же самому сеансу. Присутствие тела зависит от заголовка сеанса.

7.2.5 Транспортировка SPDU

SPDU транспортируется в поле данных одного или нескольких модулей данных транспортного протокола TPDU. Дополнительная информация содержится в описании TPDU каждой физической реализации модуля.

7.2.6 Описание заголовков сеанса

Объекты Open Session Request, Open Session Response, Create Session, Create Session Response, Close Session Request, Close Session Response, Session Number являются объектами заголовка сеанса. Только за заголовком SPDU следует поле данных — объект session_number, — который всегда сопровождается телом SPDU, содержащим несколько модулей данных протокола приложения APDU. Параметры кодирования и семантика этих объектов должны быть в соответствии со стандартом [5] (7.2.6).

7.2.7 Кодирование тегов сеанса

Таблица 5 содержит имена объектов, используемых уровнем сеанса в интерфейсе команд. Кодирование spdu_tag выполняется по правилам ASN.1. Каждый spdu_tag размещается в одном байте.

Таблица 5 — Имена объектов, используемых уровнем сеанса в интерфейсе команд

spdu_tag	Величина тега в шестнадцатиричной системе	Примитив или сконструирован	Направление передачи узел <- -> модуль
Topen_session_request	'91'	Примитив	<- - -
Topen_session_response	'92'	Примитив	- - ->
Tcreate_session	'93'	Примитив	- - ->
Tcreate_session_response	'94'	Примитив	<- - -
Tclose_session_request	'95'	Примитив	<- - ->
Tclose_session_response	'96'	Примитив	<- - ->
Tsession_number	'90'	Примитив	<- - ->

Величины в диапазонах 80-8F, 90-9F, A0-AF, B0-BF зарезервированы.

8 Интерфейс команд — уровень приложений

8.1 Введение

Уровень приложений реализует набор протоколов, основанных на понятии ресурса. Ресурс определяется как единица функциональных возможностей, которая доступна приложениям, работающим на модуль. Каждый ресурс поддерживает набор объектов и протокол для обмена при использовании ресурса. Связь с ресурсом осуществляется в сеансе, создаваемом для конкретного ресурса. Этот раздел содержит описание минимального набора ресурсов, который должны обеспечивать все узлы, совместимые с этой спецификацией. Некоторые ресурсы, указанные в разделе 8 настоящего стандарта, перед именем имеют префикс DVB. Эти ресурсы используют специфичные для DVB функции, которые могут не присутствовать в ситуациях, в которых этот интерфейс может использоваться. Ресурсы с префиксом DVB должны быть обеспечены в любом DVB совместимом узле и опционально могут быть предусмотрены в других узлах. Другие дополнительные ресурсы приведены в приложении Б.

8.2 Ресурсы

8.2.1 Введение

Ресурс может быть предоставлен непосредственно узлом, ресурс может находиться в модуле. Ресурс определяется идентификатором ресурса. Идентификатор ресурса состоит из трех компонентов: класса ресурса, типа ресурса и версии ресурса.

Класс ресурса определяет набор объектов и протоколы для их использования.

Тип ресурса определяет различные модули ресурса в пределах класса. Все типы ресурса в пределах класса используют одни и те же объекты и протокол, но предлагают различные службы или являются различными экземплярами одной и той же службы.

Версия ресурса позволяет узлу идентифицировать последнюю версию (с самым высоким номером версии) ресурса, когда присутствует больше одной версии одного и того же класса и типа. Это позволяет предоставлять модулю обновленные или улучшенные ресурсы для замены существующих ресурсов на узле или на модуле. Ресурсы с более высокими номерами версий должны иметь обратную совместимость с предыдущими версиями так, чтобы у приложений, запрашивающих предыдущую версию, был ресурс ожидаемого поведения.

Ресурсы используются приложением, создающим сеанс к ресурсу согласно 7.2 настоящего стандарта. В процессе инициализации, выполненным менеджером ресурсов, узел идентифицирует все доступные ресурсы, которые он обеспечит самостоятельно или на другом модуле, и может завершить сеанс в нужном месте. В случае создания сеанса приложение может использовать ресурс обмена объектами согласно определенному протоколу.

Пример использования концепции ресурса — класс низкоскоростного коммуникационного ресурса. Этот класс определяет общий механизм передачи данных использованием модема или обратного канала кабельной системы. Типы ресурса определяются для различных скоростей модема в соответствии с идентификацией номера Рекомендаций ITU-T [6]—[8], например, V.21, V.22, V.32bis. Современные модемы предлагают широкий диапазон скоростей. Это позволяет одному модему предлагать несколько типов ресурса так, что выбор скорости выполняется при создании сеанса для определенного типа ресурса, предлагаемого модемом.

8.2.2 Идентификатор ресурса

Идентификатор ресурса состоит из 4 байтов. Два старших значащих бита первого байта указывают, является ли ресурс общедоступным или частным и, следовательно, указывают структуру остальной части поля. Значения 0, 1 и 2 соответствуют общедоступному ресурсу. Значение 3 соответствует частному ресурсу.

Классы общедоступных ресурсов имеют значения в диапазоне от 1 до 49150. Поле resource_id_type является старшей значащей частью resource_class. Значение 0 резервируется. Максимальные значения всех полей (единицы) зарезервированы. Частные ресурсы идентифицируются частным определителем — организацией, которая определяет частный ресурс. Каждый частный определитель ресурса может определить структуру и контент поля private_resource_identity, которое он выбирает, за исключением тех значений, которые зарезервированы. Параметры кодирования поля resource_identifier представлены в таблице 6.

Таблица 6 — Кодирование resource_identifier

Синтаксис	Количество битов	Мнемоника
resource_identifier () {		
resource_id_type	**2**	**uimsbf**
if (resource_id_type! = 3) {		
resource_class	**14**	**uimsbf**
resource_type	**10**	**uimsbf**
resource_version	**6**	**uimsbf**
}		
else {		
private_resource_definer	**10**	**uimsbf**
private_resource_identity	**20**	**uimsbf**
}		
}		

8.2.3 Приложения и провайдеры ресурса

Приложения и провайдеры ресурса являются двумя типами объекта уровня приложений, который может находиться на модуле. Приложения используют ресурсы для выполнения задач пользователя узла. Провайдеры ресурса предоставляют ресурсы в дополнение к тем доступным ресурсам непосредственно в узле или более новые версии ресурса, заменяющие один ресурс, ранее предоставленный узлом, или предоставленный на другом модуле. Устранение тупиковых проблем и сложностей инициализации провайдеров ресурса должно обеспечиваться менеджером ресурсов.

8.3 Модули данных протокола приложения

8.3.1 Введение

Все протоколы на уровне приложений используют общую структуру Модуля данных протокола приложений (Application Protocol Data Unit; APDU) для передачи данных приложений между модулем и узлом или между модулями.

Структура APDU показана на рисунке 9. Параметры кодирования APDU представлены в таблице 7.

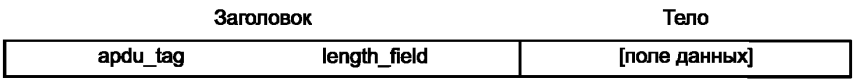

Рисунок 9 — Структура APDU

Таблица 7 — Параметры кодирования APDU

Синтаксис	Количество битов	Мнемоника

APDU () {
apdu_tag
length_field ()
for (i=0; i <length_value; i ++) {
data_byte
}
}
	24
	
	8
		uimsbf
		
		uimsbf

APDU состоит из двух частей:

- обязательного заголовка, содержащего значение тега apdu_tag, указывающего, какой параметр отправляется в поле данных, и значение закодированной длины следующего поля данных. Кодирование apdu_tag и length_field согласно основным правилам кодирования ASN.1 согласно ГОСТ Р ИСО/МЭК 8825;

- условного тела переменной длины, равной длине поля length_field.

8.3.2 Объединение в цепочку полей данных APDU

Поле данных, находящееся в APDU, может быть разделено на несколько блоков меньших размеров, если этого требуют размеры буфера узла или модуля в процессе передачи и приема. Этот механизм применим только для случаев нескольких APDU. Объединение в цепочку выполняется при использовании двух различных значений apdu_tag (M_apdu_tag и L_apdu_tag). Все поле блоков данных, кроме последнего, отправляется в APDU со значением M_apdu_tag. Это значение тега указывает то, что в другом APDU будет отправлено больше данных. Последнее поле блока данных отправляется на интервале передачи APDU со значением L_apdu_tag. После приема последнего блока принимающий объект связывает все принятые поля данных.

Применение этого механизма допустимо для всех APDU с двумя значениями тега (M_apdu_tag и L_apdu_tag). Механизм формирования цепочки приведен в стандарте [5] (рисунок 13).

8.3.3 Транспортировка модуля APDU в SPDU

Целочисленное количество модулей APDU, принадлежащих одному и тому же сеансу, может быть транспортировано в теле одного модуля SPDU.

8.4 Система управления ресурсами

8.4.1 Менеджер ресурса

Функцию менеджера ресурса выполняет ресурс, предоставляемый узлом. В классе существует только один тип ресурса, который может поддерживать любое количество сеансов. Он контролирует приобретение и предоставление ресурсов для всех приложений. Между модулем и узлом используется симметричный протокол связи для определения ресурсов, которые модуль и узел могут обеспечить. Протокол используется сначала узлом для опроса каждого транспортного соединения и определения ресурсов, которые представляются ими. Затем протокол используется приложениями для обнаружения полных доступных ресурсов. Эта операция выполняется периодически и в случаях, когда ресурсы изменяются, и для обновления представления о доступных ресурсах. Менеджер ресурсов обеспечивает узел, он не может быть заменен ресурсом модуля. Узел должен игнорировать попытки менеджера ресурсов предоставления ресурса от модуля.

Протокол менеджера ресурсов выполняет обмен данными с модулем при использовании следующих объектов: Запрос Профиля; Ответ Профиля; Измененный Профиль. Протокол менеджера ресурсов функционирует в соответствии со стандартом [5] (8.4.1.1). Параметры кодирования объектов: Запрос Профиля, Ответ Профиля, Измененный Профиль должны быть в соответсвии со стандартом [5] (8.4.1.2—8.4.1.4).

8.4.2 Ресурс информации приложения

Ресурс информации приложения позволяет приложениям предоставлять узлу стандартный набор информации о своих возможностях. Подобно менеджеру ресурсов ресурс информации приложения обеспечивает только узел и не имеет ограничений сеанса. Все приложения создают сеанс к этому ресурсу после завершения инициализации объекта Запрос Профиля. Узел после этого отправляет объект Запрос Информации Приложения приложению, которое отвечает, возвращая объект Информация Приложения с конкретной информацией. Сеанс сохраняется так, чтобы узел мог в любое время сигнализировать приложение, чтобы создать сеанс MMI в точке его входа в меню высокого уровня. Это выполняется с помощью объекта Ввод Меню. Приложение после получения объекта Ввод Меню должно создать сеанс MMI в соответствии со стандартом [5] (8.6) и вывести на экран меню высокого уровня. Узел должен гарантировать возможность создания сеанса при передаче объекта Ввод Меню.

Параметры кодирования объектов Информация Приложения: Запрос Информации Приложения, Информация Приложения, Ввод Меню должны быть в соответствии со стандартом [5] (8.4.2.1—8.4.2.3).

8.4.3 Ресурс поддержки ограниченного доступа (СОД)

Этот ресурс предоставляет набор объектов, поддерживающих приложения ограниченного доступа СОД:

- Запрос информации СОД;
- Информация СОД;
- Выбор служб для дескремблирования;
- CA_PMT;
- Ответ PMT CA.

Как и менеджер ресурсов, ресурс поддержки условного доступа обеспечивается только узлом без ограничения длительности сеанса. Все приложения СОД создают сеанс к этому ресурсу после завершения инициализации ресурса информации приложения. Узел отправляет объект Запрос информации СОД приложению, которое отвечает, возвращая объект Информация СОД с соответствующей информацией. Сеанс в этом случае сохраняется открытым для функционирования протокола, связанного с таблицей PMT СОД и объектами Ответа PMT CA. Параметры кодирования и семантики объектов: Запрос информации СОД, Информация СОД, Выбор служб для дескремблирования, CA_PMT, Ответ PMT CA должны быть в соответствии со стандартом [5] (8.4.3.1—8.4.3.5).

8.5 Ресурсы управления узлом и информацией

8.5.1 Управление узлом DVB

Ресурс управления узлом DVB дает приложению возможности ограниченного управления функционированием узла. Эти возможности включают две основные функции:

- первая функция: перенастройка узла на другую службу, связанную с изменением частоты или настройкой на другой спутник. При этом контекст состояния предыдущей настройки не сохраняется;
- вторая функция: должна обеспечиваться временная замена одной службы на другую. Замена может быть кратковременной, на время рекламы или более продолжительной на время целой программы.

Первая функция поддерживается объектом Настройка. Вторая функция поддерживается объектами Замена и Отмена Замены. Предусмотрен также объект Запрос Блокирования для узла с запросом для приложения закрыть сеанс. Управление ресурсом узла DVB обеспечивает только узел при поддержке за один раз только одного сеанса.

Параметры кодирования и семантики объектов: Настройка, Замена, Отмена Замены и Запрос Блокирования должны быть в соответствии со стандартом [5] (8.5.1.1—8.5.1.4).

8.5.2 Ресурс дата и время

Ресурс дата и время обеспечивается узлом, при этом может поддерживаться неограниченное количество сеансов. Приложение создает сеанс к ресурсу и затем запрашивает текущее время с объектом Дата-Время. Если значение response_interval равно нулю, то в ответ передается сразу единственный объект date_time. Если response_interval не равно нулю, то сразу передается ответ — объект date_time, сопровождаемый далее объектами date_time через интервал, определяемый величиной поля respons_interval. В узле, совместимом с DVB, предоставленное время извлекается из таблицы времени и даты (Time and Date Table; TDT) в информации о SI согласно стандарту [9].

Параметры кодирования и семантики объектов: Запрос Дата-Время и Дата-Время должны быть в соответствии со стандартом [5] (8.5.2.1, 8.5.2.2).

8.6 Ресурс интерфейса человек — машина

8.6.1 Введение

Ресурс MMI поддерживает взаимодействие пользователя с дисплеем и клавиатурой. Нормируются два уровня взаимодействия. Режим MMI низкого уровня обеспечивает точное управление аспектами взаимодействия, включая прием непосредственно от пользователя кодов клавиш удаленного управления, и возможность детализированного управления планировкой и другими атрибутами изображения на экране. Режим MMI высокого уровня обеспечивает приложение управления объектами с высокоуровневой семантикой, включая меню и списки. В этом случае узел определяет наглядность и восприятие изображения. Смешение режимов MMI в одном и том же сеансе не допускается. Возможность поддержки узлом более одного сеанса MMI одновременно определяется проектным решением узла.

Кодирование текста, предоставляемого объектами дисплея в режиме высокого уровня, выполняется согласно стандарту [9]. Текст, предоставляемый объектами дисплея в режиме низкого уровня, кодируется согласно стандарту [10].

8.6.2 Объекты, используемые в режимах диалога высокого и низкого уровня MMI

Параметры кодирования и семантики объектов: Закрыть MMI, Управление Дисплеем, Ответ Дисплея должны быть в соответствии со стандартом [5] (8.6.2.1—8.6.2.3).

8.6.3 Объекты клавиатуры в режиме диалога низкого уровня MMI

Параметры кодирования и семантики объектов, применяемых только в режиме диалога низкого уровня MMI: Управление Клавиатурой, Нажатие Клавиши, Таблица Кодов Клавиши должны быть в соответствии со стандартом [5] (8.6.3.1—8.6.3.3).

8.6.4 Объекты дисплея низкого уровня MMI

Режим низкого уровня дисплея MMI основан на использовании механизма субтитров DVB. Параметры кодирования и семантики объектов дисплея низкого уровня MMI: Доставка Данных Субтитров DVB, Инкапсуляция Сегмента, Сообщения Системы Отображения, Управление Временем, Загрузка Объекта, Модель Декодера Субтитров для Общего Интерфейса должны быть в соответствии со стандартом [5] (8.6.4.1—8.6.4.6).

8.6.5 Объекты режима высокого уровня MMI

Объекты, перечисленные ниже, используются только в режиме высокого уровня MMI. Объекты режима высокого уровня MMI определяют требуемые операции, они позволяют узлу определять формат и тип изображения. Приложение может использовать символы управления в тексте, однако узел может их проигнорировать. Символы управления могут использоваться узлом для представления меню.

Содержание любого текста интерпретируется в соответствии с текущим набором символов. Представление информации узлом не ограничивается использованием устройства отображения.

8.6.5.1 Объект Текст

Параметры кодирования и семантики объекта Текст должны быть в соответствии со стандартом [5] (8.6.5.1).

8.6.5.2 Объекты Enq и Answ

Объекты Enq и Answ позволяют приложению запрашивать ввод данных пользователем. Оператором запроса информации может быть только PIN-код пользователя. Ответ на Enq возвращается узлом в объекте Answ. Объект Enq позволяет приложению определить, нужно ли отображать введенные данные пользователю узла. Например, числа PIN-кода, вводимые пользователем, не должны выводиться на экран. Параметры кодирования объекта Enq представлены в стандарте [5] (8.6.5.2).

8.6.5.3 Объект Answ

Этот объект используется вместе с объектом Enq для управления вводом данных пользователем. Параметры кодирования и семантики объекта Answ представлены в стандарте [5] (8.6.5.3).

8.6.5.4 Объект Меню

Этот объект используется в соединении с объектом Меню Answ для управления меню в режиме высокого уровня MMI. Параметры кодирования объекта Меню представлены в стандарте [5] (8.6.5.4).

8.6.5.5 Объект Меню Answ

Этот объект используется в соединении с объектом Меню для возврата выбора пользователя и с объектом Список для индикации завершения пользователем работы с этим объектом. Параметры кодирования и семантика объекта Меню Answ представлены в стандарте [5] (8.6.5.5).

8.6.5.6 Объект Список

Этот объект используется для отправки списка элементов, которые будут выведены на экран, например, во время консультации о праве доступа. Синтаксис объекта аналогичен синтаксису объекта

Меню и используется в соединении с объектом Menu_answ. Параметры кодирования объекта Список представлены в стандарте [5] (8.6.5.6).

8.7 Ресурсы связи

8.7.1 Класс низкоскоростного ресурса связи

8.7.1.1 Введение

Интерфейс класса низкоскоростного ресурса связи (Low-Speed Communications Resource; LSCR) является частью узла. Этот интерфейс обеспечивает двунаправленную связь, используя, например, телефонную линию или обратный канал кабельной сети. Этот интерфейс может быть использован для поддержки функций условного доступа и интерактивных служб.

Класс ресурса определяется в общем виде. Это позволяет использовать различные базовые технологии связи с общими наборами объектов. Типы ресурса определяются в классе поддержки передачи к определенным типам устройств при использовании общих наборов объектов. Модель представляет собой двунаправленный канал (полный дуплекс), по которому возможна передача произвольных данных. Управление потоками применяется в обоих направлениях между приложением и узлом. Данные для передачи разделяются на сегменты для ограничения размеров буферов, требуемых в обоих приложениях и в узле. Ограничения на количество буферов в обоих приложениях и в узле накладываются протоколом управления потоками.

Модули APDU, описанные ниже, используются низкоскоростными функциями связи узла и приложения.

8.7.1.2 Требования класса низкоскоростного ресурса связи

Требования класса низкоскоростного ресурса связи:

- длина буфера в каждом направлении должна быть равной 254 байтов. Допускается уменьшение указанной нормы;

- должно выполняться управление потоками между узлом и приложением. Протокол должен обрабатывать данные, входящие от интерфейса связи. Узел должен обеспечивать управление потоками от соединений, установленных через линии внешней связи, скоординированное с управлением потоками между приложением и узлом.

8.7.1.3 Объекты поддержки низкоскоростного ресурса связи

В настоящем стандарте определены параметры четырех объектов LSCR:

- Comms Cmd (Команда);
- Comms Reply (Ответ);
- Comms Send (Передача);
- Comms Rcv (буфер данных, принятых от коммуникационного ресурса).

Объект Comms Cmd (Команда) отправляется приложению и позволяет выполнить несколько операций управления на ресурсе связи. Объект Comms Reply отправляется узлом, он подтверждает получение объекта Comms Cmd. Объект Comms Send содержит буфер данных, отправленных на ресурс связи (в линию). Объект Comms Rcv представляет собой буфер данных, полученный от ресурса связи (от линии). Размер буфера и величина времени ожидания тайм-аута в принятых буферах соответствуют размеру буфера и значению времени ожидания, установленным ранее в объекте Comms Cmd.

Для использования конкретного ресурса связи с этим ресурсом должен создаваться сеанс с применением стандартного механизма создания сеанса. Операции с использованием объектов LSCR выполняются в рамках этого сеанса.

Детализированные параметры кодирования объектов LSCR: Comms Cmd, Comms Reply, Comms Send и Comms Rcv должны быть в соответствии со стандартом [5] (8.7.1.4—8.7.1.7).

8.8 Идентификаторы ресурса и теги объекта приложения

8.8.1 Идентификаторы ресурса

Идентификаторы общедоступных ресурсов приведены в таблице 8.

Таблица 8 — Значения идентификатора ресурса

Ресурс	Класс	Тип	Версия	Идентификатор ресурса
Менеджер ресурса	1	1	1	00010041
Информация приложения	2	1	1	00020041

Окончание таблицы 8

Ресурс	Класс	Тип	Версия	Идентификатор ресурса
Поддержка условного доступа	3	1	1	00030041
Управление узлом	32	1	1	00200041
Дата-Время	36	1	1	00240041
MMI	64	1	1	00400041
Низкоскоростная связь	96	В соответствии с 8.8.1.1 настоящего стандарта	1	0060xxx1
Зарезервировано	Другие значения	Другие значения	Другие значения	Другие значения

8.8.1.1 Типы низкоскоростных ресурсов связи

Тип низкоскоростного ресурса связи кодируется в двух полях. Первое поле использует биты 0 и 1 и идентифицирует номер конкретного устройства. Второе поле использует биты 2—9, описывает соответствующий тип устройства и разделяется на субполя для определенных устройств, например, модемов. Структура кодирования типа низкоскоростного ресурса связи приведена на рисунке 10.

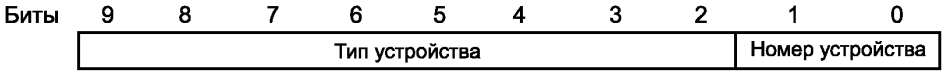

Рисунок 10 — Структура кодирования типа низкоскоростного ресурса связи

Кодирование поля Тип устройства выполняется в соответствии с таблицей 9.

Т а б л и ц а 9 — Кодирование поля Тип устройства

Описание	Значение
Модемы — смотри ниже	00—3F
Последовательные порты	40—4F
Обратный кабельный канал	50
Зарезервировано	51—FF

Тип ресурса для модемов кодируется в трех полях. Номер устройства кодируется аналогично описанному выше. Поле Тип устройства разделяется на субполе dataproc кодирования для требуемой обработки данных и субполе, указывающее тип модема, обозначенное согласно спецификации скорости при использовании номеров стандартов ITU-T Рекомендации серии V.

Структура поля Тип ресурса для модемов приведена на рисунке 11.

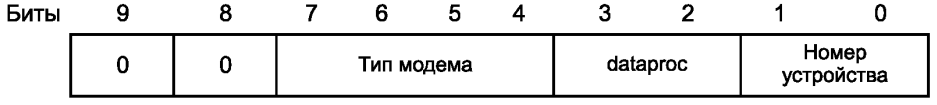

Рисунок 11 — Структура поля Тип ресурса для модемов

Кодирование поля dataproc выполняется в соответствии с таблицей 10.

Таблица 10 — Кодирование поля dataproc

Описание	Значение
По согласованию	00
Серия V.42bis не применяется	01
Обработка данных не применяется	10
Зарезервировано	11

Поле «По согласованию» означает использование двух версий: устранение ошибки V.42 и сжатия данных V.42bis при условии, если обе версии обеспечиваются на дальнем конце линии связи. В случае «V.42bis не применяется» согласовывается использование устранения ошибки в соответствии с V.42, но не используется сжатие данных V.42bis.

Поле Типа модема кодируется в соответствии с таблицей 11.

Таблица 11 — Кодирование поля Типа модема

Описание	Значение
Зарезервировано	0000
Зарезервировано	0001
V.21 (300 бит/с)	0010
Зарезервировано	0011
V.22 (1200 бит/с)	0100
V.22bis (2400 бит/с)	0101
V.23 (1200/75 бит/с)	0110
Зарезервировано	0111
V.32 (9600/4800 бит/с)	1000
V.32bis (14.4 кбит/с)	1001
V.34 (28.8 кбит/с)	1010
Зарезервировано	1011—1101
V.27ter	1110
V.29	1111

Если конкретный модем, используемый узлом, предлагает несколько скоростей в соответствии со стандартов ITU-T Рекомендации серии V, то узел должен предложить доступные для использования скорости модема, как различные типы ресурса. При использовании одного определенного типа ресурса все другие предлагаемые типы ресурса будут представлены занятыми.

8.8.2 Теги объекта приложения

Кодирование тега apdu_tag выполняется по правилам ASN.1. Каждый apdu_tag кодируется в трех байтах. В 24 битах из каждого apdu_tag 10 битов устанавливаются по правилам ASN.1 как описано на рисунке 12. Используются только примитивные теги.

Байт 1 b24 b17	Байт 2 b16 b9	Байт 3 b8 b1
10011111	1xxxxxxx	0xxxxxxx

Рисунок 12 — Кодирование примитивного тега

Значения тега объекта приложения в соответствии со стандартом [5] (таблица 58).

Приложение А
(обязательное)

Базовый физический уровень карты ПК

А.1 Общее описание

Параметры физического интерфейса между модулем и узлом, параметры конструкции и механические характеристики модуля соответствуют спецификации карты ПК согласно спецификациям [1]—[3]. На рисунке А.1 показана типичная архитектура модуля.

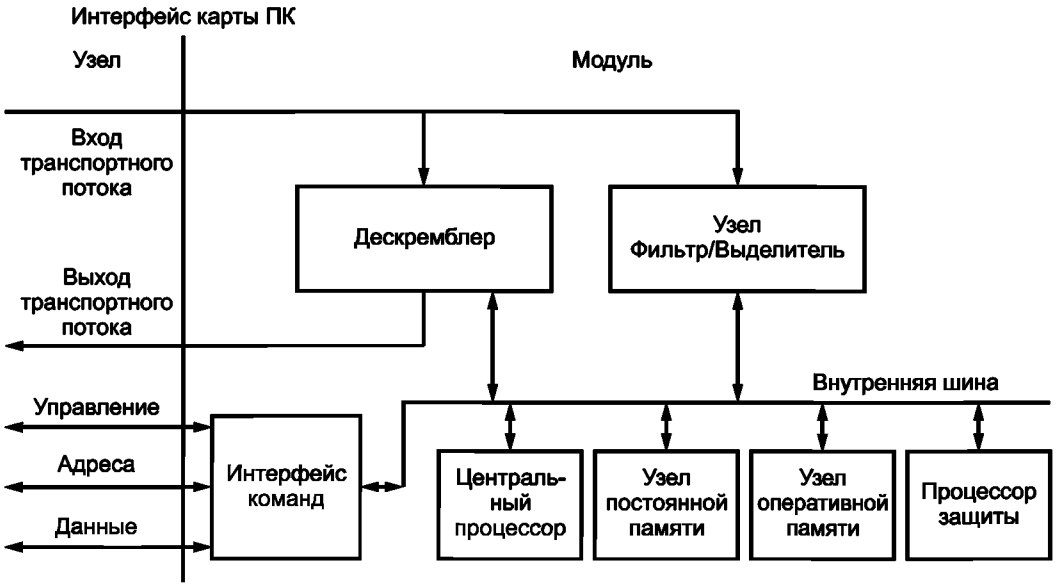

Рисунок А.1 — Типичная архитектура модуля

А.1.1 Интерфейс карты ПК

В соответствии с настоящим стандартом карта ПК имеет следующие интерфейсы:

- интерфейс транспортного потока данных MPEG-2 — на 8-разрядном параллельном входе в модуль и отдельный 8-разрядный параллельный выход, вместе с сигналами управления и байтовой синхронизации;
- интерфейс команд для трафика команд — между узлом и модулем, состоящий из 8-разрядной двунаправленной шины данных вместе с адресом и сигналами управления;
- интерфейс атрибутов памяти — для чтения структуры данных о плате модуля и для настройки модуля в нормальный режим работы.

А.1.2 Дескремблер

Дескремблер избирательно дескремблирует транспортные пакеты транспортного потока. При соответствующей конфигурации он выполняет дескремблирование пакетизированных элементарных потоков. Конфигурирование дескремблера выполняется конкретными значениями слов управления, периодически загружаемыми в него центральным процессором. Для одновременного дескремблирования нескольких служб он обычно поддерживает список слов управления, связанных с PID, которые должны быть дескремблированы. Дескремблер не будет пытаться дескремблировать любые транспортные пакеты, в которых transport_ флаг scrambling_control устанавливается в «00», даже если PID соответствует текущему слову управления. Такое же ограничение применяется, если флаг PES_scrambling_control устанавливается в «00», при скремблировании на уровне PES.

А.1.3 Узел фильтр/извлечение

Часть данных, необходимых для работы системы условного доступа, переносятся в транспортном потоке. Узел фильтр/извлечение извлекает данные, необходимые модулю условного доступа для дескремблирования программ/служб.

А.1.4 Центральный процессор

Центральный процессор выполняет процесс условного доступа и организует поток данных в модуле и между модулем и узлом для выполнения функций условного доступа.

А.1.5 Узлы ROM/EPROM и RAM/NVRAM

Узлы ROM/EPROM и RAM/NVRAM содержат программы и данные, реализующие процесс условного доступа. Узел ROM может также содержать атрибут памяти, которая является необходимым элементом процесса инициализации карты ПК.

А.1.6 Процессор защиты

Процессор защиты выполняет такие функции безопасности, как дешифрование и хранение защищаемой информации (ключи и права доступа). Процессор защиты может быть встроен в модуль карты ПК или он может находиться на соответствующем съемном модуле (смарт-карта).

А.2 Электрический интерфейс

Электрические характеристики этого интерфейса соответствуют версии 2.1 карты ПК, представленной в стандарте [5] (приложение А, А.5).

А.2.1 Интерфейс транспортного потока

Данные MPEG-2 от узла передаются модулю по 8-разрядной шине данных MDI0 к MDI7. Данные содержат два управляющих сигнала MISTRT и MIVAL. Они синхронизируют в модуле MCLK сигнал тактовой частоты. Данные MPEG-2 возвращаются из модуля по другой 8-разрядной шине данных MDO0 к MDO7. Данные содержат два управляющие сигнала MOSTRT и MOVAL. Подробное описание работы этого интерфейса представлено в стандарте [5] (приложение А, А.5).

А.2.2 Интерфейс команд

А.2.2.1 Описание аппаратного интерфейса

Аппаратный интерфейс состоит из нескольких регистров, занимающих 4 байта адресного пространства интерфейса карты ПК. Детализированное описание аппаратного интерфейса, процессов инициализации, обмена командами между модулем и узлом — в соответствии со стандартом [5] (приложение А, А.2.2.1).

А.3 Канальный уровень

А.3.1 Интерфейс транспортного потока

Интерфейс транспортного потока не относится к канальному уровню.

А.3.2 Интерфейс команд

На канальном уровне интерфейса команд выполняются две операции:

- фрагментация модулей данных (TPDU) для передачи на буфер физического уровня ограниченного размера и последующего сбора полученных фрагментов;

- передача нескольких транспортных соединений на одно соединение канального уровня чередованием фрагментов всех транспортных соединений, которые пытаются отправить модули TPDU на канальный уровень. При этом предполагается, что механизм передачи физического уровня надежен и он сохраняет данные в правильном порядке без удаления и повторения. Описание операций канального уровня в соответствии со стандартом [5] (приложение А, А.3.2).

А.4 Конкретные реализации транспортного подуровня интерфейса платы ПК

Описание объектов транспортного протокола и их взаимодействия должно быть в соответствии со стандартом [5] (4.1).

А.5 Платы ПК для использования совместимыми узлами и модулями

А.5.1 Цели

Приведена информация, необходимая для интерпретации требований к платам ПК согласно спецификациям [1]—[3]; формирует минимально-необходимую совокупность требований, которые должны использоваться при разработке узлов и модулей для обеспечения их совместимости.

А.5.2 Терминология

В этом разделе используются числа шестнадцатеричной системы счисления. Для их обозначения используются десятичные цифры от 0 до 9, для заключительных 6 цифр от 10 до 15 используются латинские символы от A до F, после которых ставят символ «h», например 7FFh.

А.5.3 Физические характеристики

А.5.3.1 Размеры карты

Узел должен поддерживать платы ПК типа I и типа II в соответствии со спецификацией [2] (раздел 3). Поддержка узлом плат типа III является опциональной.

А.5.3.2 Соединитель

Применяется в соответствии со спецификацией [2] (раздел 4).

А.5.3.3 Управление платами ПК

Применяется в соответствии со спецификацией [2] (раздел 5).

А.5.3.4 Заземление и защита от электромагнитных помех

Применяется в соответствии со спецификацией [2] (раздел 6).

А.5.3.5 Надежность соединителя

Применяется в соответствии со спецификацией [2] (раздел 7).

А.5.3.6 Прочность соединителя в условиях воздействий неблагоприятных условий окружающей среды

Применяется в соответствии со спецификацией [2] (8.2).

А.5.3.7 Плата ПК при воздействии внешней среды

Применяется в соответствии со спецификацией [2] (раздел 9). Модуль должен сохранять работоспособность при 55 °С.

А.5.4 Требования к электрическим параметрам

Модуль Общего интерфейса реализуется как 16-разрядный электрический интерфейс платы ПК в соответствии со спецификацией [1] (раздел 6). Интерфейс команды использует младший значащий байт шины данных вместе с нижней частью адресной шины (А0-А14) и соответствующими управляющими сигналами. Интерфейс команды работает в режиме интерфейса ввода-вывода.

После первого включения и до конфигурации модуль, соответствующий этим требованиям, должен вести себя как устройство памяти с ограничениями в соответствии со стандартом EN [5] (приложение А, А.5.5).

Типы плат ПК должны быть в соответствии со стандартом [5] (приложение А, А.5.5.1).

Назначение контактов должно выполняться в соответствии со стандартом [5] (приложение А, А.5.5.2).

Требования к функциям 16-разрядной платы ПК должны быть в соответствии со спецификацией [1] (4.3) с дополнительной рекомендацией в соответствии со стандартом [5] (приложение А, А.5.5.3).

Описание сигнала транспортного потока на шинах входа и выхода интерфейса транспортного потока должно быть в соответствии со спецификацией [1] (4.4, 4.6, 4.7) с уточнениями в соответствии со стандартом [5] (приложение А, А.5.5.4).

Параметры функции памяти должны быть в соответствии со спецификацией [1] (4.6). Уточнения параметров функции памяти должны быть в соответствии со стандартом [5] (приложение А, А.5.5.5).

Параметры функции синхронизации должны быть в соответствии со спецификацией [1] (4.7). Поддержка узлами атрибута памяти обязательна. Поддержка узлом общей памяти является опциональной.

Параметры электрического интерфейса должны быть в соответствии со спецификацией [1] (4.9). Поддержка узлами перекрывающихся окон адресов ввода-вывода в соответствии со спецификацией [1] (4.9.3.2) является опциональной. Модули должны использовать независимые окна адресов ввода-вывода размером 4 байта.

Обнаружение платы выполняется в соответствии со спецификацией [1] (4.10).

Требования обнаружения модулями напряжения батареи не предъявляются. Поддержка этой функции узлами является опциональной.

Параметры включения питания и выключения питания должны быть в соответствии со стандартом [5] (приложение А, А.5.5.10).

Функция ввода-вывода должна быть в соответствии со спецификацией [1] (4.13) за исключением режимов чтения и записи только в режимах 8-битных разрядов.

Конфигурация функции должна быть в соответствии со спецификацией [1] (4.14). Модули и узел должны поддерживать конфигурации дополнительных регистров опционально.

Параметры конфигурация Карты должны быть в соответствии со спецификацией [1] (4.15, 4.15.1).

А.5.5 Параметры метаформата

Параметры метаформата должны быть в соответствии со стандартом [5] (приложение А, А.5.6).

Приложение Б
(справочное)

Дополнительные объекты

Б.1 Ресурс аутентификации

Этот ресурс включается опционально в узлы, поддерживающие аутентификацию для передачи команды аутентификации между съемным модулем и узлом так, чтобы использование любой процедуры аутентификации выполнялось под управлением системного оператора условного доступа только относительно сигналов, им контролируемых. Вещатель и (или) системный оператор условного доступа, не использующий протокол аутентификации, должен быть в состоянии передать сигналы, которые пройдут через общий интерфейс без использования процессов аутентификации.

Ресурс состоит из двух объектов: Запроса Аутентификации и Ответа Аутентификации, которые используются для реализации протокола аутентификации. Вопросы конкретного применения определяются между производителями систем СА и производителями узлов, которые включают этот ресурс.

Б.1.1 Объекты Запрос Аутентификации и Ответ Аутентификации

Параметры кодирования объектов Запрос Аутентификации и Ответ Аутентификации идентичны, за исключением значений тегов, и представлены в стандарте [5] (приложение В, В.1.1).

Б.1.2 Кодирование ресурса аутентификации

Кодирование ресурса аутентификации выполняется в соответствии с таблицами Б.1, Б.2.

Таблица Б.1 — Кодирование ресурса аутентификации

Ресурс	Класс	Тип	Версия	Идентификатор ресурса
Аутентификация	16	1	1	00100041

Таблица Б.2 — Кодирование ресурса аутентификации

apdu_tag	Значение тега (hex)	Ресурс	Направление обмена узел <- -> приложение
T_{auth_req}	9F 82 00	Аутентификация	<- - -
T_{auth_resp}	9F 82 01	Аутентификация	- - ->

Б.2 Ресурс изображения телетекста EBU

Ресурс определяет объект и протокол для связи с узлом в присутствии ресурса изображения телетекста EBU (European Broadcasting Teletext; EBT).

EBT является средством поставки данных, обеспечивающих отображение текста на телевизионном экране. Телетекст обычно переносится на незанятых строках на интервале поля гашения стандартных телевизионных сигналов и передается в режиме вещания в транспортном потоке мультиплекса MPEG-2. EBT определяет конкретную возможность отображения телетекста на экране телевизора, и определяет набор объектов для передачи данных для отображения при использовании этой возможности. В полном объеме параметры объектов представлены в стандарте [11], но основные особенности отображения телетекста приведены в Б.2.1.

Б.2.1 Характеристики изображения

Изображение образуется двумерной мозаичной сеткой символов от 40 до 160 символов на строку при 101 строке. Однако, при формате изображения 4:3 с 625 строками изображенние будет ограничено 40 символами на строку при 25 строках. Стандарт использует набор символов для отображения, но позволяет использовать динамически переопределяемый набор символов (Dynamically Redefinable Character Sets; DRCS). Стандарт использует для формирования изображения стандартный набор цветов, но разрешает загрузку карты различных цветовых таблиц.

Б.2.2 Принципы передачи объекта EBT

В соответствии с принятыми в настоящем стандарте принципами данные объекта EBT должны отображаться аналогично случаю эфирного телевидения в телевизионной строке или в транспортном потоке MPEG-2. Эта задача обеспечивается пакетами объекта EBT. Сеанс настроен на ресурс изображения EBT, который успешно выполняется, если другое приложение не использует дисплей. Дисплей выделяется в порядке поступления приложений. Пос-

ле приема первого пакета объекта EBT дисплей должен быть переключен из режима изображения в текстовый режим с выводом на экран того, что было загружено. Далее пакеты объекта EBT при необходимости могут быть отправлены по требованию. После окончания сеанса дисплей должен вернуться в режим изображения.

Б.2.3 Пакеты EBT

Кодирование пакетов EBT выполняется в соответствии с таблицей Б.3.

Т а б л и ц а Б.3 — Кодирование пакетов EBT

Синтаксис	Количество битов	Мнемоника

ebt_packets () {
ebt_packet_tag
length_field ()
for (i: = 0; i <no_of_packets; i ++) {
for (j=0; j <42; j ++) {
ebt_packet_byte
}
}
}
	24
	
	8
		uimsbf
		
		uimsbf

Данные пакетов объекта EBT составляют несколько строк длиной 42 байта каждая, соответствующих формату для приложения с 625 строками в спецификации EBT. В каждой строке первые два байта содержат журнал и поле адреса пакета. Каждый байт кодирован кодом Хэмминга с 4 битами информации и 4 битами четности. Из восьми информационных битов в двух байтах первые три бита образуют журнал количества пакетов (X), а последние 5 битов образуют адреса пакетов (Y). По соглашению значения X находятся в диапазоне 1-8, где 8 представляется как 000. Значения Y находятся в диапазоне 0-31. Кодирование других 40 байтов зависит от значений полей Y и X. В общем случае при значении Y = 0 пакеты содержат заголовки пакетов, кодированные кодом Хэмминга, с последующим изображением текста. При значениях Y от 1 до 25 пакеты содержат предназначенные для отображения на экране 40 байтов текста и символы управления. При значениях Y от 26 до 31 пакеты в общем случае переносят вспомогательные изображения и информацию управления, обеспечивающую расширенные возможности.

Б.2.4 Кодирование ресурса EBT

Кодирование ресурса EBT выполняется в соответствии с таблицами Б.4, Б.5.

Т а б л и ц а Б.4 — Кодирование ресурса EBT

Ресурс	Класс	Тип	Версия	Идентификатор ресурса
EBU Телетекст	128	1	1	00800041

Т а б л и ц а Б.5 — Кодирование ресурса EBT

apdu_tag	Значение тега (hex)	Ресурс	Направление обмена узел <- -> приложение
$T_{ebt_packets}$	9F 9000	EBU Телетекст	<- - -

Б.3 Класс ресурса считывателя смарт-карты

Ресурс считывателя смарт-карты (опциональный) может быть обеспечен узлом непосредственно или через другой модуль. Этот ресурс использует команды на уровне инструкции карты (заголовок, данные и байты состояния). Команды определяются в соответствии со стандартом [5] (приложение B, B.3.1).

Установлены требования к четырем объектам:

- смарт-карта Команда и смарт-карта Ответ выполняют функции управления и функции отклика на команду соответственно;

- смарт-карта Передача и смарт-карта Прием выполняют функции передачи и приема данных.

Ресурс позволяет устанавливать несколько сеансов так, чтобы различные приложения могли опрашивать ресурс, но в любой момент времени может находиться в подключенном состоянии только один сеанс.

Ресурс предназначен для поддержки считывателя смарт-карты узла или модуля, который используется для краткосрочных сеансов, таких как обработка банковской кредитной карточки, услуги телемагазина или транзакции повременной оплаты за предоставление. Не рекомендуется использование ресурса считывателя смарт-карты для продолжительных сеансов проведения операций по обеспечению безопасности с целью поддержки доступа к

скремблированной службе. Это обусловлено именно таким назначением ресурса считывателя смарт-карты, а не другими видами использования и также в связи с отсутствием (при других видах использования) гарантии ограниченной задержки ответа.

Б.3.1 Объекты смарт-карт

Б.3.1.1 Объект Команда (Cmd)

Синтаксис объекта Команда представлен в таблицах Б.6, Б.7.

Таблица Б.6 — Синтаксис объекта Команда

Синтаксис	Количество битов	Мнемоника

smart_card_cmd () {
smart_card_cmd_tag
length_field ()
smart_card_cmd_id
}
	24
	
	8
		uimsbf
		
		uimsbf

Таблица Б.7 — Синтаксис объекта Команда

smart_card_cmd_id	Значение идентификатора
Подключение	01
Разъединение	02
power_on_card	03
power_off_card	04
reset_card	05
read_status	06
read_answ_to_reset	07
Зарезервировано	Другие значения

Эта команда выдается приложением при установлении сеанса с ресурсом. Правила выполнения доступных команд: подключение, разъединение, power_on_card, power_off_card, reset_card, read_status, read_answ_to_reset должны быть в соответствии со стандартом [5] (приложение В, В.3.1.1).

Б.3.1.2 Объект Ответ (Reply)

Правила кодирования и семантика объекта Ответ (Reply) должны быть в соответствии со стандартом [5] (приложение В, В.3.1.2).

Б.3.1.3 Объект Передача (Send)

Правила кодирования и семантика объекта Передача (Send) должны быть в соответствии со стандартом [5] (приложение В, В.3.1.3).

Б.3.1.4 Объект Прием (Rcv)

Правила кодирования и семантика объекта Прием (Rcv) карты должны быть в соответствии со стандартом 5] (приложение В, В.3.1.4).

Б.3.1.5 Кодирование ресурса считывателя смарт-карты

Правила кодирования класса, типа и версии ресурса считывателя смарт-карты должны быть в соответствии со стандартом [5] (приложение В, В.3.2).

Б.4 Класс поддержки события DVB EPG

Этот ресурс доступен приложениям EPG для поддержки обмена информацией о правах доступа от модуля CA к электронному путеводителю по программам — или на основе узла или на основе модуля.

Модули CA, предлагающие информацию о праве доступа приложениям EPG, должны создать сеанс ко всем экземплярам этого ресурса, указанным менеджером ресурсов, и должны управлять протоколом в соответствии требованиями приложения EPG.

Протокол использует два объекта: событие Запроса (event_enquiry) и событие Ответа (event_reply). Протокол позволяет EPG запрашивать о состоянии права доступа текущего или будущего события по ID события в соответствии со служебной информацией (SI). Модуль CA может сообщить о том, что право на событие доступно, недоступно или может быть доступно после выполнениия диалога (например, после оплаты за просмотр) или, что статус события права доступа неизвестен. Кроме того протокол поддерживает процесс, позволяющий модулю CA начать

диалог о потенциальной доступности права доступа, например об оплате за просмотр, а затем возвратиться к диалогу EPG.

Этот ресурс не препятствует приложениям EPG и CA каких-либо поставщиков использовать частный механизм ресурса для более полной интеграции работы EPG и CA. Однако модули условного доступа, предоставляющие информацию о правах доступа к независимым EPG, должны использовать этот ресурс.

Б.4.1 Объекты

Б.4.1.1 Объект событие Запроса

Объект событие Запроса реализует запрос от EPG к модулю условного доступа. Синтаксис объекта должен быть в соответствии со стандартом [5] (приложение В, В.4.1.1).

Б.4.1.2 Объект событие Ответа

Объект событие Ответа реализует ответ от модуля условного доступа к EPG. Синтаксис объекта должен быть в соответствии со стандартом [5] (приложение В, В.4.1.2).

Б.4.1.3 Пример использования

Пример использования протокола обмена приведен в стандарте [5] (приложение В, В.4.1.3).

Б.4.2 Параметры кодирования ресурса события EPG

Параметры кодирования ресурса события EPG должны быть в соотвествии со стандартом [5] (приложение В, В.4.2).

Библиография

[1] PC Card Standard Volume 2 — Electrical Specification, February 1995 — Personal Computer Memory Card International Association, Sunnyvale, California

[2] PC Card Standard Volume 3 — Physical Specification, February 1995 — Personal Computer Memory Card International Association, Sunnyvale, California

[3] PC Card Standard Volume 4 — Metaformat Specification, February 1995 — Personal Computer Memory Card International Association, Sunnyvale, California

[4] ISO/IEC 13818-1 — Information technology — Generic coding of moving pictures and associated audio information: Systems — Part 1

[5] EN 50221 February 1997 — Common Interface Specification for Conditional Access and other Digital Video Broadcasting Decoder Applications

[6] ITU-T Recommendation V.21 (1988, 1993) — Data communication over the telephone network. 300 bits per second duplex modem standardized for use in the general switched telephone network

[7] ITU-T Recommendation V.22 (1988, 1993) — Data communication over the telephone network. 1200 bits per second duplex modem standardized for use in the general switched telephone network and on point-to-point 2-wire leased telephone-type circuits

[8] ITU-T Recommendation V.32bis (1991) — Data communication over the telephone network. A duplex modem operating at data signalling rates of up to 14 400 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits

[9] ETS 300 468 — Specification for Service Information (SI) in Digital Video Broadcasting (DVB) systems

[10] ETSI ETS 300 743 ed.1 (1997-09) — Digital Video Broadcasting (DVB); Subtitling systems

[11] EBU SPB 492 — Teletext Specification (625 line television systems), EBU, Geneva

УДК 621.397:006.354 ОКС 33.170 ОКП 65 7400

Ключевые слова: телевидение вещательное цифровое, общий интерфейс, система ограничения доступа, транспортный поток, интерфейс, DVB, дескремблирование, сеанс, объект, класс, ресурс

Редактор *Н.А. Аргунова*
Технический редактор *В.Н. Прусакова*
Корректор *Р.А. Ментова*
Компьютерная верстка *Ю.В. Демениной*

Сдано в набор 19.12.2014. Подписано в печать 29.01.2015. Формат 60 × 84 1/8. Гарнитура Ариал.
Усл. печ. л. 3,72. Уч.-изд. л. 2,98. Тираж 33 экз. Зак. 646.

Издано и отпечатано во ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.

gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru