ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 24730-1— 2017

Информационные технологии

СИСТЕМЫ ПОЗИЦИОНИРОВАНИЯ В РЕАЛЬНОМ ВРЕМЕНИ (RTLS)

Часть 1

Прикладной программный интерфейс (API)

Издание официальное

Москва
Стандартинформ
2017

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Научно-исследовательским и испытательным центром биометрической техники МГТУ им. Н.Э. Баумана (НИИЦ БТ МГТУ им. Н.Э. Баумана) и Федеральным государственным унитарным предприятием «Всероссийский научно-исследовательский институт стандартизации и сертификации в машиностроении» (ВНИИНМАШ) на основе собственного перевода на русский язык англоязычной версии стандарта, указанного в пункте 4, при консультативной поддержке Ассоциации автоматической идентификации «ЮНИСКАН/GS1 РУС»

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 355 «Технологии автоматической идентификации и сбора данных»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 9 июня 2017 г. № 534-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 24730-1:2014 «Информационные технологии. Системы позиционирования в реальном времени (RTLS). Часть 1. Прикладной программный интерфейс (API)» (ISO/IEC 24730-1:2014 «Information technology — Real-time locating systems (RTLS) — Part 1: Application programming interface (API)», IDT).

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

5 ВВЕДЕН ВПЕРВЫЕ

6 Некоторые элементы настоящего стандарта могут быть объектами патентных прав. Международная организация по стандартизации (ИСО) и Международная электротехническая комиссия (МЭК) не несут ответственности за установление подлинности каких-либо или всех таких патентных прав

*Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном 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.e-ecolog.ru

© Стандартинформ, 2017

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

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Термины и определения 2
4 Сокращения 2
5 Связь 2
5.1 Назначение 2
5.2 Общие положения 3
5.3 Конфигурация потока сообщений 4
5.4 Безопасность 4
5.5 Цель 4
5.6 Независимость от языка 4
5.7 Архитектура 4
5.8 SLMP сообщения 4
5.9 Формат сообщений простого определения места нахождения (Simple Location Message Format) через HTTP 12
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов национальным 16
Библиография 17

Введение

Комплекс стандартов ИСО/МЭК 24730 (далее — ИСО/МЭК 24730) имеет общий заголовок «Информационные технологии. Системы позиционирования в реальном времени (RTLS)» и включает в себя следующие части:

- Часть 1: Прикладной программный интерфейс (API);
- Часть 2: Протокол радиоинтерфейса для связи на частоте 2,4 ГГц с использованием расширения спектра методом прямой последовательности (DSSS);
- Часть 21: Протокол радиоинтерфейса для связи на частоте 2,4 ГГц с использованием расширения спектра методом прямой последовательности (DSSS): Передатчики системы RTLS, работающие с одним расширяющим кодом и использующие кодирование данных DBPSK и схему расширения BPSK;
- Часть 22: Протокол радиоинтерфейса для связи на частоте 2,4 ГГц с использованием расширения спектра методом прямой последовательности (DSSS): Передатчики системы RTLS, работающие с несколькими кодами расширения спектра и использующие кодирование данных QPSK и схему расширения QPSK со смещением функции Уолша (WOQPSK);
- Часть 5: Радиоинтерфейс расширения спектра методом линейной частотной модуляции (CSS) для связи на частоте 2,4 ГГц;
- Часть 61: Протокол радиоинтерфейса для сверхширокополосной связи (UWB) с низкой частотой повторения импульсов;
- Часть 62: Протокол радиоинтерфейса для сверхширокополосной связи (UWB) с высокой частотой повторения импульсов.

ИСО/МЭК 24730 определяет несколько протоколов радиоинтерфейсов и единый прикладной программный интерфейс (API) для систем позиционирования в реальном времени («Real-time locating systems», далее — «системы RTLS») для управления инфраструктурой системы и призван обеспечить конкурентоспособность и улучшить совместимость продуктов на растущем рынке систем RTLS.

Настоящий стандарт устанавливает технические требования к системам RTLS. Для полной совместимости с ИСО/МЭК 24730 система RTLS должна соответствовать настоящему стандарту и одному протоколу радиоинтерфейса, определенному в стандартах ИСО/МЭК 24730.

Системы RTLS — это беспроводные системы с возможностью определения места нахождения отдельных объектов или предметов в любой точке заданного пространства в момент времени, соответствующий или близкий к реальному времени. Положение рассчитывается при помощи измерений физических характеристик линий радиосвязи.

Существуют четыре классификации систем RTLS:

- определение места нахождения объекта при помощи спутника — требуется прямая видимость — точность до 10 м;
- определение места нахождения объекта в контролируемой зоне, например, складское помещение, территория университета, аэропорт — интересующая область оснащается необходимым оборудованием — точность до 3 м;
- определение места нахождения объекта в более ограниченной зоне — интересующая область оснащается необходимым оборудованием — точность десятки сантиметров;
- определение места нахождения объекта над земной поверхностью с использованием установленных на поверхности Земли приемников, например, вышек сотовой связи — точность 200 м.

Существует два дополнительных метода определения места нахождения отдельных объектов или предметов, которые основываются на технологии RFID:

- позиционирование объекта или предмета по факту прохождения в определенное время точки A и непрохождения точки B;
- позиционирование объекта или предмета с помощью радиомаяка, когда пользователь с переносным устройством может найти объект или предмет.

Само позиционирование включает в себя распознавание и позиционирование, обычно путем мультилатерации, следующих видов:

- время прохождения сигнала (Time of Flight) дальномерных систем;
- амплитуда/триангуляция по мощности входного сигнала;
- разница между моментами времени поступления сигнала (Time Difference of Arrival — TDoA);
- сотовая триангуляция;
- спутниковая мультилатерация;
- угол (направление) приема сигнала (Angle of Arrival — AoA).

Кроме того, к информации о месте нахождения может быть добавлена информация об ориентации объекта в пространстве.

Настоящий стандарт определяет прикладной программный интерфейс (API), необходимый для использования в системах RTLS.

API — это граничный слой, чье прикладное программное обеспечение использует средства языков программирования для вызова сервисов. Данные средства могут включать в себя процедуры или команды, общие объекты данных и разрешения идентификаторов. Для поддержки приложений в API требуется широкий диапазон сервисов. Для документирования спецификаций API различных типов сервисов могут быть целесообразными различные методы.

Информация, проходящая через граничный слой API, определяется синтаксисом и семантикой конкретного языка программирования, чтобы пользователь данного языка мог бы со своей стороны обращаться к сервисам, предоставляемым прикладной платформой. Это подразумевает спецификацию сопоставления функций, ставших доступными прикладной платформе через синтаксис и семантику языка программирования. Спецификация API документирует сервис и/или метод доступа к сервису, который доступен в интерфейсе между приложением и прикладной платформой.

API описывает сервис позиционирования в режиме реального времени и его методы доступа, чтобы дать возможность клиентским приложениям взаимодействовать с системами RTLS. Сервис позиционирования в режиме реального времени — это минимальное условие, предоставляемое системами RTLS, чтобы API соответствовало настоящему стандарту.

В настоящем стандарте «точка» используется для отделения дробной части числа после создания выходного файла API с расширением .csv, потому что «запятая» используется для разделения значений.

Международный стандарт ИСО/МЭК 24730-1 подготовлен подкомитетом № 31 «Автоматическая идентификация и технологии сбора данных» Совместного технического комитета № 1 ИСО/МЭК «Информационные технологии» (ISO/IEC JTC 1/SC 31).

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные технологии

СИСТЕМЫ ПОЗИЦИОНИРОВАНИЯ В РЕАЛЬНОМ ВРЕМЕНИ (RTLS)

Часть 1

Прикладной программный интерфейс (API)

Information technology. Real-time locating systems (RTLS).
Part 1. Application programming interface (API)

Дата введения — 2018—12—01

1 Область применения

Настоящий стандарт позволяет программным приложениям использовать инфраструктуру систем RTLS для обнаружения объектов при помощи передатчиков. Настоящий стандарт определяет граничный слой, через который программное приложение использует средства языков программирования для сбора информации, содержащейся в блинк-посылках меток системы RTLS, полученных от инфраструктуры системы RTLS.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты, которые необходимо учитывать при его использовании. В случае датированных ссылок необходимо пользоваться только указанной редакцией. В случае недатированных ссылок следует пользоваться последней редакцией ссылочных документов, включая любые поправки и изменения к ним.

ISO/IEC 15963 Information technology — Radio frequency identification for item management — Unique identification for RF tags (Информационные технологии. Радиочастотная идентификация для управления предметами. Уникальная идентификация радиочастотных меток)

ISO/IEC 19762 Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary (Информационные технологии. Технологии автоматической идентификации и сбора данных (АИСД). Гармонизированный словарь)

IEEE Guidelines for use of a 48-bit Extended Unique Identifier (EUI-48™) (Руководство по использованию 48-битного расширенного уникального идентификатора)

IEEE Guidelines for 64-bit Global Identifier (EUI-64™) Registration Authority (Руководство по использованию 64-битного расширенного уникального идентификатора. Органы регистрации)

Extensible Markup Language (XML) 1.0, (Third Edition), W3C Recommendation, World Wide Web Consortium (W3C), 04 February 2004[1)] (Расширяемый язык разметки (XML) 1.0, (Третье издание), рекомендации W3C, Консорциум Всемирной паутины (W3C), 4 февраля 2004 года)

SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), W3C Recommendation, World Wide Web Consortium (W3C), 27 April 2007[2)] (SOAP Версия 1.2. Часть 1: Программная платформа для передачи сообщений (Второе издание), рекомендации W3C, Консорциум Всемирной паутины (W3C), 27 апреля 2007 года).

1) gost.e-ecolog.ru

2) gost.e-ecolog.ru

Издание официальное

3 Термины и определения

В настоящем стандарте применены термины в соответствии с комплексом стандартов ИСО/МЭК 19762, а также следующие термины с соответствующими определениями:

3.1 **поле** (field): Элемент записи данных, в котором хранится информация, содержащая одну и более характеристики блинк-посылки метки.

3.2 **XML-тег** (XML tag): Дескриптор (именованная метка, маркер), определяющий содержимое в XML-документе.

3.3 **постоянное подключение** (persistent connection): Сетевое соединение между сервером и клиентской частью, которое остается открытым для передачи данных на прикладном уровне или для запроса на установление соединения даже после отправки на прикладном уровне сообщения об ошибке соединения.

3.4 **статус метки** (tag status): Обязательные поля, содержащие сообщение определения места нахождения и не включающие в себя поле «Source» и поле «Format».

4 Сокращения

В настоящем стандарте применены следующие сокращения:

API — прикладной программный интерфейс (Application Programming Interface);

ASCII — американский стандартный код для обмена информацией (American Standard Code for Information Interchange);

CR — управляющий знак набора ASCII, обозначающий операцию возврата печатающей головки (каретки) (ASCII Carriage Return);

EUI — расширенный уникальный идентификатор (Extended Unique Identifier);

JMS — стандарт промежуточного программного обеспечения для рассылки сообщений, позволяющий приложениям, выполненным на платформе Java EE, создавать, посылать, получать и читать сообщения (Java Messaging Service);

HTTP — протокол передачи гипертекста (HyperText Transfer Protocol);

HTTPS — протокол передачи гипертекста с защитой (HyperText Transfer Protocol Secure);

LF — управляющий знак набора ASCII, обозначающий место перевода строки, т. е. продолжение печати текста с новой строки или уже на следующей странице (ASCII Line Feed);

OUI — уникальный идентификатор организации (Organizationally Unique Identifier);

REST — метод взаимодействия компонентов распределенного приложения в глобальной сети Интернет (Representational State Transfer);

Система RTLS — система позиционирования в реальном времени (Real Time Locating System);

S-HTTP — защищенный протокол передачи гипертекста (Secure HyperText Transfer Protocol);

SLMF — формат сообщений простого определения места нахождения (Simple Location Message Format);

SLMP — протокол сообщений простого определения места нахождения (Simple Location Message Protocol);

SOAP — простой протокол доступа к объектам (Simple Object Access Protocol);

SSL — протокол безопасных соединений (Secure Sockets Layer);

TDoA — разница между моментами времени поступления сигнала (Time Difference Of Arrival);

TCP/IP — набор сетевых протоколов передачи данных, используемых в сетях, включая глобальную сеть Интернет (Transmission Control Protocol/Internet Protocol);

XML — расширяемый язык разметки (eXtensible Markup Language);

XSD — язык описания структуры XML-документа (XML Schema Definition).

5 Связь

5.1 Назначение

Целью программного обеспечения API системы RTLS является обеспечение простого минимального интерфейса, который облегчает внедрение и ввод в эксплуатацию как системы RTLS, так и прикладной программы. Целью также является обеспечение быстрой передачи сообщений любому клиентскому приложению, подключенному к системе RTLS. Кроме того, поток сообщений должен быть удобочитаемым и легко интерпретируемым.

API должен поддерживаться устройством с минимальной функциональностью, являющимся «устройством сбора и передачи данных» («collection and forwarding device») без обязательного постоянного хранения данных на сервере и без базы данных.

Это устройство может обеспечивать сглаживание характеристик интенсивности фильтрования и определения места нахождения, но данные функции не требуются в данном API, потому что API может функционировать до или после данного типа предварительной обработки.

5.2 Общие положения

Структурная схема системы RTLS должна иметь соединение «Text over Socket», чтобы клиент по протоколу TCP/IP мог подключаться и в режиме реального времени получать поток сообщений от системы RTLS.

Сообщения отделяются друг от друга знаками <CR><LF> для облегчения отображения консоли. Формат поля отделяется запятыми.

Протокол «Text over Socket» — это минимальное обязательное требование. Если реализованы дополнительные транспортные протоколы, такие как HTTP и JMS, тогда должны быть использованы текстовый формат или XML[1]. Если реализованы REST или SOAP[2], тогда необходимо использовать формат XML. Текстовый формат включает в себя поля, разделенные запятыми, тогда как формат XML включает в себя сокращенные XML-теги. Оба формата описаны в настоящем стандарте.

Использование REST, SOAP и сервисов с большей функциональностью является необязательным, потому что они ограничивают скорость передачи.

Задачей настоящего стандарта является обеспечение передачи 3000 сообщений в секунду и больше. Несмотря на то, что во многих приложениях объем в 3000 сообщений в секунду может и не понадобиться, минимальный API систем RTLS должен его поддерживать, потому что существующее на текущий момент применение меток с 3,000 блинк-посылками на частоте 1 Гц может с легкостью поддерживать такое количество сообщений. При интенсивности фильтрования и/или нацеленности информационных систем и программного обеспечения на управление большими объемами данных, REST, SOAP и другие методы передачи сообщений могут обеспечить дополнительную функциональность. Применение текстового формата с разделением запятыми обосновано при поддержке высокой скорости передачи данных, потому что данный формат не является избыточным и позволяет методам передачи сообщений функционировать на средних и высоких скоростях.

Представленный API или протокол называется «формат сообщений простого определения места нахождения» (SLMP, Simple Location Message Protocol). Обязательный формат, разделенный запятыми, называется «протокол сообщений простого определения места нахождения» (SLMF, Simple Location Message Format). Для отдельных транспортных средств SLMF-сокеты («SLMF-Sockets») — это обязательный совместимый с TCP/IP интерфейс/API протокола SLMP, как, например, SLMF-HTTP — это дополнительный интерфейс/API, поддерживающий HTTP, как элементарный сервис REST систем RTLS. Реализация SLMP может дополнительно включать в себя формат XML.

Клиентское приложение подключается к системе RTLS при помощи TCP/IP-соединения. Система RTLS отвечает потоком сообщений, который прерывается только при закрытии соединения с клиентом.

Система RTLS должна передавать сообщения, подтверждающие ее активность, если линия молчит в течение длительных промежутков времени. Настоящий стандарт не устанавливает обязательного интервала для сообщений, подтверждающих активность, а оставляет этот вопрос на усмотрение поставщика системы RTLS (см. 5.7.3). В случае потери безопасного соединения с системой RTLS, клиентское приложение должно периодически повторять попытки подключения.

Система RTLS предоставляет API через устройство с минимальными техническими характеристиками, которое собирает сообщения от считывателей, определяет место нахождения и пересылает сообщения. Это устройство не требует постоянного хранения данных на сервере, а также хранения архивных данных или статуса последней метки во время активной сессии. Тем не менее, данный API не предоставляет статус метки, а предоставляет только события, связанные с меткой. Статус метки оставлен для приложения, имеющего в контексте работы знание массива данных, либо для API более высокого уровня, выходящего за рамки настоящего стандарта.

---

[1] Extensible Markup Language (XML) 1.0, (Third Edition), W3C Recommendation, World Wide Web Consortium (W3C), 04 February 2004. (gost.e-ecolog.ru).

[2] SOAP Version 1.2 Part0: Primer, W3C Recommendation, World Wide Web Consortium (W3C), 24 June 2003, (gost.e-ecolog.ru).

API, представленный в настоящем стандарте, поддерживает несколько одновременных клиентских подключений.

5.3 Конфигурация потока сообщений

В настоящем стандарте не устанавливаются конкретные методы фильтрования исходящего потока сообщений системы RTLS; тем не менее, поставщик системы RTLS при необходимости может реализовать фильтрование в устройстве сбора и передачи данных, соответствующее настоящему стандарту. Например, поставщик мог бы реализовать метод подтверждения на стороне клиента, удовлетворяющий характеристикам фильтра системы RTLS. Система RTLS могла бы затем использовать характеристики фильтра для ограничения исходящего к клиентам потока сообщений. С другой стороны, поставщик мог бы реализовать конфигурацию характеристик фильтра на стороне сервера для обращения ко всем клиентским соединениям.

5.4 Безопасность

Протоколы системы безопасности аппаратуры обмена сообщениями; системы RTLS не рассматриваются в настоящем стандарте, потому что вопросы безопасности можно переадресовать к существующим стандартам по безопасности и технологиям на коммуникационных уровнях, основанных на предпочтениях и стратегии индивидуальных заказчиков. Например, при соединении по протоколу TCP/IP используется протокол системы безопасности SSH, который легко реализовать (совместимо с настоящим стандартом). Аналогично, протоколы системы безопасности, такие как HTTPS и S-HTTP, также могут быть реализованы, как и вышеуказанный протокол.

5.5 Цель

API, представленный в настоящем стандарте, определяет стандартный механизм доступа клиентского приложения к блинк-посылкам меток с более точным положением от системы RTLS.

5.6 Независимость от языка

API, представленный в настоящем стандарте, определяет независимый от языка программирования интерфейс по отношению к сервису RTLS. Это достигается использованием стандартизованного протокола производственной (вычислительной) сети «Text over Socket» (TCP/IP) для соединения с сервисом RTLS.

5.7 Архитектура

На рисунке 1 описан API обмена сообщениями между клиентским приложением и системой RTLS. В настоящем стандарте API допускает несколько клиентских подключений, таким образом API поддерживает состояние соединения TCP/IP для каждого клиента.

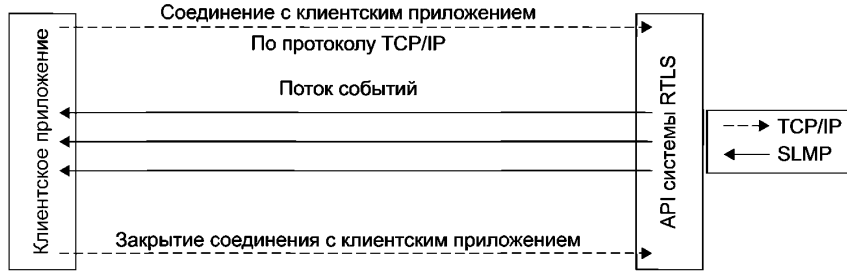


Рисунок 1 — Архитектура протокола SLMP

5.8 SLMP сообщения

В данном подразделе описаны сообщения, которые используются в настоящем стандарте. Каждый тип сообщения включает в себя набор полей, которые поставщик системы RTLS реализует в соот-

ветствии с определениями, приведенными в настоящем стандарте. Все данные полей и наименования XML-тегов, которые подробно определены в настоящем стандарте, чувствительны к регистру.

5.8.1 Типы данных

Типы данных, описанные в данном пункте, относятся к полям, связанными с сообщениями, определенными в настоящем стандарте. Для сообщений определения места нахождения (Locate message) поставщик системы RTLS может дополнительно включать в состав поля, не описанные в настоящем стандарте. Для таких полей поставщик может выбирать тип данных по своему усмотрению.

DateTime

Данный тип данных представляет собой формат даты и времени (date time format) аналогично международному стандарту ИСО 8601: YYYY-MM-DDThh:mm:ss-hh:mm.

Год в виде **YYYY**-MM-DD
Месяц в виде YYYY-**MM**-DD
День в виде YYYY-MM-**DD**

«T» показывает место начала отображения времени «**T**ime will follow».

Часы в виде **hh**:mm:ss
Минуты в виде hh:**mm**:ss
Секунды в виде hh:mm:**ss**

Плюс или минус смещение от универсального глобального времени (по Гринвичу) в часах и минутах **(-hh:mm or +hh:mm)**.

Пример — 2010-11-24T09:07:04-08:00 // для стандартного тихоокеанского времени.

Необходимо отметить, что дробная часть с точностью до одной десятой миллисекунды (.0001 с) может быть добавлена к элементу времени низшего порядка. Например, чтобы показать 14 ч, 30 мин и 12.359 с, необходимо представить это время как 14:30:12.359.

Double

Данный тип данных представляет собой числовой формат с плавающей точкой, включающий в себя дополнительно закодированный десятичный разделитель, и может отображаться с экспонентой и мантиссой или без них. Примеры включают в себя: 2345.334, —98.7, 1.0, 4, 0.0, 0.5, 9.87+E8.

Диапазон значений поля типа «Double»: от 1.7E—308 до 1.7E+308, максимальная длина строки — 256 символов.

HexBinary

Данный тип данных представляет собой структурированные или неструктурированные данные, которые можно представить в шестнадцатиричном формате, где каждый байт является бинарным октетом. Полубайт старшего разряда представляется как (крайний слева) полубайт в октете, и каждая шестнадцатеричная строка содержит четное число полубайт.

Максимальная длина поля для типа поля «HexBinary» — 256 байт.

Integer

Данный тип данных представляет собой числа, которые могут быть записаны без дробной или десятичной части и входят в набор {..., −2, −1, 0, 1, 2, ...}.

Диапазон значений поля типа «Integer»: от −2,147,483,648 до 2,147,483,647.

String

Данный тип данных представляет собой набор ASCII-символов, ограниченный следующими символами:

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, space, !, (,), [,], *, #, $, %, &, +, -, _, ., /, ?, =

Максимальная длина поля для поля типа «String» — 256 символов.

5.8.2 Заголовок сообщения (Header Message)

При установке соединения с клиентским приложением система RTLS отправляет отдельный заголовок сообщения.

Последовательность полей:

<Appliance_ID>, SLMF, <SLMF_version>, <SLMF_vendor_version>, <Greeting> <CR> <LF>

Поля:

Appliance_ID (обязательное поле)

XML-тег: <aid>

Тип данных: String

Длина: от 1 до 10 символов

Представляет собой систему RTLS или устройство, отправляющее сообщения определения места нахождения.

SLMF_version (обязательное поле)

XML-тег: <ver>

Тип данных: String

Длина: от 1 до 10 символов

Версия реализации формата SLMF, установленная настоящим стандартом. Версия формата SLMF для данной версии настоящего стандарта — 1.0.

SLMF_vendor_version (обязательное поле)

XML-тег: <vver>

Тип данных: String

Длина: от 1 до 10 символов

Версия поставщика, реализующего формат SLMF. Данное поле используется, когда поставщик предоставляет более одного формата не по умолчанию, связанных с сообщениями определения места нахождения. Если это поле не используется, то оно остается незаполненным.

Greeting (обязательное поле)

XML-тег: <greet>

Тип данных: String

Длина: от 1 до 256 символов

Сообщение определяется поставщиком системы RTLS, который представляет приветствие.

Пример — MyAppl, SLMF, 1.0,1.3, Welcome to the RTLS Text Stream interface. <CR> <LF>

5.8.3 Сообщения с определениями полей (Field Definition Message)

Сообщения с определениями полей отправляются сразу же после отправки системой RTLS заголовка сообщения и представляют собой поля, используемые в сообщениях определения места нахождения. Сообщения определения места нахождения включают в себя обязательные и необязательные поля с определениями, которые должны быть включены в сообщение в случае их использования поставщиком системы RTLS.

Последовательность полей:

FieldDefinition, <Name>, <Type> <CR> <LF>

Поля:

Name (необходимое поле)

XML-тег: <nam>

Тип данных: String

Длина: от 1 до 256 символов

Наименование поля данных, которое связано с одним или несколькими сообщениями определения места нахождения. Если поле явно определено в 5.8, тогда значение поля «Name» в сообщении с определениями полей должно совпадать с наименованием поля, определенным в 5.8. При использовании XML наименование поля должно совпадать с XML-тегом, определенным в 5.8.

Type (необходимое поле)

XML-тег: <typ>

Тип данных: String

Длина: от 1 до 256 символов

Тип данных, связанный с полем. Если поле определено в 5.8, тогда значение поля «Type» в сообщении с определениями полей должно совпадать с типом данных соответствующего поля в 5.8.

Пример — FieldDefinition, Source, String <CR> <LF>.

5.8.4 Сообщения с определениями сообщения определения места нахождения (Locate Message Definition Message)

Сообщения с определениями сообщения определения места нахождения отправляются сразу же после отправки сообщений с определениями полей. Одно сообщение с определениями сообще-

ния определения места нахождения должно быть определено для каждой уникальной пары <Source>, <Format>.

Последовательность полей:

LocateMessageDefinition, <Source>, <Format>, <Field1>, <Field2>, <Field3>, ...<CR> <LF>

Поля:

Source (обязательное поле)

XML-тег: <src>

Тип данных: String

Длина: от 1 до 64 символов

Определение поля: FieldDefinition, Source, String <CR> <LF>

Поле «Source», как правило, представляет собой детальную методику определения места нахождения или семейство программных продуктов. Например, если система RTLS создает сообщения определения места нахождения, которые созданы на основе семейства программных продуктов A и семейства программных продуктов B, то поставщик системы RTLS может устанавливать значение поля «Source» для каждого из двух семейств программных продуктов. Значения поля «Source» определяются поставщиками системы RTLS; тем не менее, поставщик системы RTLS может предоставить метод, позволяющий потребителям дополнительно наносить на карту альтернативные значения поля «Source» по их собственному выбору. Настоящий стандарт не устанавливает определенный метод нанесения на карту значения поля «Source»; это оставлено на усмотрение поставщиков системы RTLS.

Format (обязательное поле)

XML-тег: <fmt>

Тип данных: String

Длина: от 1 до 64 символов

Определение поля: FieldDefinition, Format, String<CR><LF>

Данный формат представляет собой набор полей, содержащихся в сообщении определения места нахождения. Если сообщение определения места нахождения содержит необязательные поля, то комбинация полей «Format» и «Source» должна использоваться клиентскими приложениями для определения того, как производить обработку сообщения; в противном случае, в поле формат должно быть значение «DFT», показывающее, что в сообщении определения места нахождения содержатся только обязательные поля. Значения поля формат, отличные от «DFT», определяются поставщиками системы RTLS.

Field Names (обязательное поле)

См. 5.8.6 для наименования полей, относящихся к сообщениям определения места нахождения. Поставщик системы RTLS может определять собственные наименования полей, для дополнительных полей, не указанных явно в 5.8.6.

LocateMessageDefinition, <Source>, <Format>, Tag_ID_Format, Tag_ID, X, Y, Z, Battery, Timestamp, Ext1, Ext2, Ext3,...<CR> <LF>

Примеры

1 LocateMessageDefinition, MySourceA, DFT, Tag_ID_Format, Tag_ID, X, Y, Z, Battery, Timestamp <CR> <LF>.

2 LocateMessageDefinition, MySourceB, DFT, Tag_ID_Format, Tag_ID, X, Y, Z, Battery, Timestamp <CR> <LF>.

3 LocateMessageDefinition, MySourceB, S, Tag_ID_Format, Tag_ID, X, Y, Z, Battery, Timestamp, Algorithm <CR> <LF>.

4 LocateMessageDefinition, MySourceB, T, Tag_ID_Format, Tag_ID, X, Y, Z, Battery, Timestamp, Data <CR> <LF>.

5.8.5 Сообщение, подтверждающее активность (Keep-Alive Message)

Система RTLS может дополнительно включать в себя структуру, инициирующую отправку сообщения, подтверждающего активность, если время, прошедшее с момента отправки предыдущего сообщения определения места нахождения, превышает указанный срок. Сообщение, подтверждающее активность, также отправляется каждый раз при установлении соединения клиентского API с системой RTLS.

Последовательность полей:

KeepAlive, <Period> <CR> <LF>

Поля:

Period (необходимое поле)

XML-тег: <per>

Тип данных: String

Длина: от 1 до 3600 символов

Временной интервал в секундах, который инициирует отправку сообщения, подтверждающего активность, когда линия молчит. Значение данного поля устанавливается в системе RTLS.

Пример — KeepAlive, 60 <CR> <LF>

5.8.6 Сообщение определения места нахождения (Locate Message)

Событие, связанное с меткой системы RTLS, обычно выражается как сообщение определения места нахождения, содержащее обязательные поля, плюс любое число дополнительных полей, которые относятся к «расширениям». Поставщики систем RTLS могут определять их собственные дополнительные поля и типы данных, если их нет в данном пункте. Если система RTLS не может определить значение <x>, <y> или <z>, то неизвестные поля остаются пустыми.

Максимальная длина сообщения определения места нахождения — 2^14 (или 16384) символов.

Последовательность полей:

<Source>, <Format>, <Tag_ID_Format>, <Tag_ID>, <X>, <Y>, <Z>, <Battery>, <Timestamp>, <Ext1>, <Ext2>, <Ext3> ...<CR> <LF>

Поля:

Source (обязательное поле)

См. 5.8.4 для определения поля. Для данного сообщения пара Source/Format должна совпадать с парой Source/Format сообщения с определениями сообщения определения места нахождения.

Format (обязательное поле)

См. 5.8.4 для определения поля. Для данного сообщения пара Source/Format должна совпадать с парой Source/Format сообщения с определениями сообщения определения места нахождения.

Tag_ID_Format (обязательное поле)

XML-тег: <idfmt>

Тип данных: HexBinary

Длина: 1 байт

Определение поля: FieldDefinition, Tag_ID_Format, HexBinary <CR> <LF>

Показывает формат, использующийся для серийного номера сегмента идентификатора радиочастотной метки (Tag ID). Поле «Tag_ID_Format», определенное в настоящем стандарте, должно принимать одно из значений, приведенных в таблице 1.

Таблица 1 — Форматы Tag ID

Tag_ID_Format	Формат
0x01	ISO/IEC 15963
0x02	IEEE EUI-48
0x03	IEEE EUI-64

Tag_ID (обязательное поле)

XML-тег: <tid>

Тип данных: HexBinary

Длина: переменная

Определение поля: FieldDefinition, Tag_ID, HexBinary <CR> <LF>

Уникальный идентификатор метки. Поле «Tag ID» имеет переменную длину в зависимости от поля «Tag_ID_Format». Ниже приведены примеры значений поля «Tag_ID» на основе значений поля «Tag_ID_Format».

ИСО/МЭК 15963

Представляет собой 48-битовый формат ИСО/МЭК 15963, включающий в себя код категории, идентификатор изготовителя и серийный номер. Код категории и идентификатор изготовителя определены в стандарте ИСО/МЭК 15963 и представляют собой первые 16 бит поля «Tag ID». Серийный номер занимает 32 бита. Предположим, что поле «Tag_ID_Format» имеет значение 0x01. Если код категории метки 0x00, идентификатор изготовителя 0x01 и серийный номер 0x00BC614E, тогда поле «Tag_ID» будет содержать значение 0x000100BC614E и отображаться в сообщении как *000100BC614E*.

IEEE EUI

Представляет собой 48-битный или 64-битный формат IEEE EUI, который включает в себя уникальный идентификатор организации и серийный номер. Уникальный идентификатор организации име-

ет размер 24 бита и присваивается Институтом инженеров по электротехнике и электронике (IEEE). Серийный номер имеет размер 24 бита для формата IEEE EUI-48 и 40 бит для формата IEEE EUI-64. Предположим, что поле «Tag_ID_Format» имеет значение 0x02. Если уникальный идентификатор организации метки 0x00003A и серийный номер 0x01E64A, тогда поле «Tag_ID» должно быть представлено как 0x00003A01E64A и отображаться в сообщении как *00003A01E64A*.

X, Y, Z (обязательное поле)

XML-тег: <x>, <y>, <z>

Тип данных: Double

Определение поля: FieldDefinition, X, Double <CR> <LF>

Определение поля: FieldDefinition, Y, Double <CR> <LF>

Определение поля: FieldDefinition, Z, Double <CR> <LF>

Место нахождения метки определяется относительно известной точки. Единицы определения места нахождения выражаются как декартовы координаты в метрах или дробных частях метров и могут иметь как отрицательные, так положительные значения.

В случае, когда метка обнаружена, а ее декартовы координаты не могут быть определены, тогда значениями для <X>, <Y> или <Z> из сообщения определения места нахождения можно пренебречь; тем не менее, сообщение определения места нахождения должны включать в себя разделители для координат в виде запятых.

Battery (обязательное поле)

XML-тег: <bat>

Тип данных: Integer

Длина: 0, 1, 2 или 3

Определение поля: FieldDefinition, Battery, Integer <CR> <LF>

Допустимые целочисленные значения приведены в таблице 2.

Таблица 2 — Состояние батареи, значение поля «Battery»

Значение	Состояние
0	Хорошее
1	Необходима замена
2	Зарезервировано для использования настоящим стандартом
3	Состояние батареи неизвестно

Timestamp (обязательное поле)

XML-тег: <t>

Тип данных: DateTime

Определение поля: FieldDefinition, Timestamp, DateTime <CR> <LF>

Поле «Timestamp» представляет собой дату и время определения места нахождения метки.

Classification (необязательное поле)

XML-тег: <cls>

Тип данных: String

Длина: от 1 до 64 символов

Определение поля: FieldDefinition, Classification, String <CR> <LF>

Поле «Classification» — это представление совокупности меток, основанное на едином наборе атрибутов, связанных с функцией меток или объектов, к которым они подключены. Это поле полезно, когда приложению необходимо трактовать классы объектов по-другому, или когда устройству сбора данных необходимо применять специальную логику для метки, основанную на ее типе перед публикацией через API. Устройство сбора данных может включать в себя метод классификации меток, решение о реализации которых принимается поставщиком системы RTLS.

Например, поле «Classification» может иметь значение «Asset», а другое — «Visitor». Дополнительные примеры значений поля «Classification»: «Trailer», «Tractor», «Container», «Pallet» и «Forklift».

Поле «Classification» является дополнительным, потому что в некоторых приложениях классификация меток входит в состав бизнес-логики. Примером этого является приложение морского терминала, где транспортное средство, управляющее оборудованием, имеет несколько меток, и их классификация связана с особой ролью, которую метки выполняют на конкретных типах транспортных средств. Следо-

вательно, приложение взаимодействует с типами транспортных средств, и метки являются всего лишь частью оборудования транспортных средств.

Zone (необязательное поле)

XML-тег: <zon>

Тип данных: String

Длина: от 1 до 256 символов

Определение поля: FieldDefinition, Zone, String <CR> <LF>

Наименования поля «Zone», такие как место для стоянки, здание или комната. Поле «Zone» также может быть выражено в форме пути, например, Здание/Строение/Комната.

Обычно поле «Zone» рассчитывается по координатам x, y, z путем проверки, в какую именно зону входит точка с координатами x, y, z. Перечень значений поля «Zone» — это список именованных полигонов.

Поле «Zone» является необязательным, потому что в некоторых приложениях определения поля «Zone» привязаны к бизнес-логике, определение места нахождения зоны и ее поиск вынесены на прикладной уровень. Примером этого является операционная система морского терминала, где места нахождения контейнеров в штабелях определяется поперечными рядами, рядами по ширине, ярусами по высоте (row-bay-tier), и таким образом зоны легко детализируются.

Exciter_ID (необязательное поле)

XML-тег: <exc>

Тип данных: String

Длина: от 1 до 64 символов

Определение поля: FieldDefinition, Exciter_ID, String <CR> <LF>

Возбудитель — это устройство, которое заставляет метку посылать сигналы при приближении. Сообщение с блинк-посылкой метки может включать поле «Exciter_ID» в тело сообщения. Значением поля «Exciter_ID» обычно является целое число, но оно также может принимать значения IP-адреса или DNS-имени.

Antenna_ID (необязательное поле)

XML-тег: <ant>

Тип данных: Integer

Длина: от 0 до 255

Определение поля: FieldDefinition,Antenna_ID,Integer<CR><LF>

Поле «Antenna_ID» обычно применяется с пассивными радиочастотными метками и представляет собой физическую контрольную точку, находящуюся в зоне покрытия. При использовании радиочастотных антенн вывод о месте нахождения метки можно сделать благодаря известному положению стационарной антенны.

Data (необязательное поле)

XML-тег: <dat>

Тип данных: HexBinary

Длина: от 1 до 123 байт

Определение поля: FieldDefinition, Data, HexBinary <CR> <LF>

Данное поле содержит неструктурированную информацию, передаваемую меткой, которая может быть декодирована клиентским API. Например, сенсоры, подключенные к метке, могут передавать температуру, уровень топлива и состояние двигателя по беспроводной связи в систему RTLS. Система RTLS декодирует неструктурированную информацию перед ее отправкой клиенту, но допускается для рациональности производить декодирование на уровне приложения.

Algorithm (необязательное поле)

XML-тег: <alg>

Тип данных: String

Длина: от 1 до 64 символов

Определение поля: FieldDefinition, Algorithm, String <CR> <LF>

Формулы, определяемые поставщиком, обычно успешно определяют место нахождения по координатам X, Y, Z. Например, значение «P» может показывать «Presence» («Наличие»), когда место нахождения метки получено от единственного локационного датчика.

A, B, C (необязательное поле)

XML-тег: <a>, <b>, <c>

Тип данных: Double

Длина: A и C — это радианы в интервале от 0 до 2π, а B — это радиан в интервале от 0 до π

Определение поля: FieldDefinition, A, Double <CR> <LF>
Определение поля: FieldDefinition, B, Double <CR> <LF>
Определение поля: FieldDefinition, C, Double <CR> <LF>

Пространственная трехмерная ориентация метки. Ориентация представлена углами Эйлера, внутренним вращением относительно системы отсчета в трехмерном евклидовом пространстве. Три эйлерова угла A, B и C однозначно определяют композицию трех поворотов. Данная спецификация требует использования конвенции Тейта-Брайена, разложения поворота на три последовательных внутренних поворота вокруг осей X (первый поворот, вокруг неподвижной оси абсцисс), Y' (второй поворот, вокруг поменявшей направление ориентации Y', перемещаемой по оси ординат) и Z'' (третий поворот, вокруг поменявшей направление ориентации Z'', вторично перемещаемой по оси аппликат). Другими словами, конвенция X-Y'-Z'' используется, когда углы A, B и C описывают точные углы Эйлера в целевой системе координат относительно системы отсчета.

В том, что касается API, A и C равны модулю 2π радиан, а B — это радиан в диапазоне $[0, \pi]$.

Примеры не расширенных сообщений определения места нахождения с определением положения:

MySourceA,DFT,01,000100BC614E,100,150,8,0,2010-11-24T09:07:04-08:00<CR><LF>
MySourceB,DFT,02,000040E60A11,53,-40.3,1.5,0,2010-11-24T09:07:04-08:00<CR><LF>

Примеры не расширенных сообщений определения места нахождения без определения положения:

MySourceB,DFT,02,000040E60A11,,,,0,2010-11-24T09:07:04-08:00<CR><LF>

Примеры расширенных сообщений определения места нахождения:

MySourceA,S,01,000100BC614E,24,903,8,0,2010-11-24T09:07:04-08:00,3D<CR><LF>
MySourceA,T,01,000100BC614E,100,150,8,0,2010-11-24T09:07:04-08:00,2D,0FE321AB<CR><LF>

Поле «Format» может быть использовано, чтобы показать подробное расширение поля или набор расширений полей. В примерах, приведенных выше, сообщение с форматом = 'S' представляет собой расширенное сообщение с единственным дополнительным полем, тогда как сообщение с форматом = 'T' представляет собой расширенное сообщение с двумя дополнительными полями. Оба значения формата определяются поставщиком системы RTLS, т. к. они описывают сообщения, включающие в себя расширенные поля.

5.8.7 Пример последовательности сообщений

Последовательность сообщений, приведенная ниже, представляет собой пример сообщений, отправленных системой RTLS, после установления связи с клиентским приложением.

MyAppl,SLMF,1.0,Welcome to the RTLS Text Stream interface.<CR><LF>
FieldDefinition,Source,String<CR><LF>
FieldDefinition,Format,String<CR><LF>
FieldDefinition,Tag_ID_Format,HexBinary<CR><LF>
FieldDefinition,Tag_ID,HexBinary<CR><LF>
FieldDefinition,X,Double<CR><LF>
FieldDefinition,Y,Double<CR><LF>
FieldDefinition,Z,Double<CR><LF>
FieldDefinition,Battery,HexBinary<CR><LF>
FieldDefinition,Timestamp,DateTime<CR><LF>
FieldDefinition,Algorithm,String<CR><LF>
FieldDefinition,Data,HexBinary<CR><LF>
FieldDefinition,A,Double<CR><LF>
FieldDefinition,B,Double<CR><LF>
FieldDefinition,C,Double<CR><LF>
LocateMessageDefinition, MySourceA,DFT, Tag_ID_Format,Tag_ID,X,Y,Z,Battery, Timestamp <CR><LF>
LocateMessageDefinition,MySourceA,S,Tag_ID_Format,Tag_ID,X,Y,Z,Battery, Time stamp, Algorithm<CR><LF>
LocateMessageDefinition,MySourceA,T,Tag_ID_Format,Tag_ID,X,Y,Z,Battery, Timestamp, Algorithm, Data<CR><LF>
LocateMessageDefinition,MySourceC,SP,Tag_ID_Format,Tag_ID,X,Y,Z, Battery, Timestamp, Algorithm,Data,A,B,C<CR><LF>

KeepAlive,30<CR><LF>
MySourceA,DFT,01,000100BC614E,100,150,8,1,2010-11-24T09:07:04-08:00<CR><LF>
MySourceA,S,01,000100BC614E,100,150,8,0,2010-11-24T09:07:04-08:00,2D<CR><LF>
MySourceA,S,01,000100BC614E,100,150,8,0,2010-11-24T09:07:04-08:00,L<CR><LF>
MySourceA,T,01,000100BC614E,100,150,8,0,2010-11-24T09:07:04-08:00,2D,0A46E137 <CR><LF>
MySourceA,S,01,000100BC614E,100,150,8,0,2010-11-24T09:07:04-08:00,P<CR><LF>
MySourceC,SP,01,000100BC614E,100,150,8,0,2010-11-24T09:07:04-08:00,3D,B7F349, 3.14,
0.785,0.0<CR><LF>

5.9 Формат сообщений простого определения места нахождения (Simple Location Message Format) через HTTP

5.9.1 Назначение

Целью SLMF-HTTP является обеспечение простого и быстрого потока сообщений через API, который легко интегрировать с промышленными системами и с облачными приложениями через HTTP. Смысл данного формата — поддержание определения полей SLMF и максимально высокая скорость передачи данных.

5.9.2 Протокол

Система RTLS — это клиент. Сервер — это предприятие или облачный прикладной веб-сервис. Система RTLS отправляет HTTP-вызов (через POST), содержащий сообщения, накопленные за определенный период времени. Период времени конфигурируется в системе RTLS, в зависимости от скорости передачи данных и вычислительной среды. Кроме того, система RTLS может контролировать отправку HTTP-вызова на основе числа сообщений, но всегда с ограничением по времени, таким образом, что одиночному сообщению не нужно постоянно ждать отправки.

HTTP-метод позволяет подключаться через исходящий порт 80, что допускается промышленными межсетевыми экранами, а также используется для просмотра веб-страниц. Этот метод также позволяет обеспечить безопасность через HTTPS.

5.9.3 Конфигурация

В системе RTLS коннектор SLMF-HTTP может быть остановлен и запущен заново (например, через консоль администратора). Этому коннектору необходимы следующие параметры:

- HTTP_Service_URL;
- HTTP_Service_Port (по умолчанию, HTTP использует порт 80).

Дополнительные параметры конфигурации могут включать в себя:

- параметры накопления сообщений (Message_Accumulation):
- максимальное число сообщений (max number of messages);
- период накопления (accumulation period);
- параметры восстановления при возникновении ошибок (Error Recovery):
- число повторных попыток после возникновения ошибки (number of retries after error);
- число сохраненных сообщений до потери информации (number of messages to keep before data loss).

5.9.4 Определения XML-схемы (XML Schema Definitions)

Сообщение определения места нахождения:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="gost.e-ecolog.ru                    >
    <xs:element name="src">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:minLength value="1"/>
                <xs:maxLength value="64"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="fmt">
        <xs:simpleType>
```

```
            <xs:restriction base=”xs:string”>
                <xs:minLength value=”1”/>
                <xs:maxLength value=”64”/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name=”idfmt”>
        <xs:simpleType>
            <xs:restriction base=”xs:hexBinary”>
                <xs:length value=”1”/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name=”tid”>
        <xs:simpleType>
            <xs:restriction base=”xs:hexBinary”>
                <xs:minLength value=”6”/>
                <xs:maxLength value=”8”/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name=”x” type=”xs:double”/>
    <xs:element name=”y” type=”xs:double”/>
    <xs:element name=”z” type=”xs:double”/>
    <xs:element name=”bat”>
        <xs:simpleType>
            <xs:restriction base=”xs:int”>
                <xs:enumeration value=”0”/>
                <xs:enumeration value=”1”/>
                <xs:enumeration value=”3”/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name=”t” type=”xs:dateTime”/>
    <xs:element name=”cls”>
        <xs:simpleType>
            <xs:restriction base=”xs:string”>
                <xs:minLength value=”1”/>
                <xs:maxLength value=”64”/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name=”zon”>
        <xs:simpleType>
            <xs:restriction base=”xs:string”>
                <xs:minLength value=”1”/>
                <xs:maxLength value=”256”/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name=”exc”>
        <xs:simpleType>
            <xs:restriction base=”xs:string”>
                <xs:minLength value=”1”/>
                <xs:maxLength value=”64”/>
            </xs:restriction>
        </xs:simpleType>
```

```
        </xs:element>
        <xs:element name="ant">
            <xs:simpleType>
                <xs:restriction base="xs:int">
                    <xs:minInclusive value="0"/>
                    <xs:maxInclusive value="255"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="dat">
            <xs:simpleType>
                <xs:restriction base="xs:hexBinary">
                    <xs:minLength value="1"/>
                    <xs:maxLength value="123"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="alg">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:minLength value="1"/>
                    <xs:maxLength value="64"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="SLMF">
            <xs:complexType>
                <xs:sequence>
                    <xs:element ref="src" minOccurs="1" maxOccurs="1"/>
                    <xs:element ref="fmt" minOccurs="1" maxOccurs="1"/>
                    <xs:element ref="idfmt" minOccurs="1" maxOccurs="1"/>
                    <xs:element ref="tid" minOccurs="1" maxOccurs="1"/>
                    <xs:element ref="x" minOccurs="1" maxOccurs="1"/>
                    <xs:element ref="y" minOccurs="1" maxOccurs="1"/>
                    <xs:element ref="z" minOccurs="1" maxOccurs="1"/>
                    <xs:element ref="bat" minOccurs="1" maxOccurs="1"/>
                    <xs:element ref="t" minOccurs="1" maxOccurs="1"/>
                    <xs:choice>
                        <xs:element ref="cls" minOccurs="0" maxOccurs="1"/>
                        <xs:element ref="zon" minOccurs="0" maxOccurs="1"/>
                        <xs:element ref="exc" minOccurs="0" maxOccurs="1"/>
                        <xs:element ref="ant" minOccurs="0" maxOccurs="1"/>
                        <xs:element ref="dat" minOccurs="0" maxOccurs="1"/>
                        <xs:element ref="alg" minOccurs="0" maxOccurs="1"/>
                    </xs:choice>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="Push_ Events">
            <xs:complexType>
                <xs:sequence>
                    <xs:element ref="SLMF" minOccurs="1" maxOccurs="1024"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:schema>
```

5.9.5 Пример XML экземпляра класса

```
<Push_Events>
<SLMF>
    <src>MySourceA</src> <fmt>DFT</fmt>
    <idfmt>01</idfmt>
    <tid>000100BC614E</tid>
    <x>-46.2</x> <y>10</y> <z>4</z>
    <bat>0</bat> <t>2010-11-24T09:07:04-08:00</t>
    <exc>1020</exc>
</SLMF>
<SLMF>
    <src>MySourceA</src> <fmt>DFT</fmt>
    <idfmt>01</idfmt>
    <tid>000100BC614E</tid>
    <x>-58.7</x> <y>5.5</y> <z>4</z>
    <bat>0</bat> <t>2010-11-24T09:09:02-08:00</t>
    <exc>1020</exc>
</SLMF>
<SLMF>
    <src>MySourceB</src> <fmt>T</fmt>
    <idfmt>01</idfmt>
    <tid>000100BC7F01</tid>
    <x>102.75</x> <y>203.4</y> <z>4.91</z>
    <bat>1</bat> <t>2010-11-24T09:07:04.3-08:00</t>
    <cls>Laptop</cls>
</SLMF>
</Push_Events>
```

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ISO/IEC 15963	IDT	ГОСТ Р ИСО/МЭК 15963—2011 «Информационные технологии. Радиочастотная идентификация для управления предметами. Уникальная идентификация радиочастотных меток»
ISO/IEC 19762	—	*
IEEE Guidelines for use of a 48-bit Extended Unique Identifier (EUI-48™)	—	*
IEEE Guidelines for 64-bit Global Identifier (EUI-64™)	—	*
Extensible Markup Language (XML) 1.0, (Third Edition), W3C Recommendation, World Wide Web Consortium (W3C)	—	*
SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), W3C Recommendation, World Wide Web Consortium (W3C)	—	*

* Соответствующий национальный стандарт отсутствует.

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

- IDT — идентичный стандарт.

Библиография

[1] ISO 8601, Data elements and interchange formats — Information interchange — Representation of dates and times

[2] IETF RFC 854: May 1983. Telnet Protocol Specification (gost.e-ecolog.ru)

[3] IETF RFC 2616: June 1999. Hypertext Transfer Protocol -- HTTP/1.1 (gost.e-ecolog.ru rfc2616.txt)

[4] XML Schema Part 1: Structures Second Edition, W3C Recommendation, World Wide Web Consortium (W3C), Cambridge Massachusetts, 2 May 2001. (gost.e-ecolog.ru)

[5] XML Schema Part 2: Datatypes Second Edition, W3C Recommendation, World Wide Web Consortium (W3C), Cambridge Massachusetts, 2 May 2001. (gost.e-ecolog.ru)

[6] SOAP Version 1.2 Part 1: Messaging Framework, W3C Recommendation, World Wide Web Consortium (W3C), 24 June 2003. (gost.e-ecolog.ru)

[7] SOAP Version 1.2 Part 2: Adjuncts, W3C Recommendation, World Wide Web Consortium (W3C), 24 June 2003. (gost.e-ecolog.ru)

УДК 621.398(083.744):006.89 ОКС 35.040 П85

Ключевые слова: информационные технологии, системы позиционирования в реальном времени (RTLS), прикладной программный интерфейс (API)

БЗ 11—2016/173

Редактор *Е.С. Неверова*
Технический редактор *И.Е. Черепкова*
Корректор *Р.А. Ментова*
Компьютерная верстка *А.А. Ворониной*

Сдано в набор 13.06.2017. Подписано в печать 20.07.2017. Формат 60×84 1/8. Гарнитура Ариал.
Усл. печ. л. 2,79. Уч.-изд. л. 2,52. Тираж 25 экз. Зак. 1194.
Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Издано и отпечатано во ФГУП «СТАНДАРТИНФОРМ», 123001 Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru