[Elec.ru](https://www.elec.ru/)

Электротехническая библиотека Elec.ru

ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

## Н А Ц И О Н А Л Ь Н Ы Й С Т А Н Д А Р Т Р О С С И Й С К О Й Ф Е Д Е Р А Ц И И

**ГОСТР**

**МЭК 61784-3-12—**

**2016**

**Промышленные сети ПРОФИЛИ**

**Ч а с т ь 3-12**

**Функциональная безопасность полевых шин. Дополнительные спецификации для CPF 12**

**(IEC 61784-3-12:2010, IDT)**

Издание официальное

Москва Стандартинформ

2017

#### ГОСТ Р МЭК 61784-3-12—2016

**Предисловие**

1. ПОДГОТОВЛЕН Обществом с ограниченной ответственностью «Корпоративные электронные системы» (КЭЛС) на основе собственного перевода на русский язык англоязычной версии междуна­ родного стандарта, указанного в пункте 4
2. ВНЕСЕН Техническим комитетом по стандартизации ТК 58 «Функциональная безопасность»
3. УТВЕРЖДЕН И 8ВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому ре­ гулированию и метрологии России от 30 ноября 2016 г. № 1882-ст
4. Настоящий стандарт идентичен международному стандарту МЭК 61784-3-12:2010 «Промыш­ ленные сети. Профили. Часть 3-12. Функциональная безопасность полевых шин. Дополнительные спе­ цификации для CPF 12» (IEC 61784-3-12:2010, Industrial communication networks — Profiles — Part 3-12: Functional safety fieldbuses — Additional specifications for CPF 12. IDT).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных междуна­ родных стандартов соответствующие им национальные и межгосударственные стандарты, сведения о которых приведены в дополнительном приложении ДА

1. ВВЕДЕН ВПЕРВЫЕ

*Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г.* Afe *162-ФЗ «О стандартизации в Российской Федерации». Информация* об из­ менениях *к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок* — е 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* ин­ формационной системе *общего пользования* — *на официальном сайте Федерального агентства по техническому регулированию и* метрологии в *сети Интернет (*[*www.gost.ru*](http://www.gost.ru/)*)*

© Стандартинформ. 2017

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и рас­ пространен в качестве официального издания без разрешения Федерального агентства по техническо­ му регулированию и метрологии

II

#### ГОСТ Р МЭК 61784-3.12—2016

**Содержание**

1. [Область применения. 1](#_bookmark0)
2. [Нормативные ссылки. 1](#_bookmark1)
3. [Термины, определения, обозначения и сокращения. 2](#_bookmark2)
   1. Термины и определения 2
   2. Обозначения и сокращения. 6
   3. Условные обозначения. 6
4. [Обзор FSCP12/1 (CC-Link-Safety"\*). 7](#_TOC_250001)
5. [Общие положения 8](#_bookmark3)
   1. Внешние документы, предоставляющие спецификации для профиля 8
   2. Функциональные требования безопасности 8
   3. Меры безопасности. 9
   4. [Структура коммуникационного уровня безопасности. 9](#_bookmark4)
   5. Связи с FAL (и DLL. PhL). 10
6. [Услуги коммуникационного уровня безопасности. 10](#_bookmark5)
   1. Соединение FSoE. 10
   2. Цикл FSoE 11
   3. Услуги FSoE 11
7. [Протокол коммуникационного уровня безопасности. 12](#_TOC_250000)
   1. Формат PDU безопасности 12
   2. Процедура коммуникаций FSCP 12/1 17
   3. Реакция на ошибки коммуникаций. 26
   4. Таблица состояний для ведущего устройства FSoE. 27
   5. Таблица состояний для ведомого устройства FSoE. 44
8. [Управление коммуникационным уровнем безопасности 68](#_bookmark6)
   1. Обработка параметров FSCP 12/1. 68
   2. Параметры коммуникаций FSoE 68
9. [Системные требования. 69](#_bookmark7)
   1. Индикаторы и коммутаторы. 69
   2. Руководство по установке. 71
   3. Время реакции функции безопасности. 71
   4. Длительность запросов на обслуживание. 74
   5. Ограничения на вычисление характеристик системы. 74
   6. Техническое обслуживание 76
   7. Руководство по безопасности. 76
10. [Оценка 76](#_bookmark8)

Приложение А (справочное) Дополнительная информация для профилей коммуникаций

функциональной безопасности CPF 12. 77

Приложение В (справочное) Информация для оценки профилей коммуникаций

функциональной безопасности CPF 12. 82

Приложение ДА (справочное) Сведения о соответствии ссылочных международных

стандартов национальным стандартам. 83

Библиография. 84

ГОСТ Р МЭК 61784-3-12—2016

**О Введение**

* 1. **Общие положения**

Стандарт МЭК 61158. посвященный полевым шинам, вместе с сопутствующими ему стандарта­ ми МЭК 61784-1 и МЭК 61784-2 определяет набор протоколов передачи данных, которые позволяют осуществлять распределенное управление автоматизированными приложениями. 8 настоящее время технология полевых шин считается общепринятой и хорошо себя зарекомендовала. Именно поэтому появляются многочисленные расширения, направленные на еще не стандартизированные области, та­ кие как приложения реального времени, связанные с безопасностью и защитой.

Настоящий стандарт рассматривает важные принципы функциональной безопасности коммуни­ каций на основе подхода, представленного в комплексе стандартов МЭК 61508, и определяет несколь­ ко коммуникационных уровней безопасности (профилей и соответствующих протоколов) на основе про­ филей передачи данных и уровней протоколов, описанных в МЭК 61784\*1. МЭК 61784-2 и в комплексе стандартов МЭК 61158. Настоящий стандарт не рассматривает вопросы электробеэоласности и искро- безопасности.

На рисунке 1 представлена связь настоящего стандарта с соответствующими стандартами, посвя­ щенными функциональной безопасности и полевым шинам е среде машинного оборудования.

На рисунке 2 представлена связь настоящего стандарта с соответствующими стандартами, посвя­

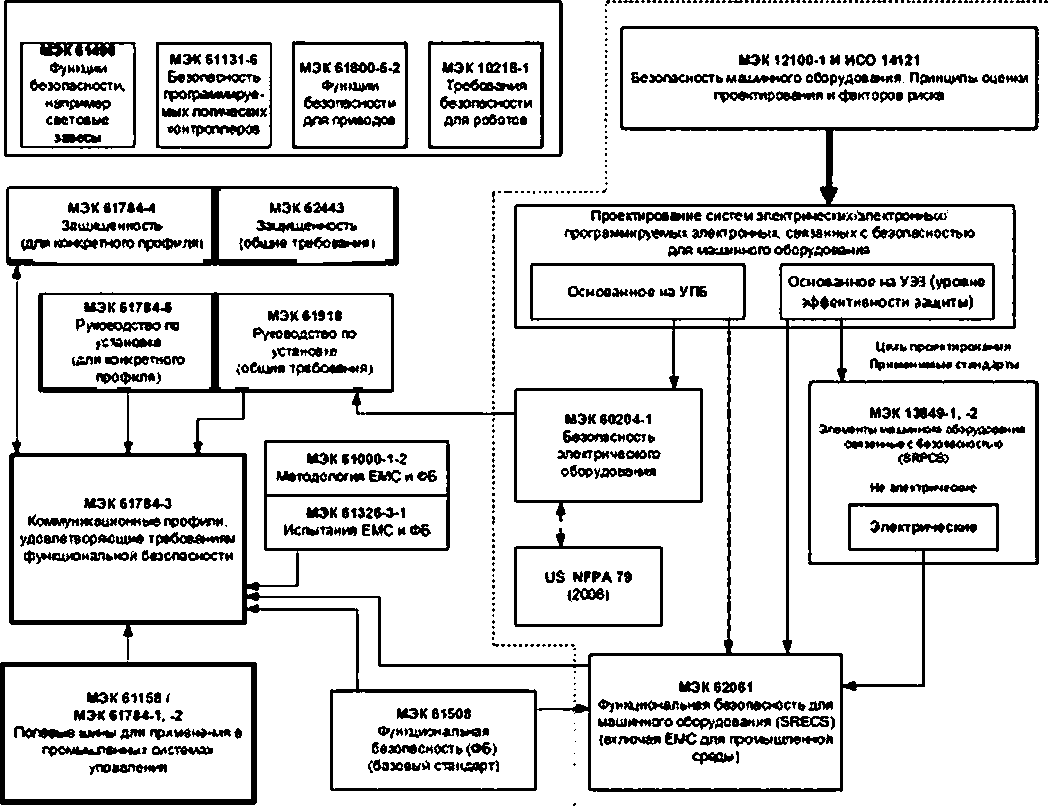
щенным функциональной безопасности и полевым шинам в области промышленных процессов.

Коммуникационные уровни безопасности, реализованные в составе систем, связанных с безопас­ ностью. в соответствии с МЭК 61508, обеспечивают необходимую достоверность при передаче сообще­ ний (информации) между двумя и более участниками, использующими полевые шины в системе, свя­ занной с безопасностью, или же достаточную уверенность в безопасном поведении при возникновении ошибок или отказов в полевой шине.

Коммуникационные уровни безопасности, определенные в настоящем стандарте, обеспечивают уверенность в том. что полевые шины могут использоваться в применениях, требующих обеспечение функциональной безопасности для конкретного уровня полноты функциональной безопасности (УПБ), для которого определен соответствующий ему профиль коммуникации, удовлетворяющий требовани­ ям функциональной безопасности.

#### IV

ГОСТ Р МЭК 61784-3-12—2016



Обомачеиия:

□ (желтый) • стандарты, связанные с безопасностью.

(голубой) - стандарты. с«язаи>ме с полевыми шинами

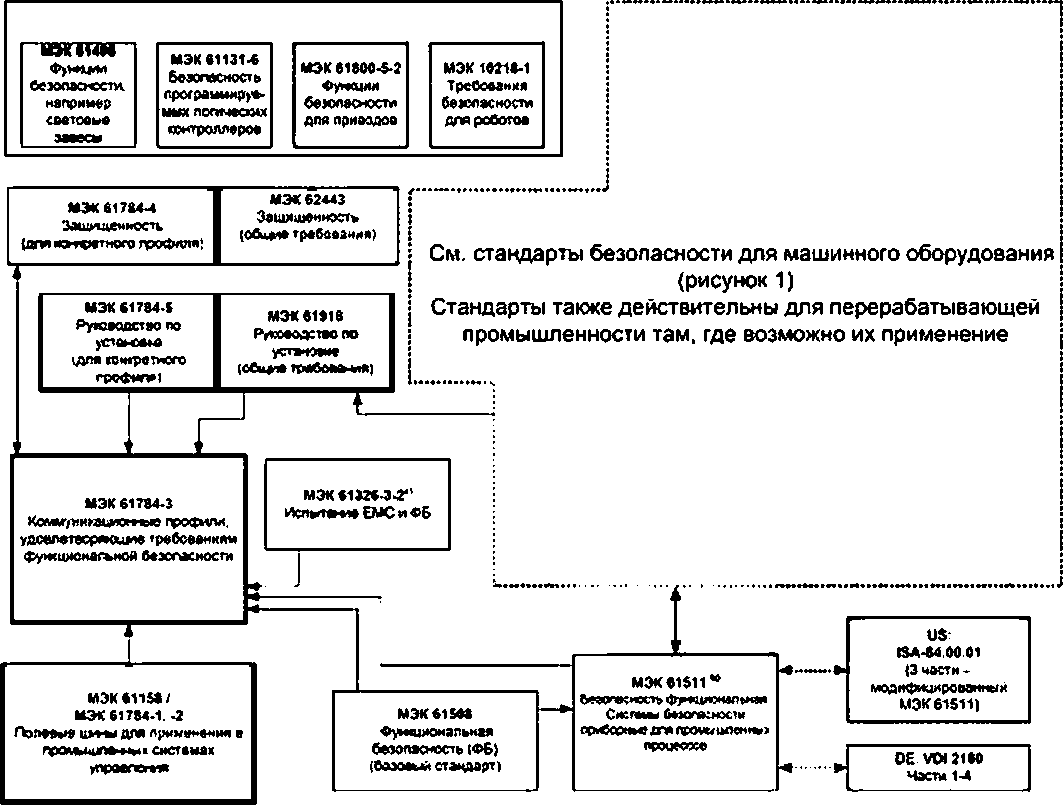
| | (бледно желтый) •»«стоящий стандарт

Примечание — Подразделы б.7.6.4 {высокая степень сложности) и 6.7.8.1.6 {низкая степень сложности) МЭК 62061 устанавливают связь между PL (Категорией) и УПБ.

Рисунок 1 — Связь МЭК 61158-3 с другими стандартами (машинное оборудование)

V

#### ГОСТ Р МЭК 61784-3-12—2016



Обозначения:

**О** (хоптый) • стандарты связанные с безопасностью

 (голубой) ■ стандарты связанные с полевыми шинами,

**ш** <6л«»«жетм»)- настоящий стандарт

Рисунок 2 — Связь МЭК 61156-3 с другими стандартами (промышленные промессы)

Результирующий УПБ. заявляемый для системы, зависит от реализации выбранного профиля коммуникации, удовлетворяющего требованиям функциональной безопасности, внутри зтой системы. Но реализации профиля коммуникации, удовлетворяющего требованиям функциональной безопасно­ сти. в стандартном устройстве недостаточно для того, чтобы устройство считалось устройством без­ опасности.

Настоящий стандарт описывает:

* основные принципы реапизации требований комплекса стандартов МЭК 61508 для связанной с безопасностью передачи данных, включая возможные сбои при передаче данных, меры по устранению неисправностей и факторы, влияющие на полноту данных;
* индивидуальные описания профилей, удовлетворяющих требованиям функциональной без­ опасности. для нескольких семейств профилей передачи данных, представленных в МЭК 61784-1 и МЭК 61784-2;
* расширения уровня безопасности до служб передачи данных и разделов протоколов в стандар­ тах комплекса МЭК 61158.

#### VI

ГОСТ Р МЭК 61784-3.12—2016

* 1. Декларация патента

Международный электротехнический комитет (МЭК) обращает внимание на то, что соблюдение требований настоящего стандарта может включать использование патентов, относящихся к профилям коммуникаций, соответствующих требованиям функциональной безопасности. Патенты для семейства 1 приведены ниже, где обозначение (хх) указывает на держателя патента:

|  |  |  |
| --- | --- | --- |
| DE 10 2004 044 764.0 | [SI] | Datenubertragungsverfahren und Automatisierungssystem zum Einsatz eines solchen Datenubertragungsverfahrens |
| ЕР 05 733 921.0 | [SI] | Sicherheitssteuerung |

МЭК не занимается подтверждением обоснованности, подтверждением соответствия и областью применения прав данных патентов.

Правообладатели на данные патенты заверили МЭК. что они готовы рассмотреть использование лицензий на разумных и недискриминационных условиях и положениях с заявителями по всему миру. Такие заявления обладателей прав на данные патенты зарегистрированы в МЭК.

Информация доступна в:

[Si] Beckhoff Automation GmbH

Eiserstrasse 5. 33415 Veri GERMANY

Обращаем ваше внимание на то. что некоторые элементы настоящего стандарта могут быть субъ­ ектом патентных прав, отличных от указанных ранее. МЭК не несет ответственности за идентификацию (частично или полностью) подобных патентных прав.

#### VII

ГОСТ Р МЭК 61784-3-12—2016

Н А Ц И О Н А Л Ь Н Ы Й С Т А Н Д А Р Т Р О С С И Й С К О Й Ф Е Д Е Р А Ц И И

Промышленные сети ПРОФИЛИ

Часть 3-12

Функциональная безопасность полевых шин. Дополнительные спецификации для CPF 12

industrial communication networks. Profiles. РаП 3-12. Functional safety fieidbuses. Additional specifications for CPF 12

Дата введения — 2018—01—01

## Область применения

Настоящий стандарт описывает коммуникационный уровень безопасности (услуги и протокол) на основе CPF 12. представленного в МЭК 61784-2 и МЭК 61158. Тип 12. Настоящий стандарт идентифи­ цирует принципы коммуникаций, удовлетворяющих требованиям функциональной безопасности, опре­ деленных в МЭК 61784-3, что важно для этого коммуникационного уровня безопасности.

Примечание — Настоящий стандарт не затрагивает вопросы электро безопасности и и скро безопасно­ сти. Элвктробеэоласность связана с угрозами, такими как электрический шок. Искробеэопасность связана с угро­ зами. относящимися к возможным взрывам в атмосфере.

Настоящий стандарт определяет механизмы передачи важных для безопасности сообщений меж­ ду участниками распределенной сети, использующей технологию полевых шин. в соответствии с требо­ ваниями функциональной безопасности, представленными е комплексе МЭК 615084 Эти механизмы могут широко использоваться в промышленности, например в управлении процессом автоматизации производства и в машинном оборудовании.

Настоящий стандарт содержит руководства как для разработчиков, так и для оценщиков соответ­ ствующих приборов и систем.

Примечание — Результирующий УПБ. заявляемый для системы, зависит от реализации выбранно­ го профиля коммуникации, удовлетворяющей требованиям функциональной безопасности, внутри этой системы. Но в соответствии с настоящим стандартом реализации выбранного профиля коммуникации, удовлетворяющей требованиям функциональной безопасности, в стандартном устройства недостаточно для того, чтобы устройство считалось устройством безопасности.

## Нормативные ссылки

Ссылки на следующие незаменимые для данного документа стандарты представлены ниже. Для датированныхссылок применяют только указанное издание ссылочного документа, для недатированных ссылок применяют последнее издание ссылочного документа (включая асе его изменения).

IEC 60204-1. Safety of machinery — Electrical equipment of machines — Part 1: General require­ ments (Безопасность машинного оборудования. Электрическое оборудование машин. Часть 1. Общие требования)

IEC 61000-6-2, Electromagnetic compatibility (EMC) — Part 6-2: Generic standards — Immunity for industrial environments (Совместимость технических средств электромагнитная. Устойчивость к электро- магнитным помехам технических средств, применяемых в промышленных зонах. Часть 6-2. Требования и методы испытаний)

1> Далее в настоящем стандарте используется «МЭК 61506» вместо «комплекс МЭК 61508».

Издание официальное

1

#### ГОСТ Р МЭК 61784-3-12—2016

IEC 61131\*2. Programmable controllers — Part 2: Equipment requirements and tests (Программируе­ мые контроллеры. Часть 2. Требования к оборудованию и тестирование)

IEC 61156-2, Industrial communication networks — Fieidbus specifications — Part 2: Physical layer specification and service definition (Промышленные сети связи. Спецификации полевых шин. Часть 2: Спецификация физического уровня и определение сервиса)

IEC 61158-3-12. Industrial communication networks — Fieidbus specifications — Part 3-12: Data-link layer service definition — Type 12 elements (Промышленные сети связи. Спецификации полевых шин. Часть 3-12: Определение сервиса канального уровня. Элементы типа 12)

IEC 61158-3-12. Industrial communication networks — Fieidbus specifications — Part 3\*12: Data-link layer protocol definition — Type 12 elements (Промышленные сети связи. Спецификации полевых шин. Часть 3-12: Определение протокола канального уровня. Элементы типа 12)

IEC 61158-5-12, Industrial communication networks — Fieidbus specifications — Part 5-12: Application layer service definition — Type 12 elements (Промышленные сети связи. Спецификации полевых шин. Часть 5-12: Определение сервиса прикладного уровня. Элементы типа 12)

IEC 61158-6-12, Industrial communication networks — Fieidbus specifications — Part 6-12: Application layer protocol specification — Type 12 elements (Промышленные сети связи. Спецификации полевых шин. Часть 6-12: Спецификация протокола прикладного уровня. Элементы типа 12)

IEC 61326-3-1, Electrical equipment for measurement, control and laboratory use — EMC requirements — Part 3-1: Immunity requirements for safety-related systems and for equipment intended to perform safety related functions (functional safety) — General industrial applications (Электрооборудование для измере­ ний. управления и лабораторного применения. Часть 3-1. Требования защищенности для систем, свя­ занных с безопасностью, и для оборудования, предназначенного для выполнения функций, связанных с безопасностью (функциональная безопасность). Общие промышленные приложения)

IEC 61326-3-2. Electrical equipment for measurement, controiand laboratory use—EMC requirements — Part 3-2: Immunity requirements for safety-related systems and for equipment intended to perform safety related functions (functional safety) — Industrial applications with specified electromagnetic environment (Электрооборудование для измерений, управления и лабораторного применения. Часть 3\*1. Требова­ ния защищенности для систем, связанных с безопасностью, и для оборудования, предназначенного для выполнения функций, связанных с безопасностью (функциональной безопасностью). Промышлен­ ные приложения с определенной электромагнитной средой)

IEC 61508 (all parts), Functional safety of eiecthcal/etectronicfprogrammable electronic safety-related systems (Функциональная безопасность систем электрических/электронных/программируемых элек­ тронных. связанных с безопасностью)

IEC 61784-2. Industrial communication networks — Profiles — Part 2: Additional fieidbus profiles for real-time networks based on ISO/IEC 8802-3 (Промышленные сети. Профили. Часть 2. Дополнительные профили полевых шин для сетей реального времени, основанные на ИСО/МЭК 8802-3)

IEC 61784-3. Industrial communication networks —• Profiles — Part 3: Functional safety fieidbuses — General rules and profile definitions (Промышленные сети. Профили. Часть 3. Функциональная безопас­ ность полевых шин. Общие правила и определения профиля)

IEC 61918. Industrial communication networks — Installation of communication networks in industrial premises (Промышленные сети. Установка сетей связи в промышленных помещениях)

## Термины, определения, обозначения и сокращения

* 1. Термины и определения

В настоящем стандарте используются следующие термины и определения:

* + 1. Общепринятые термины и определения
       1. готовность (availability): Вероятность того, что в течение заданного промежутка времени в автоматизированной системе не наблюдается неисправных состояний в системе, приводящих к потере производительности.
       2. черный канал (black channel): Канал связи, для которою отсутствуют доказательства тою. что проектирование и подтверждение соответствия были выполнены в соответствии с МЭК 61508.
       3. канал связи (communication channel): Логическое соединение между двумя оконечными точками в коммуникационной системе.
       4. коммуникационная система (communication system): Система (устройство), состоящая из технических средств, программною обеспечения и среды распространения, которая обеспечивает передачу сообщений (прикладной уровень по ИСО/МЭК 7498) от одного приложения другому.

2

#### ГОСТ Р МЭК 61784-3.12—2016

* + - 1. соединение (connection): Логическое связывание между двумя прикладными объектами в одном или е разных устройствах.
      2. циклический контроль избыточности [Cyclic Redundancy Check (CRC)] Получаемые из блока данных (значений) избыточных данных, которые запоминаются и передаются вместе с этим бло­ ком данных, для обнаружения искажения данных. Процедуре (метод), использующаяся для вычисле­ ния избыточных данных.

Примечания

1 Термины «CRC код» и «CRC подпись» и обозначения, такие как «CRC 1» и «CRC 2». также могут приме- мяться в данном стандарте в отношении избыточных данных.

2 См. также [32]. [33].

* + - 1. ошибка (error) Расхождение между вычисленным, наблюдаемым или измеренным значе­ нием или условием и истинным, установленным или теоретически верным значением или условием.

[МЭК 61506-4:2010), (МЭК 61156}

Примечания

1. Ошибки могут вознжнуть вследствие ошибок проектирования аппаратных средств/ программного обеспече­ ния и/или вследствие искажения данных, вызванного электромагнитными помехами и/или другими воздействиями.
2. Ошибки не обязательно являются причиной отказов или сбоев.
   * + 1. отказ (failure) Прекращение способности функционального блока выполнять необходимую функцию либо функционирование этого блока любым способом, отличным от требуемого.

Примечание — В МЭК 61508-4 приведено такое же определение, но дополнено примечаниями.

[МЭК 61508-4:2010, модифицирован]. [ИСО/МЭК 2382-14.01.11, модифицирован]

Примечание — Причиной отказа может служить ошибка (например, проблема, связанная с проектиро­ ванием программного обеспечвния/аппарагных средств или с нарушением при передаче сообщений).

* + - 1. сбой (fault): Ненормальный режим, который может вызвать снижение или потерю способ­ ности функционального блока выполнять требуемую функцию.

Примечание — Международный электротехнический словарь (191-05-01) определяет «сбой» как состояние, характеризуемое неспособностью выполнить необходимую функцию, исключая неспособность, воз­ никающую во время профилактических работ или других плановых мероприятий, либо в результате недостатка внешних ресурсов.

[МЭК 61508-4:2010. модифицирован]. [ИСО/МЭК 2382-14.01.10, модифицирован]

* + - 1. полевая шина (fieldbus): Коммуникационная система, основанная на последовательной передаче данных и применяющаяся в промышленной автоматизации или приложениях управления процессами.
      2. система полевых шин (fieldbus system): Система, использующая полевую шину с под­ ключенными устройствами.
      3. кадр (frame): Упрощенный синоним для DLPDU (Блок данных протокола канала передачи

данных)

* + - 1. последовательность проверки кадра [frame check sequence (FCS)]: Дополнительные данные, полученные из блока данных DLPDU (кадра) с помощью хеш-функции, которые запоминаются и передаются вместе с этим блоком данных для обнаружения искажения данных.

Примечания 1 Значение FCS может быть получено, испогъзуя. например. CRC или другую хеш-функцию. 2 См. также [34]. [35].

* + - 1. хеш-функция (hash function): (Математическая) функция, которая преобразует значения из (вероятно, очень) большого набора значений е (обычно) меньший диапазон значений.

Примечания

1. Хеш-функции могут применяться для обнаружения искажений данных.
2. Распространенные хеш-функции включают в себя контроль четности, вычисление контрольной суммы или

CRC.

[МЭК/ТО 62210. модифицирован]

* + - 1. опасность (hazard): Состояние или набор условий в системе, которые вместе с другими,

связанными с этими, условиями неизбежно приведут к причинению вреда человеку, имуществу или окружающей среде.

3

#### ГОСТ Р МЭК 61784-3-12—2016

* + - 1. ведущее устройство (master): Активный объект коммуникации, способный инициировать и управлять во времени коммуникационной деятельностью других станций, которые могут быть как ведущими, так и ведомыми.
      2. сообщение (message): Упорядоченные последовательности октет, предназначенные для передачи информации.

(ИСО/МЭК 2382-16.02.01, модифицирован].

* + - 1. уровень эффективности защиты; УЭЗ [performance level (PL)]: Дискретный уровень, применяющийся для определения способности связанных с безопасностью частей системы управле­ ния выполнять функцию безопасности в прогнозируемых условиях.

[ИСО 13849-1]

* + - 1. защитное сверхнизкое напряжение (protective extra-low-voltage. PELV): Электрическая цепь, в которой значение напряжения не может превышать среднеквадратичное значение переменного напряжения в 30 В. пиковое напряжение 42.4 В или постоянное напряжение 60 8 при нормальных усло­ виях и одиночном сбое за исключением короткого замыкания на землю в других цепях.

Примечание — Электрическая цепь PELV аналогична цепи SELV с защитным заземлением.

[МЭК 61131-2]

* + - 1. избыточность (redundancy): существование более одного средства выполнения необхо­ димой функции или представления информации.

Примечание — В МЭК 61508-4 такое же определение, но дополнено примером и примечаниями.

[МЭК 61508-4:2010. модифицирован]. [ИСО/МЭК 2382-14.01.12. модифицирован)

* + - 1. надежность (reliability): Вероятность того, что автоматизированная система может вы­ полнять требующуюся функцию в заданных условиях на протяжении заданного промежутка времени

(М2>.

Примечания

1. Принято считать, что автоматизированная система в состоянии выполнять данную требующуюся функцию в начале заданного промежутка времени.
2. Понятие «надежность» также используется для обозначения показателя надежности, измеряемого данной вероятностью.
3. На протяжении среднего времени между отказами (MTBF) или среднего времени до отказа (MTTF) вероят­ ность того, что автоматизированная система вьлолнит требующуюся функцию, уменьшается.
4. Надежность отличается от готовности.

[МЭК 62059-11, модифицирован]

* + - 1. риск (risk) Сочетание вероятности события причинения вреда и тяжести этого вреда.

Примечание — Более подробно эго понятие обсуждается в приложении А МЭК 61508-5:2010.

[МЭК 61508-4:2010]. [ИСО/МЭК Руководство 51:1999. определение 3.2]

* + - 1. коммуникационный уровень безопасности. КУБ (safety communication layer. SCL): Уро­ вень коммуникации, включающий все необходимые меры для обеспечения безопасной передачи ин­ формации в соответствии с требованиями МЭК 61508.
      2. данные безопасности (safety data): Данные, передаваемые через безопасную сеть, ис­ пользуя протокол безопасности.

Примечание — Коммуникационный уровень безопасности не гарантирует безопасность самой инфор­ мации. а только го. что она передается безопасно.

* + - 1. устройство безопасности (safety device): Устройство, спроектированное в соответствии с МЭК 61508 и реализующее профиль коммуникации, удовлетворяющий требованиям функциональной безопасности.
      2. безопасное сверхнизкое напряжение (safety extra-low-voitage. SELV): Электрическая цель, в которой значение напряжения не может превышать среднее квадратическое значение перемен­ ного напряжения в 30 В. пиковое напряжение 42.4 В или постоянное напряжение 60 В при нормальных условиях и одиночном сбое, включая короткое замыкание на землю в других цепях.

Примечание — Цепь SELV не подсоединена к защитному заземлению.

[МЭК 61131-2]

* + - 1. функция безопасности (safety function): Функция, реализуемая Э/Э/ПЭ (электрической, электронной, программируемой электронной) системой, связанной с безопасностью, или другими мв-

4

#### ГОСТ Р МЭК 61784-3.12—2016

рами по снижению риска, предназначенная для достижения или поддержания безопасного состояния управляемого оборудования по отношению к конкретному опасному событию.

Примечание — В МЭК 61508-4 такое же определение, но дополнено примером и примечанием.

(МЭК 61508-4:2010, модифицирован]

* + - 1. время реакции функции безопасности (safety function response time): Наихудшее время между срабатыванием датчика системы безопасности, подключенного к полевой шине, и достижением соответствующего безопасного состояния с помощью необходимого исполнительного устройства этой системы безопасности при наличии ошибок или отказов в канале функции безопасности.

Примечание —Данная концепция введена в МЭК 61784-3:2010 и реализуется профилем коммуникации, удовлетворяющим требованиям функциональной безопасности, определенным в настоящем стандарте.

* + - 1. уровень полноты безопасности, УПБ (safety integrity level SIL): дискретный уровень (принимающий одно из четырех возможных значений), соответствующий диапазону значений полноты безопасности, при котором уровень полноты безопасности, равный 4. является наивысшим уровнем полноты безопасности, а уровень полноты безопасности, равный 1. соответствует наименьшей полноте безопасности.

Примечания

1. Целевые значения отказов (см. МЭК 61508-4:2010. п. 3.5.17) для четырех уровней полноты безопасности указаны в МЭК б1508-1:2010. таблицы 2 и 3.
2. Уровни полноты безопасности используют при определении требований полноты безопасности для функ­ ций безопасности, которые должны быть распределены по Э/Э/ПЭ системам, связанным с безопасностью.
3. Уровень полноты безопасности (УПБ) не является свойством системы, подсистемы, элемента или компо­

нента. Правильная интерпретация фразы «УПБ системы, связанной с безопасностью, равен л» (где л « 1. 2. 3 или

1. означает: система потенциально способна к реализации функций безопасности с уровнем полноты безопасно­ сти до значения, равного *п.*

(МЭК 61508-4:2010)

* + - 1. мера безопасности (safety measure) <в данном стандарте» средство управления воз­ можными ошибками коммуникаций, спроектированное и реализованное в соответствии с требования­ ми МЭК 61508.

Примечания

1. На практике, как правило, объединяют несколько мер безопасности для достижения требуемого уровня полноты безопасности.
2. Ошибки коммуникаций и связанные с ними меры безопасности подробно рассмотрены в МЭК 61784-3:2010, 5.3 и 5.4.
   * + 1. приложение, связанное с безопасностью (safety-related application): Программы, раз­ работанные е соответствии с МЭК 61508 и удовлетворяющие требованиям УПБ приложения.
       2. система, связанная с безопасностью (safety-related system): система, выполняющая

функцию безопасности в соответствии с МЭК 61508.

* + - 1. ведомое устройство (slave): Пассивный объект коммуникации, способный принимать сообщения и отправлять их в ответ на другой объект коммуникации, который может быть ведомым или ведущим.
      2. ложное аварийное отключение (spurious trip): Аварийное отключение, вызванное систе­ мой безопасности, без запроса от процесса.
    1. CPF 12. Дополнительные термины и определения
       1. отказоустойчивые данные (fail-safe data): Выражение для данных, которые, е случае ини­ циализации или ошибки, принимают заранее определенное значение.

Примечание — В настоящем стандарте значение отказоустойчивых данных всегда должно равняться

0 

* + - 1. соединение FSoE (FSoE Connection): Уникальная связь между ведущим устройством

FSoE и ведомым устройством FSoE.

* + - 1. цикл FSoE (FSoE Cycle): Коммуникационный цикл, включающий один PDU ведущего устройства и соответствующее PDU ведомого устройства.
      2. Безопасный ввод (Safelnput): Данные процесса безопасности, передаваемые от ведомо­ го устройства FSoE ведущему устройству FSoE.

5

#### ГОСТ Р МЭК 61784-3-12—2016

* + - 1. Безопасный вывод (SafeOutput): Данные процесса безопасности, передаваемые от веду­ щего устройства FSoE ведомому устройству FSoE.
      2. POU ведущего устройства безопасности (Safety Master PDU): PDU безопасности, пере­ даваемое от ведущего устройства FSoE ведомому устройству FSoE.

3.1.27 PDU ведомого устройства безопасности (Safety Slave PDU): PDU безопасности, пере\* даваемое от ведомого устройства FSoE ведущему устройству FSoE.

* 1. Обозначения и сокращения
     1. Общие обозначения и сокращения

|  |  |  |
| --- | --- | --- |
| Сокращение | Полное выражение | Источник |
| СР | Профиль коммуникаций | {МЭК 61784-1) |
| CPF | Семейство профилей коммуникации | [МЭК617&4-1] |
| CRC | Циклический контроль избыточности | — |
| DLL | Уровень канала данных | [ИСО/МЭК 7498-1] |
| DLPDU | Блок данных протокола канала передачи данных | — |
| ЭМС | Электромагнитная совместимость | — |
| УО | Управляемое оборудование | {МЭК 61508-4:2010] |
| Э/Э/ПЭ | Электрические/эпектронные/программируемые электронные | {МЭК 61508-4:2010] |
| FAL | Прикладной уровень полевой шины (Fietdbus АррЬсаЬоп Layer) | {МЭК61158-5] |
| FCS | Последовательность проверки кадра | — |
| ФБ | Функциональная безопасность | — |
| FSCP | Профиль коммуникации, удовлетворяющий требованиям функциональной безопасности | — |
| MTBF | Среднее время между отказами | — |
| MTTF | Среднее время до отказа | — |
| PDU | Блок данных протокола | {ИСО/МЭК 7498-1] |
| PELV | Защитное сверхнизкое напряжение | — |
| PhL | Физический уровень | {ИСО/МЭК 7498-1] |
| PL | Уровень эффективности защиты | {ИСО 13849-1] |
| PLC | Программируемый логический контроллер | — |
| SCL | Коммуникационный уровень безопасности | — |
| SELV | Безопасное сверхнизкое напряжение | — |
| УПБ | Уровень полноты безопасности | {МЭК 61508-4:2010] |

* + 1. CPF 12: Дополнительные термины и определения SIS — Инструментальная система безопасности (safety instrumented systems)

|  |  |
| --- | --- |
| Сокращение | Полное выражение |
| ASIC | Специализированная интегральная схема |
| FSoE | Отказоустойчивость по CPF 12 |
| 10 | Идентификатор |
| UML | Унифицированный язык моделирования |

* 1. Условные обозначения

Условные обозначения, используемые для описаний объектов, услуг и протоколов, рассмотрены в МЭК 61158-3-12. МЭК 61158-4-12. МЭК 61158-5-12 и МЭК 61118-6-12.

6

#### ГОСТ Р МЭК 61784-3.12—2016

При необходимости для описания концпеций настоящий стандарт использует структурные схемы и UML диаграммы последовательности действий.

Состояния е диаграммах состояний представлены прямоугольниками, переходы состояний показаны в виде стрелок. Названия состояний и переходов диаграммы состояний соответствуют названиям в текстовом списке переходов состояний. Третья колонка содержит действие(я). которые должны быть выполнены. Последняя колонка содержит следующее состояние.

Таблица 1 — Элементы описания конечного автомата

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следу «шее состояние |
|  |  |  |  |

Каждое состояние и его переход описаны в отдельном подразделе. Для каждого события, которое может произойти в состоянии, вводится дополнительный подраздел.

## Обзор FSCP 12/1 (CC-Link Safety™)

Серия 12 профилей коммуникаций (общеизвестная как EtherCAT™’))определяет профили комму\* кикаций. основанные на МЭК 61158\*2.Тип 12. МЭК 61158\*3\*12, МЭК 61158\*4\*12, МЭК 61158\*5\*12 и МЭК 61158\*6\*12.

Базовые профили СР 12/1 и СР 12/2 определены в МЭК 61784\*2. Коммуникационный профиль, удовлетворяющий требованиям функциональной безопасности. FSCP 12/1 (Safety-over-EtherCAT™1\*) серии CPF 12 основан на базовых профилях CPF 12 из МЭК 61784\*2, а также на спецификациях комму\* никационного уровня безопасности, определенных е настоящем стандарте.

FSCP *12/1* описывает протокол для пересыпки данных безопасности до УПБ 3 между устройства\* ми FSCP 12/1. PDU безопасности пересылаются подчиненной полевой шиной, которая не включена в требования обеспечения безопасности, так как она может считаться черным каналом. PDU безопасно\* сти, которыми обмениваются два партнера по связи, воспринимаются подчиненной полевой шиной как данные процесса, которыми они циклически обмениваются.

FSCP 12/1 использует уникальную связь ведущего/ведомого между ведущим и ведомым устрой\* ствами FSoE, она называется соединение FSoE (рисунок 3). В соединении FSoE каждое устройство, как только получает новое сообщение от устройства-партнера, возвращает только свое собственное новое сообщение. Полный путь пересылки между ведомым устройством FSoE и ведущим устройством FSoE наблюдается отдельным сторожевым таймером, установленным на обоих устройствах в каждом цикле FSoE.

ведущее устройство FSoE может обрабатывать более одного соединения FSoE для поддержки нескольких ведомых устройств FSoE.

ElhefCAT™ и Safety-оver-EtherCAT™ являются торговыми марками BecKhoff. Vert. Данная информация приведена для удобства использования данного международного стандарта и не означает, что МЭК поддерживает мнение обладателя торговой марки или его продукцию. Соответствие этому стандарту не требует использова­ ния наименований EtherCAT™ и Safety\*over\*EtherCAT,M. Использование торговых марок EtherCAT"\* и Safety-over\* EtherCAT™ требует разрешения со стороны Beckhoff, Vert.

7

#### ГОСТ Р МЭК 61784-3-12—2016

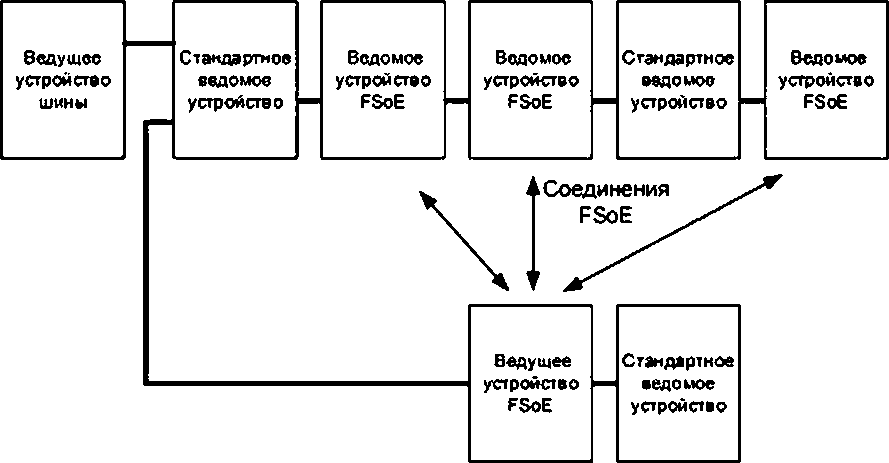


Рисунок 3 — Базовая система FSCP 12/1

Целостность передачи данных безопасности обеспечивается следующим образом:

* + номер сеанса для обнаружения буферизации завершенной последовательности загрузки:
  + порядковый номер для определения коммутации, повторения, внесения или потери целых сообщений;
  + идентификация уникального соединения для безопасного обнаружения сообщения, ошибочно направленного по другому маршруту, с помощью уникальной связи адреса;
  + сторожевой оперативный контроль для безопасного обнаружения недопустимых задержек на

коммуникационном пути:

* + проверка целостности данных циклическим избыточным кодом для обнаружения искажения сообщений от источника приемнику.

Смены состояний инициируются ведущим устройством FSoE и подтверждаются ведомым устройством FSoE. Машина состояний FSoE также предполагает обмен информацией и ее проверку для коммуникационной связи.

## Общие положения

* 1. Внешние документы, предоставляющие спецификации для профиля Нижеприведенный документ полезен для понимания конструкции протокола FSCP 12/1:
  + GS-ET-26 {33J.
  1. Функциональные требования безопасности

Следующие требования применяются к разработке устройств, реализующих протокол FSCP 12/1.

Те же требования были использован в разработке FSCP 12/1.

* + Протокол FSCP 12/1 спроектирован для поддержки уровня полноты безопасности 3 (УПБ 3) (см. МЭК 61508).
  + Реализации FSCP 12/1 должны соответствовать МЭК 61508.
  + Базовые требования для разработки протокола FSCP 12/1 содержатся в МЭК 61784-3.
  + Протокол FSCP 12/1 реализуется, используя метод черного канала: отсутствует какая-либо за­ висимость от стандартных коммуникационных профилей CPF 12. Оборудование для передачи данных, такое как контроллеры. ASIC схемы, каналы, соединительные устройства и т. д., должны избегать мо­ дификаций.
* Окружающие условия должны соответствовать общим требованиям автоматизации, в основном стандартам МЭК 61326-3\*1. для испытаний запаса безопасности, если отсутствуют конкретные стан­ дарты для самого изделия.

8

#### ГОСТ Р МЭК 61784-3.12—2016

* + Коммуникации безопасности и коммуникации, не связанные с безопасностью, должны быть не­ зависимы друг от друга. Тем не менее, устройства, не связанные с безопасностью, и устройства без­ опасности должны быть способны использовать один коммуникационный канал.
* Реализация протокола FSCP 12/1 должна быть ограничена оконечными устройствами коммуни­ каций (ведущим устройством FSoE и ведомым устройством FSoE).
  + Между ведомым устройством FSoE и его ведущим устройством FSoE должна всегда присутство­ вать коммуникационная связь 1:1.
  + Коммуникации безопасности не должны ограничивать минимальное время цикла коммуникаци­ онной системы.
  1. Меры безопасности

Меры обеспечения безопасности, используемые в FSCP12/1 для обнаружения коммуникационных ошибок, перечислены в таблице 2. Меры безопасности должны обрабатываться и контролироваться для каждого отдельного устройства безопасности.

Таблица 2 — Коммуникационные ошибки и механизмы их обнаружения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ошибки коммуникаций | Меры обеспечения безопасности | | | | |
| Порядковый номер  (см 7 >.3.4) | Временное ожидание (см.6.2)41 | Аутентификация соединения (см.7.2.2.4)М | Сообщение обратной свя­ зи <см. 7.2.1) | Обеспечение целостности данных  (см. 7.1.3} |
| Искажение |  |  |  |  | X |
| Непреднамеренное по­ вторение | X |  |  |  | X |
| Неверная последова­ тельность | X |  |  |  | X |
| Потеря | X | X |  | X | X |
| Недопустимая задержка |  | X |  | X | X |
| Внесение | X |  |  |  | X |
| Подмена |  | X |  | X | X |
| Адресация |  |  | X |  |  |
| Периодически повторя­ ющие отказы памяти в коммутаторах | X |  |  |  | X |

а> В настоящем стандарте этот экземпляр именуется как «Сторожевой таймер FSoE».

ь> В настоящем стандарте этот экземпляр именуется как «Ю соединения FSoE».

* 1. Структура коммуникационного уровня безопасности

Протокол FSCP 12/1 расположен поверх стандартного сетевого протокола. На рисунке 4 показано, как протокол связан с уровнем CPF 12. Уровень безопасности принимает данные безопасности, посту­ пающие от приложения, связанного с безопасностью, и передает эти данные посредством протокола FSCP 12/1.

PDU безопасности, содержащий данные безопасности и требуемые меры обнаружения ошибок, входит в объекты данных процесса коммуникаций (PDO). Отображение в данных процесса коммуни­ кационной системы и запуск коммуникационного конечного автомата не являются частью данного про­ токола безопасности.

9

#### ГОСТ Р МЭК 61764-3-12—2016

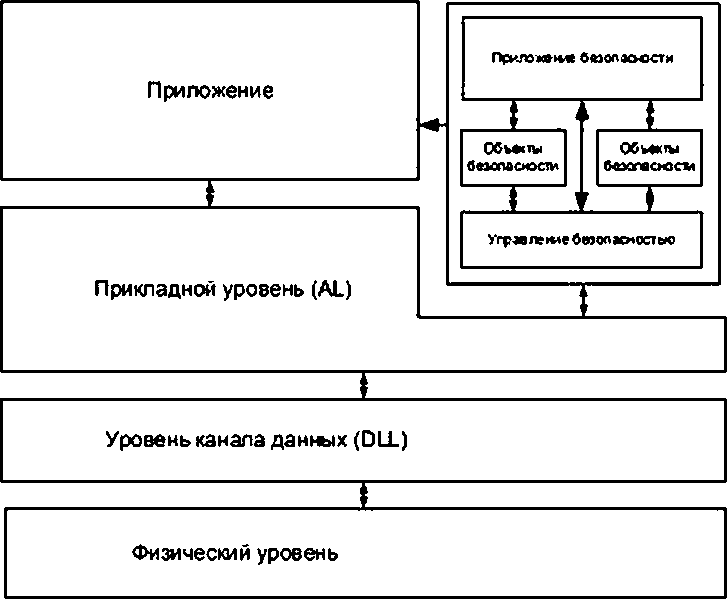


Рисунок 4 — Архитектура программного обеспечения FSCP 12/1

Вычисление вероятности возникновения остаточной ошибки для протокола FSCP 12/1 не связано с механизмами обнаружения ошибок коммуникационной системы. Это означает, что протокол можно также использовать для передачи в других коммуникационных системах. Может использоваться любой канал передачи, включая системы полевых шин. Ethernet или подобные сстемы передачи. волоконно- оптические кабели, медные провода или даже радиоканалы.

* 1. Связи с FAL (и DLL, PhL)
     1. Общие положения

Коммуникационный уровень безопасности спроектирован для использования совместно с коммуникационными профилями CPF 12. Но он не ограничивается лишь этим коммуникационным про\* филем.

* + 1. Типы данных

Профили, определенные в настоящем стандарте, поддерживают все типы данных CPF 12. задан­ ные в МЭК 61158-5-12.

## Услуги коммуникационного уровня безопасности

* 1. Соединение FSoE

Соединение между двумя коммуникационными партнерами по FSCP 12/1 (узлами с FSCP 12/1) именуется соединением FSoE. В соединении FSoE один коммуникационный партнер всегда является ведущим устройством FSoE. а другой ведомым устройством FSoE.

Ведущее устройство FSoE инициализирует соединение FSoE после включения литания или после сбоя коммуникации, в то время как ведомое устройство FSoE ограничивается ответами. Ведущее устройство FSoE устанавливает параметры коммуникаций, связанные с безопасностью, а также (не обязательно) параметры приложения, связанного с безопасностью, ведомого устройства FSoE.

Данные процесса безопасности, передаваемые от ведущего устройства FSoE ведомому устрой­ ству FSoE. называются безопасными выводами. Данные безопасности, передаваемые от ведомою устройства FSoE ведущему устройству FSoE. называются безопасными вводами.

PDU безопасности, передаваемый от ведущего устройства FSoE ведомому устройству FSoE. называется PDU безопасности ведущего устройства. PDU безопасности, передаваемый от ведомого устройства FSoE ведущему устройству FSoE. именуется PDU безопасности ведомого устройства.

10

#### ГОСТ Р МЭК 61784-3.12—2016

* 1. Цикл FSoE

Ведущее устройство FSoE отправляет PDU безопасности ведущего устройства ведомому устройству безопасности FSoE и запускает сторожевой таймер FSoE.

После проверки целостности PDU безопасности ведомое устройство FSoE передает безопасные выводы приложению безопасности. Оно вычисляет PDU безопасности ведомого устройства с безопасными выводами, полученными от приложения безопасности и отправляет этот PDU ведущему устройству FSoE. Ведомое устройство FSoE также запускает свой сторожевой таймер FSoE. Это по­ казано на рисунке 5.

После получения действительного PDU ведомого устройства цикл FSoE завершается.

Uh o t

FsoE

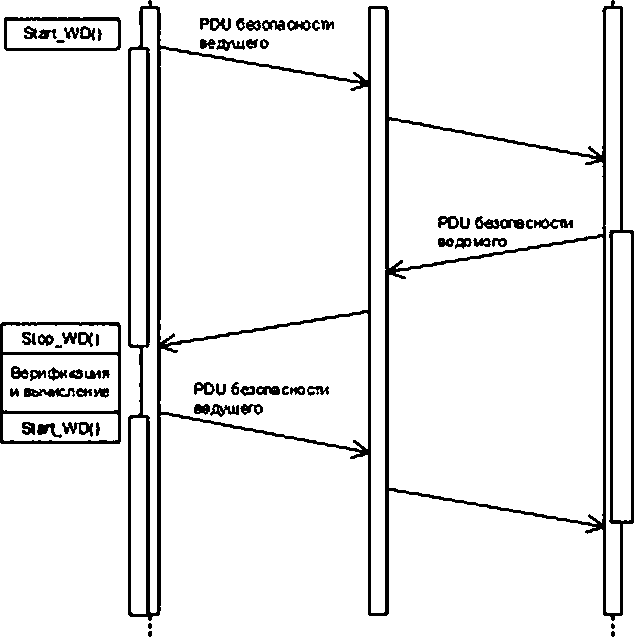


Рисунок 5 — Цжл FSoE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ведущий |  | Компонент |  | Ведомы\* |
| FSoE |  | сети |  | FSoE |

ВеБифмв1»1я

и вычисление SlartJMX)

Stcp.WOO

* 1. Услуги FSoE

Для каждого соединения FSoE ведущее устройство FSoE должно поддерживать обработчик ведущего устройства FSoE для управления связанным с ним ведомым устройством FSoE.

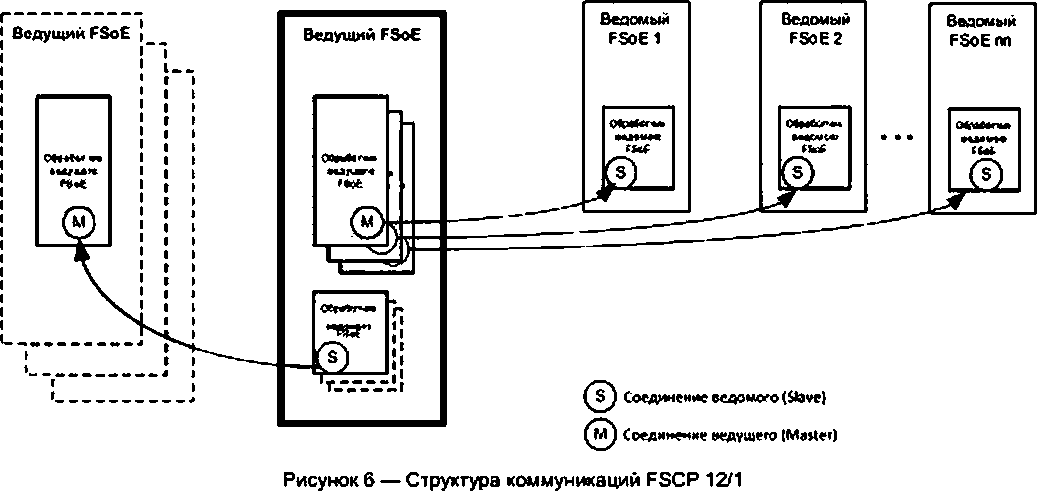
Для коммуникаций ведущего устройства FSoE с ведущим устройством FSoE ведущее устройство

FSoE должно поддерживать один или несколько обработчиков ведомого устройства FSoE. На рисунке 6 показан возможный функционал FSoE для ведущих и ведомых устройств FSoE.

11

#### ГОСТ Р МЭК 61764.3-12—2016

Коммуникации



## Протокол коммуникационного уровня безопасности

* 1. Формат PDU безопасности
     1. Структура PDU безопасности

На рисунке 7 показана структура одного PDU безопасности, встроенного в POU Типа 12. В табли­ це 3 представлена общая структура PDU безопасности.

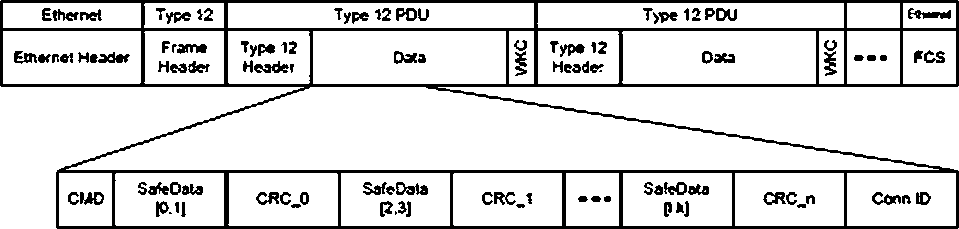


Рисунок 7 — PDU безопасности для CPF 12. встроенное 8 PD4J Типа 12

PDU безопасности циклически передается через подчиненные полевые шины. Каждый узел FSCP 12/1 обнаруживает новый PDU безопасности, если хотя бы один бит в PDU безопасности был изменен.

PDU безопасности обладает переменной длиной, установленной в описании ведомого устройства

FSoE.

Длина данных безопасности может составлять 1 октет или же четное число октет. Длина данных

безопасности может отличаться в зависимости от направления (ввод или вывод).

Более короткая из двух длин данных безопасности в PDU ведущего устройства безопасности и PDU ведомого устройства безопасности определяет, как много данных безопасности используется во время фазы инициализации соединения FSoE с информацией о параметрах. Для более длинного из двух PDU блоков данных безопасности назначается значение ноль.

12

#### ГОСТ Р МЭК 61784-3.12—2016

Таблица 3 — Общий PDU безопасности

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 0 | Command | Команда |
| 1 | SafeOata(0] | данные безопасности, октет 0 |
| 2 | SafeData(1] | данные безопасности, октет 1 |
| 3 | CRCO.Lo | младший октет (биты 0-7) 16-биговый CRC\_0 |
| 4 | CRCO.Hi | старший октет (биты 8-15) 16-битовый CRC\_0 |
| 5 | SafeData(2] | данные безопасности, октет 2 |
| 6 | SafeData[3] | данные безопасности, октет 3 |
| 7 | CRC\_1\_Lo | младший октет (биты 0-7) 16-биговый CRC\_1 |
| 8 | CRC\_1\_Hi | старший октет (биты 8-15) 16-битовый CRC\_1 |
| ... |  |  |
| (о-1) \*2-1 | SafeData|n-2] | данные безопасности, октет п-2 |
| (о-1) «2 | SafeData(n-l) | данные безопасности, октет п-1 |
| (о-1) «2+1 | CRC\_{n-2y2\_Lo | младший октет (биты 0-7) 16-биговый CRC\_(n-2)/2 |
| (о-1) «2+2 | CRC\_(n-2y2\_Hi | старший октет (биты 8-15) 16-битовый CRC\_(n-2)/2 |
| (о-1) «2+3 | Conn\_ld\_Lo | уникальный Id соединения, младший октет |
| (о-1) «2+4 | Conn\_ld\_Hi | уникальный Id соединения, старший октет |

PDU безопасности может передавать *п* октетов данных безопасности. Два октета данных пере- даются с помощью 2<октетного CRC.

Короткий PDU безопасности состоит из 6 октетов, которые могут испопьэоваться для передачи 1 октета данных безопасности, как зто показано в таблице 4.

Таблица 4 — Короткий POU безопасности

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 0 | Command | Команда |
| 1 | SafeDala(0] | данные безопасности, октет 0 |
| 2 | CRC\_0\_Lo | младший октет (биты 0-7) 16-биговый CRC\_0 |
| 3 | CRC.O.Hi | старший октет (биты 8-15) 16-битовый CRC\_0 |
| 4 | Conn\_kJ\_Lo | уникатъный Id соединения, младший октет |
| 5 | Conn\_W\_Hi | уникальный Id соединения, старший октет |

* + 1. Команда PDU безопасности

Команда PDU безопасности определяет значение данных безопасности, основываясь на схеме, показанной в таблице 5.

Таблица 5 — Команда POU безопасности

|  |  |
| --- | --- |
| Команда | Описание |
| 0x36 | ProcessData (ДанныеПроцесса) |
| 0х2А | Reset (Перезапуск) |
| 0х4Е | Session (Сеанс) |
| 0x64 | Connection (Соединение) |
| 0x52 | Parameter (Параметр) |
| 0x08 | FailSafeData (ОткаэоусгойчивыеДанные) |

13

#### ГОСТ Р МЭК 61784-3-12—2016

* + 1. CRC блока PDU безопасности
       1. Вычисление CRC

Два октета данных безопасности передаются с помощью соответствующих двух октетов CRC.

Кроме передаваемых данных (command, data. ConnID), CRC\_0 блока PDU безопасности также включает виртуальный порядковый номер. CRC\_0 последнего полученного PDU безопасности и три дополнительных нулевых октета. Если передаются только данные безопасности одного октета, то Safe\* Data[1) не учитывается в вычислении.

CRC\_0 :• f{received CRC\_0, ConnID, Sequence\_Number, command, SafeData(0],SafeData[l], 0x**0 0 0** )~

В таблице 6 показана последовательность вычислений CRC\_0.

Таблица 6 — Последовательность вычислений CRC\_0

|  |  |
| --- | --- |
| Шаг | Аргумент |
| 1 | полученный CRC\_0 (бит 0-7) |
| 2 | полученный CRC\_0 (бит 8-15) |
| 3 | ConnID (бит 0-7) |
| 4 | ConnID (бит 8-15) |
| 5 | Sequence\_Number (бит 0-7) |
| 6 | Sequence\_Number (бит 8-15) |
| 7 | Command |
| 8 | SafeData|0| |
| 9 | SafeData[1] |
| 10 | 0 |
| 11 | 0 |
| 12 | 0 |

CRC\_i (0 < i *<-* ((п-2)/2}) блока PDU безопасности также включает индекс CRC —i.

CRC\_i Г (received CRC\_0, ConnID, Sequence\_Nuober, cootnand, 1, Sa£eData(i \* 2}, SafeData(i \* 2 \* 1), 0)

В таблице 7 показана последовательность вычислений CRCJ.

Таблица 7 — Последовательность вычислений CRC\_i (i>0>

|  |  |
| --- | --- |
| Шаг | Аргумент |
| 1 | полученный CRC\_0 (бит 0-7) |
| 2 | полученный CRC\_0 (бит 8-15} |
| 3 | ConnID (бит 0-7) |
| 4 | ConnID (бит 8-15) |
| 5 | Sequence\_Number (бит 0-7) |
| 6 | Seqoence\_Number (бит 8-15) |
| 7 | Command |
| 8 | Индекс i (bit 0-7} |
| 9 | Индекс i (bit 8-15) |
| 10 | SafeDala|0| |
| 11 | SafeData(1] |
| 12 | 0 |
| 13 | 0 |
| 14 | 0 |

14

#### ГОСТ Р МЭК 61784-3-12—2016

* + - 1. Выбор полинома CRC

Полином 0x13987 используется для вычисления сигнатур CRC и называется полиномом безопасности.

Для того чтобы разрешить передачу PDU безопасности по черному каналу, чьи характеристики

передачи не связаны с безопасностью, при определении вероятности возникновения остаточной ошибки должна испоьзоеаться частота битовых сбоев, равная 10'2. Вероятность возникновения оста\* точной ошибки не должна превышать 10~9.

Безопасность обеспечивается за счет того, что ведущее устройство FSoE и ведомое устройство FSoE переходят в состояние сброса (т. е. безопасное состояние), как только обнаруживается ошибка.

Все параметры вычисления CRC за исключением данных безопасности обладают фиксированным

ожидаемым значением, чтобы в вычислении вероятности остаточной ошибки учитывались только данные безопасности.

Математическое доказательство, демонстрирующее, что вероятность возникновения остаточной

ошибки в случае полинома безопасности для 16-битовых данных безопасности и частоты битовых сбоев, равной 10~2. не превышает 1СГ9. включено в отдельный документ, где представлена полная количественная оценка.

* + - 1. Наследование CRC

Включение (наследование) CRCJ последней полученной телеграммы в вычисление CRC гарантирует, что два последовательных PDU блока ведущего устройства безопасности или PDU ведомого устройства безопасности отличаются друг от друга, даже если другие данные не претерпели изменений.

Наследование CRC\_0 также гарантирует безопасную и постоянную передачу данных, распростра­ няемых несколькими PDU блоками Типа 12 по причине их длины.

CRC\_0 полученного PDU безопасности включено в вычисление всех CRC J для PDU безопасности, которое будет отправлено.

1. таблице 8 показан пример для наследования CRCJ.

Таблица 8 — Пример наследования CRC\_0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Цикл FSoE | ведущее устройство FSoE | | ведомое устройство FSoE | |
| старый CRC\_0 | новый CRCJ | старый CRCJ | новый CRCJ |
| И | CRCJ) (2 \* j - 3) | CRC.O (2 x j - *2)* | CRCJ (2\*j-2) | CRCJ (2 \* j - 1) |
| j | CRCJ) (2 \* j - 1) | CRCJ (2 \* j) | CRCJ (2 \* j) | CRCJ (2 \* j \* 1) |
| J+1 | CRCJ) (2 \* j + 1) | CRCJ (2 \* j + 2) | CRCJ (2 \* j + 2) | CRCJ (2 \* | + 3) |

В цикле FSoE j ведущее устройство FSoE получает PDU ведомого устройства безопасности с CRCJ (2 \* j - 1). Так как значение CRC\_0 (2 \* j - 2). которое было включено в вычисление CRC\_0 (2 х j - 1). было вычислено ведущим устройством FSoE в FSoE цикле (j -1), ведущее устройство FSoE может проверить CRC\_0 (2 \* j - 1) в PDU ведомого устройства безопасности.

В свою очередь, в FSoE цикле j ведомое устройство FSoE получает PDU ведущего устройства без­ опасности с CRC\_0 (2 х j) и также способно проверять этот PDU. так как CRC\_0 (2 \* j -1) вычисляется ведомым устройством FSoE в FSoE цикле (j - 1).

* + - 1. Порядковый номер

8 таблице 8 CRC\_0 (2 \* j) может быть равен CRCJ) {2 \* j - 2}. В случае коротких PDU блоков без­ опасности это может привести к тому, что PDU ведущего устройства безопасности в FSoE цикле (j - 1) будет таким же. как и PDU ведущего устройства безопасности в FSoE цикле j, в результате чего ведомое устройство FSoE не признает PDU ведущего устройства безопасности как новый PDU в FSoE цикле j и сработает сторожевой таймер FSoE.

CRC коды PDU ведущего устройства безопасности, тем самым, включают виртуальный 16-битовый порядковый номер ведущего устройства, который ведущее устройство FSoE увеличивает с каждым новым PDU ведущего устройства безопасности. CRC код PDU ведомого устройства безопасности также включает виртуальный 16-битовый порядковый номер ведомого устройства, увеличиваемый ведомым устройством FSoE с каждым новым PDU ведомого устройства безопасности.

Если CRC\_0 (2 \* j) равен CRCJ) (2 \* j - 2) несмотря на эти меры, то порядковый номер ведущего устройства увеличивается дальше до тех пор, пока CRC\_0 (2 \* j) не станет равным CRC\_0 (2 \* j - 2).

15

#### ГОСТ Р МЭК 61784-3-12—2016

Такой алгоритм используется как для генерации ведущим устройством FSoE блока PDU ведущею устройства безопасности, так и для тою, чтобы ведомое устройство FSoE могло проверить PDU ведущего устройства безопасности.

Если CRC\_0 (2 \* j + 1) равно CRC\_0 (2 \* j \* 1), то порядковый номер ведомого устройства увеличивается дальше до тех лор. пока CRC\_0 (2 \* j + 1) не станет равен CRCJD (2 \* j • 1). Такой алгоритм используется как для генерации ведомым устройством FSoE блока PDU ведомою устройства безопасности, так и для того, чтобы ведущее устройство FSoE могло проверить PDU ведомого устройства безопасности.

Порядковый номер может принимать значения от 1 до 65 535. После 65 535 последовательность запускается снова, начиная с 1. т. е. О не учитывается.

* + - 1. Индекс CRC

Если передается более двух октетов данных безопасности и. тем самым. 2 или несколько кодов CRC (например CRC\_0 и CRC\_1), то мер. описанных выше, недостаточно для обнаружения всех возможностей для реверсирования в рамках POU безопасности, см. пример в таблице 9.

Таблица 9 — Пример для 4 октетов данных безопасности с заменой октетов 1 -4 на октеты 5-8.

|  |  |  |
| --- | --- | --- |
| Октет | Намание | Описание |
| 0 | Command | Команда |
| 1 | SareData(2] | данные безопасности, октет 2 |
| 2 | SafeData[3) | данные безопасности, октет 3 |
| 3 | CRC\_1\_Lo | младший октет (биты 0-7) 16-битовый CRC\_1 |
| 4 | CRC\_1\_H» | старший октет (биты 8-15) 16-битовый CRC\_1 |
| 5 | SafeData(O) | данные безопасности, октет 0 |
| 6 | SafeData(1J | данные безопасности, октет 1 |
| 7 | CRC\_0\_Lo | младший октет (биты 0-7) 16-битовый CRC\_0 |
| 8 | CRC\_0\_Hi | старший октет (биты 8-15) 16-битовый CRC\_0 |
| 9 | Conn\_ld\_Lo | младший октет (биты 0-7) уникального id соединения |
| 10 | Conn\_ld\_Ht | старший октет (биты 8-15} уникального Id соединения |

Индекс i (двухоктетное значение), тем самым, также включается в соответствующий CRC\_i.

Это позволяет обнаруживать реверсирование октетов 1\*4 и 5-8.

* + - 1. Дополнительные нулевые октеты

Вероятность возникновения остаточной ошибки вычисляется через соотношение обнаруженных и необнаруженных ошибок. Необнаруженные ошибки, по сути, являются ошибками, которые уже были обнаружены полиномом CRC для черного канала (стандартный полином), так как эти ошибки не очевидны на уровне безопасности, будучи отфильтрованными заранее. Наихудшее соотношение между обнаруженными ошибками (ошибками, не обнаруженными стандартным полиномом, но обнаруженные полиномом CRC уровня безопасности) и необнаруженными ошибками (ошибками, уже обнаруженными стандартным полиномом) возникает, если стандартный полином делится на полином безопасности без остатка.

В таком случае, для обеспечения надлежащей независимости двух полиномов друг от друга, в вычисление включаются три нулевых октета.

* + - 1. ID сеанса

Неисправное устройство, хранящее телеграммы (например, коммутатор), в особенности в случае полевых шин. передаваемых посредством Ethernet, может привести к тому, что правильная последовательность телеграмм вносится в неправильное время. Наследование CRC означает, что последовательность PDU безопасности всегда полагается на историю.

Передача произвольно генерируемою ID сеанса во время начальной настройки соединения FSoE гарантирует, что две последовательности PDU безопасности отличаются друг от друга после включения питания.

ID сеанса может принимать значения от 0 до 65 535.

16

#### ГОСТ Р МЭК 61784-3.12—2016

* 1. Процедура коммуникаций FSCP 12/1
     1. Цикл сообщения

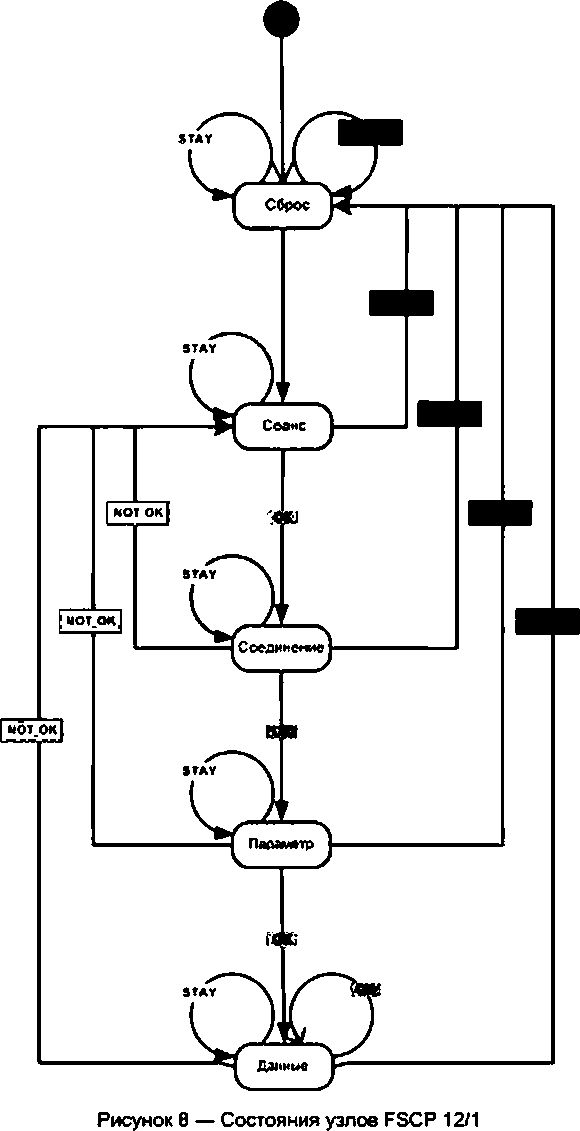
Коммуникации FSCP 12/1 функционируют а рамках признанного цикла сообщения (FSoE цикл), т. е. ведущее устройство FSoE отправляет PDU ведущего устройства безопасности ведомому устройству FSoE и ожидает получить в ответ POU безопасности. И только после этого генерируется следующий PDU ведущего устройства безопасности.

* + 1. Состояния узлов FSCP 12/1

7.2.2.1 Общие положения

После установления соединения FSoE узлы FSCP 12/1 переходят в разные состояния перед тем. как данные безопасности становятся подтвержденными и состояние безопасности покидается.

На рисунке 8 показаны состояния узлов FSoE.



17

#### ГОСТ Р МЭК 61784-3-12—2016

После включения питания или ошибки коммуникаций ведущее и ведомое устройство FSoE находятся в состоянии сброса. Узлы FSoE также переключаются в reset-state (состояние сброса), если они обнаруживают ошибку в коммуникациях или локальном приложении. После выполнения команды FSoE Reset (сброс FSoE). поступившей от ведомого устройства FSoE. ведущее устройство FSoE переключается в состояние сеанса (переходы обозначены оранжевым цветом). После выполнения команды сброса, поступившей от ведущего устройства FSoE. ведомое устройство FSoE переключается е состояние сброса. Затем может быть принято состояние данных через состояния сеанса, соединения и параметров. Выход из состояния безопасного вывода может быть осуществлен только в состоянии данных.

*1.2.22* Состояние сброса

Состояние сброса используется для повторной инициализации соединения FSoE после включения питания или возникновения ошибки коммуникаций FSoE. Ведущее устройство FSoE выходит из состояния сброса, когда оно отправляет PDU ведущего устройства безопасности с командой Session (сеанс) ведомому устройству FSoE. Ведомое устройство FSoE выходит из состояния сброса после того, как оно получает подтвержденный PDU ведущего устройства безопасности вместе с командной Ses­ sion.

В состоянии сброса порядковый номер и CRC последней телеграммы, используемые в вычислении CRC. сбрасываются.

В таблице 10 показан пример PDU ведущего устройства безопасности для четырех октетов данных

безопасности вместе с командой сброса.

Т а б л и ц а 10 — PDU ведущего устройства безопасности для четырех октетов данных безопасности с command = Reset после сброса (сброса соединения, т. е. перезапуска) или ошибки

|  |  |  |
| --- | --- | --- |
| Октет | Немение | Описание |
| 0 | Command | Reset (сброс) |
| 1 | SafeData(O) | кед ошибки (бит 0-7). 0 для перезапуска |
| 2 | SafeData[1] | не используется (=0) |
| 3 | CRC\_0\_Lo | младший октет (биты 0-7) 16-битовый CRC\_0 |
| 4 | CRC\_0\_H« | старший октет (биты 8-15) 16-битовый CRC\_0 |
| 5 | SafeData|2] | не используется (=0) |
| б | SafeData[3] | не используется (=0) |
| 7 | CRC\_1\_Lo | младший октет (биты 0-7) 16-битовый CRC\_1 |
| 8 | CRC\_1\_Hi | старший октет (биты 8-15) 16-битовый CRC\_1 |
| 9 | Conn\_ld\_Lo | кв используется (И)) |
| 10 | Conn\_ld\_Hi | кв используется (И)) |

Ведомое устройство FSoE подтверждает команду Reset, устанавливая SafeData в значение 0.

1. таблице 11 показан пример PDU ведомого устройства безопасности для четырех октетов данных безопасности вместе с командой сброса.

Таблица 11 — PDU ведомого устройства безопасности для четырех октетов данных безопасности вместе с командой сброса

|  |  |  |
| --- | --- | --- |
| Октет | Немение | Описание |
| 0 | Command | *Reset* (сброс) |
| 1 | SafeData(O) | 0 |
| 2 | SafeData(1j | 0 |
| 3 | CRC\_0\_Lo | младший октет (биты 0-7) 16-битовый CRC\_0 |
| *4* | CRC\_0\_Hi | старший октет (биты 8-15) 16-битоаый CRC\_0 |
| 5 | SafeData(2] | не используется (=0) |

18

#### ГОСТ Р МЭК 61784-3.12—2016

*Окончание таблицы 11*

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 6 | SafeData(3| | не используется (=0) |
| 7 | CRC\_1\_Lo | младший октет (биты 0-7) 16-битовый CRC\_1 |
| 6 | CRC\_1\_Hi | старший октет (биты 6-15) 16-битовый CRC\_1 |
| 9 | Conn\_kJ\_Lo | не используется (=0) |
| 10 | ConnJd.Hi | не используется (=0) |

Ведомое устройство FSoE также отправляет PDU безопасности с командой Reset во время перезапуска (сброс соединения) или в случае возникновения ошибки. Это показано в таблице 12. как пример для четырех октетов данных безопасности.

Т а б л и ц а 12 — PDU ведомого устройства безопасности для четырех октетов данных безопасности с command = Reset после перезапуска (сброс соединения) или ошибки

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 0 | Command | *Reset* (сброс) |
| 1 | SafeData(O) | кед ошибки (бит 0-7), 0 для сброса |
| 2 | SafeDala(1] | не используется (=0) |
| 3 | CRC.O.Lo | младший октет (биты 0-7) 16-битовый CRC\_0 |
| 4 | CRC.O.Hi | старший октет (биты 8-15) 16-битовый CRC\_0 |
| 5 | SafeDala(2) | не используется (=0) |
| 6 | SafeData(3] | не используется (=0) |
| 7 | CRC\_1\_Lo | младший октет (биты 0-7) 16-битовый CRC\_1 |
| 8 | CRCJ.Hi | старший октет (биты 8-15) 16-битовый CRC\_1 |
| 9 | Coon\_M\_Lo | не используется (=0) |
| 10 | ConnJd.Hi | не используется (=0) |

Ведущее устройство FSoE подтверждает команду Reset, отправляя PDU ведущего устройства безопасности с командой Session.

* + - 1. Состояние сеанса

Во время перехода в состояние сеанса или при нахождении в нем. 16'6итовый ID сеанса ведущего устройства передается от ведущего устройства FSoE ведомому устройству FSoE. которое в ответ воз\* вращает свой собственный Ю сеанса ведомого устройства.

Оба узла FSoE генерируют ID сеанса как произвольный номер, используемый для различия множественных последовательностей PDU безопасности в случае нескольких перезапусков соединения FSoE.

8 таблице 13 показан пример PDU ведущего устройства безопасности для четырех октетов данных безопасности с командой Session.

Т а б л и ц а 13 — PDU ведущего устройства безопасности для четырех октетов данных безопасности с command = Session

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 0 | Command | Session (сеанс) |
| 1 | SafeData(0] | *td* Сеанса ведущего устройства, октет 0 |
| 2 | SafeDala(1] | *td* Сеанса ведущего устройства, октет 1 |

19

#### ГОСТ Р МЭК 61784-3-12—2016

*Окончание таблицы 13*

|  |  |  |
| --- | --- | --- |
| Октет | Немение | Описание |
| 3 | CRC\_0\_Lo | младший октег (биты 0-7) 16-битовый CRC\_0 |
| 4 | CRC\_0\_Hi | старший октет (биты 8-15) 16-битовый CRC\_0 |
| 5 | SafeData|2] | не используется (=0) |
| б | SafeData[3) | не используется (И)) |
| 7 | CRC\_1\_lo | младший октет (биты 0-7) 16-битовый CRC\_1 |
| 3 | CRC\_1\_Hi | старший октет (биты 8-15} 16-биговый CRC\_1 |
| 9 | Conn\_ld\_Lo | не используется (И)) |
| 10 | Conn\_ld\_Hi | не используется (=0) |

Ведомое устройство FSoE подтверждает команду Session, отправляя назад ID сеанса ведомого устройства. В таблице 14 показан пример PDU ведомого устройства безопасности для четырех октетов SafeData (БезопасныеДанныв) с командой Session.

Таблица 14 — PDU ведомого устройства безопасности для четырех октетов SafeData (БезопэсныеДанные) с command = Session

|  |  |  |
| --- | --- | --- |
| Октет | Немение | Описание |
| 0 | Command | *Session* (сеанс) |
| 1 | SafeData(0] | *Id* Сеанса ведомого устройства, октет 0 |
| 2 | SafeData[1] | *Id* Сеанса ведомого устройства, октет 1 |
| 3 | CRC\_0\_Lo | младший октег (биты 0-7) 16-битовой CRC\_0 |
| 4 | CRC\_0\_Hi | старший октет (биты 8-15) 16-битовой CRC\_0 |
| 5 | SafeData(2] | не используется (=0) |
| 6 | SafeData[3) | не используется (=0) |
| 7 | CRC\_1\_Lo | младший октег (биты 0-7) 16-битовой CRC\_1 |
| 8 | CRC\_1\_Hi | старший октет (биты 8-15) 16-битовой CRC\_1 |
| 9 | Conn\_ld\_Lo | не используется (И)) |
| 10 | Conn\_ld\_Hi | не используется (И)) |

Если PDU безопасности содержит хотя бы 2 октета данных безопасности. Ю сеанса может быть передан с одним FSoE циклом. Если, с другой стороны. PDU безопасности содержит только 1 октет данных безопасности, то ID сеанса должен быть передан с двумя FSoE циклами.

Значение ID сеанса не имеет никакого значения для безопасности, т. е. коммутатор в узле FSoE, получающем PDU безопасности, не нуждается в анализе проблем безопасности. Таким образом. ID соединения для команды Session устанавливается в значение 0.

Ведущее устройство FSoE выходит из состояния сеанса после того, как оно передало полный ID сеанса и получило связанные с ним подтверждения от ведомого устройства FSoE. отправив PDU безопасности с командой Connection (соединение) ведомому устройству FSoE. Ведомое устройство FSoE выходит из состояния сеанса после того, как оно получает PDU безопасности с командой Connec­ tion от ведущего устройства FSoE.

И ведущее, и ведомое устройства FSoE также выйдут из состояния сеанса, если они обнаружат ошибку коммуникаций FSoE.

В ведущем устройстве FSoE после получения команды RESET и порядковый номер, и CRC

последней телеграммы, используемые в вычислении CRC. сбрасываются.

* + - 1. Состояние соединения

В состоянии соединения 16-битовый ID соединения передается от ведущего устройства FSoE ведомому устройству FSoE. ID соединения должен быть уникальным и сгенерированным конфигуратором ведущего устройства FSoE. Если в коммуникационной системе присутствует несколько

20

#### ГОСТ Р МЭК 61784-3.12—2016

ведущих устройств FSoE. то пользователь должен убедиться, что используемые идентификаторы ID соединения уникальны.

8 дополнение к 16-битовому ID соединения передается также уникальный адрес ведомого устройства FSoE. В таблице 15 показано содержание данных безопасности, передаваемых в состоя, нии соединения.

Таблица 15 — Данные безопасности, передаваемые в состоянии соединения

|  |  |
| --- | --- |
| Оствт данных безопасности | Описание |
| 0 | мпедший октет (биты 0\*7) ID соединения |
| 1 | старший октет (биты 6\*15) ID соединения |
| 2 | младший октет (биты 0\*7) Адреса ведомого устройства FSoE |
| 3 | старший октет (биты 8-15) Адреса ведомого устройства FSoE |

В зависимости от длины данных безопасности, требуется до четырех FSoE цикла. Для PDU без- опасности с четырьмя октетами данных безопасности требуется только один FSoE цикл, что показано в таблицах 16 и 17.

Таблица 16 — PDU ведущего устройства безопасности для четырех октетов данных безопасности в состоянии соединения

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 0 | Command | *Connection* (соединение) |
| 1 | SafeDatafOJ | Id Соединения, младший октет |
| 2 | SafeData(1] | Id Соединения, старший октет |
| 3 | CRC.O.Lo | младший октет (биты 0-7) 16-битовой CRC\_0 |
| 4 | CRC\_0\_Hi | старший октет (биты 6-15) 16-битовой CRC\_0 |
| 5 | SafeData[2| | Адреса ведомого устройства FSoE. младший октет |
| 6 | SafeData[3] | Адреса ведомого устройства FSoE. старший октет |
| 7 | CRC\_1\_Lo | младший октет (биты 0-7) 16-битоеой CRC\_1 |
| 8 | CRC\_1\_Hi | старший октет (биты 6-15) 16-битовой CRC\_1 |
| 9 | Conn\_kJ\_Lo | Id соединения, младший октет |
| 10 | ConnJd.Hi | Id соединения, старший октет |

Ведомое устройство FSoE подтверждает команду Connection отправкой назад данных безопасности.

Таблица 17 — PDU ведомого устройства безопасности для четырех октетов данных безопасности в состоянии соединения.

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 0 | Command | Connection (соединение) |
| 1 | SafeOata(0] | Id соединения, младший октет |
| 2 | SafeData(1] | Id соединения, старший октет |
| 3 | CRC.O.Lo | младший октет (биты 0-7) 16-битовой CRC\_0 |
| 4 | CRC.O.Hi | старший октет (биты 8-15) 16-битовой CRC\_0 |
| 5 | SafeDala(2] | Адреса ведомого устройства FSoE. младший октет |
| 6 | SafeData(3| | Адреса ведомого устройства FSoE. старший октет |
| 7 | CRC\_1\_Lo | младший октет (биты 0-7) 16-битоеой CRC\_1 |
| в | CRC\_1\_Hi | старший октет (биты 6-15) 16-битовой CRC\_1 |
| 9 | Conn\_W\_Lo | Id соединения, младший октет |
| 10 | Conn\_ld\_Hi | Id соединения, старший октет |

21

#### ГОСТ Р МЭК 61784-3-12—2016

Адрес ведомого устройства FSoE должен быть уникальным в рамках коммуникационной системы. Он может быть установлен на соответствующем ведомом устройстве FSoE. Передавая адрес ведомого устройства FSoE вместе с ID соединения, ведомое устройство FSoE может проверять, было ли оно в действительности адресовано, для обнаружения недействительной адресации. Так как ID соединения также является уникальным в рамках коммуникационной системы. ID соединения всегда отправляет в последующих PDU блоках безопасности для того, чтобы и ведущее, и ведомое устройство FSoE могло обнаружить, адресуются ли им телеграммы. Тем самым, уникальный ID соединения включает 65 535 соединений FSoE для их реализации в коммуникационной системе (ID соединения = 0 не допускается).

* + - 1. Параметрическое состояние

В параметрическом состоянии осуществляется передача параметров коммуникаций, связанных с безопасностью, и приложений, связанных с безопасностью и зависящих от устройства. Приложения могут иметь произвольную длину. Наследование CRC гарантирует безопасность и постоянство передачи параметров.

В таблице 18 показано содержание данных безопасности, передаваемых в параметрическом со\* стоянии.

Таблица 18 — Данные безопасности, передаваемые в параметрическом состоянии

|  |  |
| --- | --- |
| Октет денных бееопес\* нести | Описание |
| 0 | младший октет (биты 0-7) длина параметров коммуникации в октетах (=2) |
| 1 | старший октет (биты 8\*15) длина параметров коммуникации в октетах (=0) |
| 2 | младший октет (биты 0-7) сторожевого таймера FSoE (в мс> |
| 3 | старший октет {биты 8-15) сторожевого таймера FSoE (в мс) |
| *4* | младший октет (биты 0-7) длина параметров приложения в октетах |
| 5 | старший октет {биты 8-15) длина параметров приложения в октетах |
| 6 | 1-й октет параметра приложения, связанного с безопасностью |
|  |  |
| о + 5 | л-й октет параметра приложения, связанного с безопасностью |

Число циклов FSoE в параметрическом состоянии зависит от длины параметров приложения, связанного с безопасностью, и длины данных безопасности в PDU безопасности. Если не все октеты данных безопасности требуются в последнем FSoE цикле, то они должны передаваться как 0.

Для PDU безопасности с четырьмя октетами данных безопасности и двумя октетами параметров приложения, связанных с безопасностью, требуется два FSoE цикла; зто показано в таблицах 19—22.

Таблица 19 — Первый PDU ведущего устройства безопасности для четырех октетов данных безопасности в параметрическом состоянии

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 0 | Command | *Parameter* (параметр) |
| 1 | SafeData[0] | младший октет (биты 0-7) дгына параметров коммуникации в октетах (=2) |
| 2 | SafeData[1] | старший октет (биты 8-15) длина параметров коммуникации в октетах (И)) |
| 3 | CRC.O.Lo | младший октет (биты 0-7) 16-битовой CRC\_0 |
| *4* | CRCO.Hi | старший октет (биты 8-15) 16-битовой CRC\_0 |
| 5 | SafeData|2] | младший октет (биты 0-7) сторожевого таймера FSoE (в мс) |
| 6 | SafeData[3] | старший октет (биты 8-15) сторожевого таймера FSoE (в мс) |
| 7 | CRC\_1\_Lo | младший октет (биты 0-7) 16-битовой CRC\_1 |
| 8 | CRC\_1\_Hi | старший октет (биты 8-15) 16-битовой CRC\_1 |
| 9 | Conn\_ld\_Lo | М соединения, младший октет |
| 10 | Conn\_ld\_Hi | hJ соединения, старший октет |

22

#### ГОСТ Р МЭК 61784-3.12—2016

Ведомое устройство FSoE подтверждает корректную команду Parameter возвращением данных безопасности.

Таблица 20 — Первый PDU ведомого устройства безопасности для четырех октетов данных безопасности в параметрическом состоянии

|  |  |  |
| --- | --- | --- |
| Октет | Немение | Описание |
| 0 | Command | *Parameter (параметр)* |
| 1 | SafeData(O) | младший октет (биты 0-7) длина параметров коммуникации в октетах (=2) |
| 2 | SafeOata[1] | старший октет {биты 8\*15) длина параметров коммуникации в октетах (=0) |
| 3 | CRC\_0\_Lo | младший октет (биты 0-7) 16-битовой CRC\_0 |
| 4 | CRC\_0\_Hi | старший октет (биты 8-15) 16-битовой CRC\_0 |
| 5 | SafeData[2] | младший октет (биты 0-7) сторожевого таймера FSoE (а мс) |
| 6 | SafeDa!a[3) | старший октет (биты 8-15) сторожевого таймера FSoE (в мс) |
| 7 | CRC\_t\_Lo | младший октет (биты 0-7) 16-битоеой CRC\_1 |
| 8 | CRC\_1\_Hi | старший октет (биты 8-15) 16-битоеой CRC\_1 |
| 9 | Conn\_ld\_Lo | Id соединения, младший октет |
| 10 | Conn\_ld\_Hi | Id соединения, старший октет |

Ведущее устройство FSoE отправляет второй PDU ведущего устройства безопасности после того, как он корректно принял первый PDU ведомого устройства безопасности.

Таблица 21 — Второй POU ведущего устройства безопасности для четырех октетов данных безопасности в параметрическом состоянии

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 0 | Command | *Parameter (параметр)* |
| 1 | SafeOala(0] | младший октет (биты 0-7) длина параметров приложения в октетах (=2) |
| 2 | SafeData[1) | старший октет (биты 8-15) длина параметров приложения в октетах (=0) |
| 3 | CRC\_0\_Lo | младший октет (биты 0-7) 16-битоеой CRC\_0 |
| 4 | CRC\_0\_Hi | старший октет (биты 8-15) 16-битоеой CRC\_0 |
| 5 | SafeData[2) | 1-й октет параметра приложения, связанного с безопасностью |
| 6 | SafeData(3) | 2-й октет параметра приложения, связанного с безопасностью |
| 7 | CRC\_1\_Lo | младший октет (биты 0-7) 16-битоеой CRC\_1 |
| 8 | CRC\_1\_Hi | старший октет (биты 8-15) 16-битовой CRC\_1 |
| 9 | Conn\_ld\_Lo | Id соединения, младший октет |
| 10 | Conn\_ld\_Hi | Id соединения, старший октет |

Ведомое устройство FSoE подтверждает корректную команду Parameter возвращением данных безопасности.

Таблица 22 — PDU ведомого устройства безопасности для четырех октетов данных безопасности в параме­ трическом состоянии

|  |  |  |
| --- | --- | --- |
| Остет | Название | Описание |
| 0 | Command | *Parameter (параметр)* |
| 1 | SafeData(0] | младший октет (биты 0-7) длина параметров приложения е октетах (=2) |
| 2 | SafeDataJI] | старший октет (биты 8-15) длина параметров приложения в октетах (=0) |
| 3 | CRC\_0\_Lo | младший октет (биты 0-7) 16-битоеой CRC\_0 |

23

#### ГОСТ Р МЭК 61784-3-12—2016

*Окончание таблицы 22*

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 4 | CRCO.Hi | старший октет (биты 8-15) 16-битовой CRC\_0 |
| 5 | SafeData[2| | 1 -й октет параметра приложения, связанного с безопасностью |
| 6 | SafeData[3) | 2-й октет параметра приложения, связанного с безопасностью |
| 7 | CRC\_1\_Lo | младший октет (биты 0-7) 16-битовой CRC\_1 |
| 8 | CRC\_1\_Hi | старший октет (биты 8-15) 16-битовой CRC\_1 |
| 9 | Conn\_ld\_Lo | Id соединения, младший октет |
| 10 | Conn\_ld\_Hi | Id соединения, старший октет |

Параметры сторожевого таймера FSoE и приложения, связанного с безопасностью, конфигуриру­ ются посредством конфигуратора безопасности ведущего устройства FSoE.

* + - 1. Состояние данных
         1. Действительные (подтвержденные) данные

В то время как в предыдущих состояниях число FSoE циклов было фиксированным, в состоянии данных FSoE циклы передаются до тех пор. пока не возникнет коммуникационная ошибка или пока узел FSoE не будет локально остановлен. Ведущее устройство FSoE отправляет безопасные выводы ведомому устройству FSoE.

В таблице 23 показан пример PDU ведущего устройства для четырех октетов SafeOutputs с командой ProcessData (данные процесса).

Таблица 23 — PDU ведущего устройства безопасности для четырех октетов ProcessData в состоянии данных.

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 0 | Command | *ProcessData* (данные процесса) |
| 1 | SafeData(O) | 1-й октет SafeOutputs |
| 2 | SafeData[1] | 2-й октет SafeOutputs |
| 3 | CRC\_0\_Lo | младший октет (биты 0-7) 16-битовой CRC\_0 |
| 4 | CRC\_0\_Hi | старший октет (биты 8-15) 16-битовой CRC\_0 |
| 5 | SafeData[2) | 3-й октет SafeOutputs |
| 6 | SafeData[3) | 4-й октет SafeOutputs |
| 7 | CRC\_1\_Lo | младший октет (биты 0-7) 16-битовой CRC\_1 |
| 8 | CRC\_1\_Hi | старший октет (биты 8-15) 16-биговой CRC\_1 |
| 9 | Conn\_ld\_Lo | id соединения, младший октет |
| 10 | Conn\_ld\_Hi | Id соединения, старший октет |

Ведомое устройство FSoE подтверждает PDU ведущего устройства безопасности и отправляет Safelnputs ведущему устройству FSoE.

В таблице 24 показан пример PDU ведомого устройства безопасности для четырех октетов Safeln­

puts с командой ProcessData.

Таблица 24 — PDU ведомого устройства безопасности для четырех октетов ProcessData в состоянии данных.

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 0 | Command | *ProcessData* (данные процесса) |
| 1 | SafeData(O) | 1-й октет Safelnputs |
| 2 | SafeData[1] | 2-й октет Safelnputs |
| 3 | CRC\_0\_Lo | младший октет (биты 0-7) 16-битовой CRC\_0 |

24

#### ГОСТ Р МЭК 61784-3.12—2016

*Окончание таблицы 24*

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 4 | CRC.O.Hi | старший октет (биты 8\*15) 16-битовой CRC\_0 |
| 5 | SafeData(2] | 3-й октет Safelnputs |
| 6 | SafeData(3] | 4-й октет Safelnputs |
| 7 | CRC\_1\_Lo | младший октет (биты 0-7) 16-битовой CRC\_1 |
| 8 | CRC\_1\_Hi | старший октет {биты 8-15) 16-битовой CRC\_1 |
| 9 | Conn\_kJ\_Lo | Id соединения, младший октет |
| 10 | ConnJd.Hi | Id соединения, старший октет |

* + - * 1. Команда FailSafeData

Если ведущее устройство FSoE локально обнаруживает, что безопасные выводы (SafeOutputs) не действительны или требуется перевести их в безопасное состояние, то оно отправляет команду FailSafeData.

8 таблице 25 показан пример PDU ведущего устройства безопасности для четырех октетов Fail. safeData с командой FailSafeData.

Таблица 25 — PDU ведущего устройства безопасности для четырех октетов отказоустойчивых данных в со\* стоянии дажых

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 0 | Command | *FailSafeData* |
| 1 | SafeData(O) | Отказоустойчивые данные = 0 |
| 2 | SafeData( 1] | Отказоустойчивые данные = 0 |
| 3 | CRC.O.Lo | младший октет (биты 0-7) 16-битовой CRC\_0 |
| 4 | CRC.O.Hi | старший октет (биты 8-15) 16-битовой CRC\_0 |
| 5 | SafeData(2] | Отказоустойчивые данные = 0 |
| 6 | SafeData[3] | Отказоустойчивые данные = 0 |
| 7 | CRC\_1\_Lo | младший октет (биты 0-7) 16-битовой CRC\_1 |
| 8 | CRC\_1\_Hi | старший октет (биты 8-15) 16-битовой CRC\_1 |
| 9 | Conn\_W\_Lo | Id соединения, младший октет |
| 10 | Conn\_W\_Hi | Id соединения, старший октет |

Если ведомое устройство FSoE локально обнаруживает, что безопасные вводы (Safelnputs) не действительны или требуется перевести их в безопасное состояние, то оно отправляет команду Fail\* SafeData.

8 таблице 26 показан пример PDU ведомого устройства безопасности для четырех октетов Fail\* safeData с командой FailSafeData.

Таблица 26 — PDU ведомого устройства безопасности для четырех октетов отказоустойчивых данных в со\* стоянии данных

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 0 | Command | *FailSafeData* |
| 1 | SafeData[0] | Отказоустойчивые данные = 0 |
| 2 | SafeData(1] | Отказоустойчивые данные = 0 |
| 3 | CRC\_0\_Lo | младший октет (биты 0-7) 16-битовой CRC\_0 |

25

#### ГОСТ Р МЭК 61784-3.12—2016

*Окончание таблицы 26*

|  |  |  |
| --- | --- | --- |
| Октет | Название | Описание |
| 4 | CRC\_0\_Hi | старший октет (биты 8-15) 16-биговой CRC\_0 |
| 5 | SafeData[2] | Отказоустойчивые данные = 0 |
| 6 | SafeData[3) | Отказоустойчивые данные = 0 |
| 7 | CRC\_1\_lo | младший октет (биты 0-7) 16-битовой CRC\_1 |
| 8 | CRC\_1\_Hi | старший октет (биты 8-15) 16-биговой CRC\_1 |
| 9 | Conn\_ld\_Lo | id соединения, младший октет |
| 10 | Conn\_ld\_Hi | Id соединения, старший октет |

Передача ProcessData или FailSafeOata не зависит от команды полученного PDU безопасности.

Она зависит только от локальных обстоятельств.

* 1. Реакция на ошибки коммуникаций

Узел FSoE может обнаруживать ошибки, перечисленные в таблице 27.

Таблица 27 — Коммуникационная ошибка FSoE

|  |  |
| --- | --- |
| Ошибка | Описание |
| Неожиданная команда | Полученная команда не допускается в данном состоянии |
| Неизвестная команда | Полученная команда не определена |
| ID неправильного соединения | ID соединения не соответствует ID соединения, переданному в состоянии соединения |
| Ошибка CRC | Хотя бы один из полученных CRC J не соответствует вычисленному CRC\_i |
| Сторожевой таймер истек | За время сторожевого таймера не было получено ни одного действитель­ ного PDU безопасности |
| Недействительиьы адрес ведомого устройства FSoE | Адрес ведомого устройства FSoE. переданный в состоянии соединения, не соответствует локальному адресу, установленному на ведомом устрой­ стве FSoE |
| Недействительные SafeOata | Данные безопасности, возвращенные ведомым устройством FSoE *в* со­ стояниях сеанса, соединения и параметров, не соответствуют ожидаемым значениям |
| Неисправные SafePara | SafePara. отправленные ведомому устройству FSoE в параметрическом состоянии, не двйствительтш |
| Недопустимая длина параметров коммуникаций | Неправильная длина параметра коммуникаций |
| Недопустимый коммуникационный параметр | Неправильное содержание параметра коммуникаций |
| Недопустимая длина параметра приложения | Неправильная длина параметра приложения |
| Недопустимый параметр приложения | Неправильное содержание параметра приложения |

Узел FSoE обнаруживает коммуникационную ошибку, отправляется команда Reset, а также свя­ занный с ней код ошибки в SafeData(O) для диагностических целей, ведущее устройство FSoE затем переключается в состоянии сеанса, ведомое устройство FSoE в состоянии сброса. Коды ошибок коммуникаций FSoE перечислены в таблице 28.

Таблица 28 — Коды ошибок коммуникаций FSoE

|  |  |
| --- | --- |
| Октет | Описание |
| 0 | Локальный сброс или подтверждение команды сброса |

26

#### ГОСТ Р МЭК 61784-3.12—2016

*Окончание таблицы 28*

|  |  |
| --- | --- |
| Октет | Описание |
| 1 | Неожиданная команда (INVALID\_CMD) |
| 2 | Неизвестная команда (UNKNOWN.CMD) |
| 3 | Недействительное соединение (INVAUD\_CONNlD) |
| 4 | Ошибка CRC (1NVAUD.CRC) |
| 5 | Сторожевой таймер истек (WD.EXPIREO) |
| 6 | Недействительный адрес ведомого устройства FSoE (INVAUD\_ADDRESS) |
| 7 | Недействительные данные безопасности (INVALID\_DATA) |
| 8 | Недопустимая длина параметра коммуникаций (INVALIO.COMMPARALEN) |
| 9 | Недействительные данные параметра коммуникаций (INVALiO.COMPARA) |
| 10 | Недопустимая длина параметра приложения (INVALIDJJSERPARALEN) |
| 11 | Недействительные данные параметра приложения (INVALID\_USERPARA) |
| 0x80-0xFF | Недействительные SafePara (зависит от устройства) |

* 1. Таблица состояний для ведущего устройства FSoE
     1. Машина состояний ведущего устройства FSoE
        1. Обзор

В зависимости коммуникационной процедуры ведущее устройство FSoE может находиться в со­ стояниях. перечисленных в таблице 29.

Таблица 29 — Состояния ведущего устройства FSoE

|  |  |
| --- | --- |
| Состояние | Описание |
| Сброс | Соединение FSoE сброшено (выводы в безопасном состоянии) |
| Сеанс | Передается 1D сеанса (выводы в безопасном состоянии) |
| Соединение | Передается Ю соединения (выводы в безопасном состоянии) |
| Параметры | Передаются параметры (вывозы в безопасном состоянии) |
| Данные | Передаются данные процесса или отказоустойчивые данные (выводы активны, только если получена команда *ProcessData)* |

Диаграмма состояний для ведущего устройства FSoE показана на рисунке 9.

В следующих секциях проводится анализ событий, которые могут произойти на ведущем устройстве FSoE для каждого состояния.

Каждое событие рассматривается в условиях различных действий или проистекающих состояний.

* + - 1. События

Событие может включать в себя различные параметры, на которые приводятся ссылки в таблицах состояний. В таблице 30 приведен список используемых событий.

Таблица 30 — События в таблице оостояний ведущего устройства FSoE

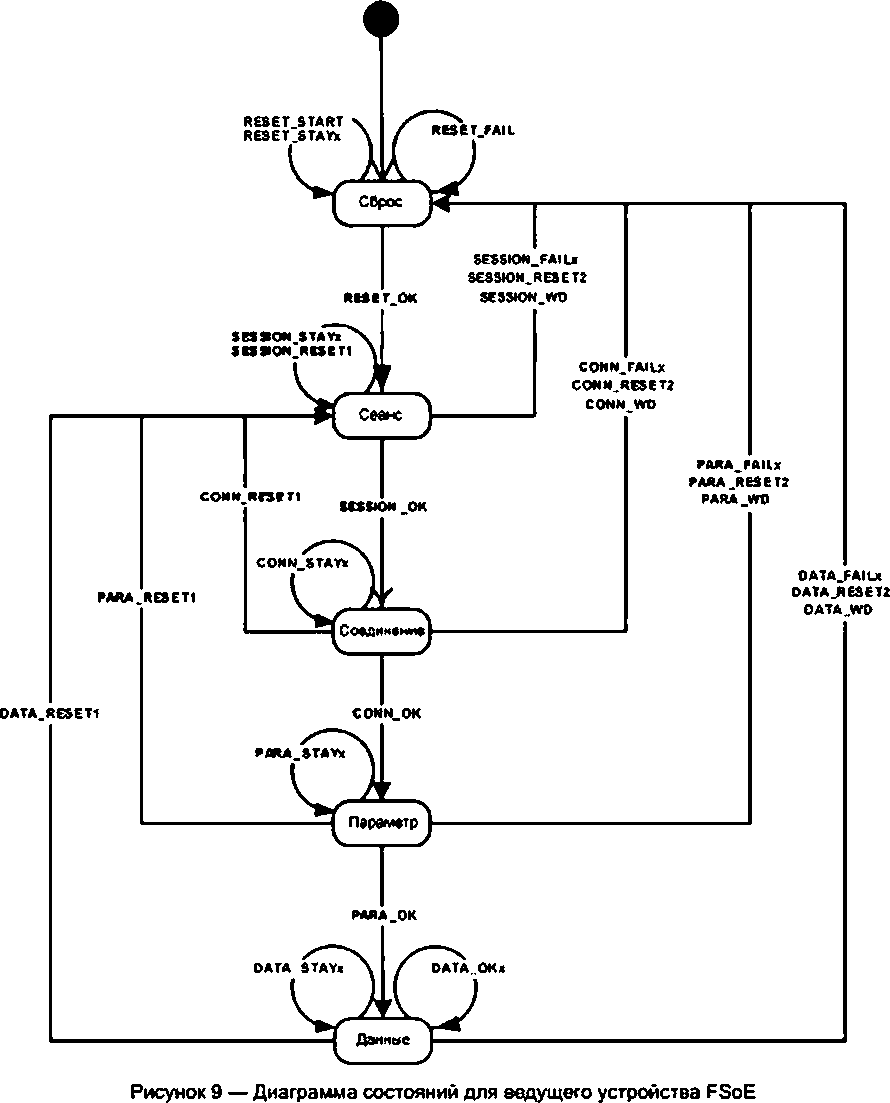
|  |  |
| --- | --- |
| Событие | Описание |
| Получен кадр | Был получен PDU безопасности, т. е. хотя бы один бит в PDU безопасности был изменен. Параметры:  Frame — полученный PDU безопасности:  Frame .Command — команда полученного PDU безопасности: Frame .ОсО— CRC\_0 полученного PDU безопасности:  Frame .Сопл Id — ID соединения полученного PDU безопасности: Frame.SafeData — данные безопасности полученного PDU безопасности |

27

#### ГОСТ Р МЭК 61784-3-12»2016

*Окончание таблицы 30*

|  |  |
| --- | --- |
| Событие | Описание |
| Истек сторо­ жевой таймер | Истек сторожевой таймер FSoE. т. е. за время сторожевого таймера не было получено никаких PDU безопасности.  Параметры: нет |
| Сброс Соеди­ нения | Запрос посредством локального интерфейса на сброс соединения FSoE. Данное событие долж­ но возникать при включении питания для запуска коммуникаций с ведомым устройством FSoE. Параметры: нет |
| Команда Set Data | Запрос посредством локального интерфейса на переключение SafeOulputs в состояние без­ опасности или на выход из состояния безопасности.  Параметры:  DataCmd — *FailSafeData или ProcessData* |



28

#### ГОСТ Р МЭК 61784-3.12—2016

* + - 1. Действия

1. за еисимостиот разных условий выполняются определенные действия, если происходит событие.

В таблицах состояний действия показаны в виде вызовов функций или присваиваний переменных.

6 таблице 31 перечислены функции, используемые в таблице состояний ведущего устройства

FSoE.

Таблица 31 — Функции в таблице состояний ведущего устройства FSoE

|  |  |
| --- | --- |
| Функция | Описание |
| SendFrame(cmd. safeData. lastCrc, connld. seqNo, old\* Crc.  bNew) | Отправлен кедр ведущего устройства FSoE. Параметры:  cmd — команда кадра:  SafeData — ссыгка на данные безопасности, отправленные с кадром;  tastCrc — CRC\_0 последнего PDU ведомого устройства безопасности, включенный в вычисление CRC для кадра:  connld — ID соединения, которое необходимо ввести в кадр и включить в вычисление CRC;  seqNo — Указатель на Порядковый номер ведущего устройства, включенный в вычисление CRC для кадров. Возвращается увеличенный (возможно неоднократно) seqNo;  oldCrc: указатель на CRC\_0 последнего отправленного PDU ведущего устройства безопасности. Возвращается вычисленный CRC\_0.:  bNew: если bNew = TRUE и oldCrc равен вычисленному crc. то вычисление CRC повто­ ряется с приращенным seqNo до тех пор. пока вычисленный сгс не станет равен oldCrc (процедура соответствует 7.1.3.4) |

8 таблице 32 перечислены переменные, используемые в таблице состояний ведущего устройства

FSoE.

Таблица 32 — Переменные, испогъзуемые а таблице состояний ведущего устройства FSoE.

|  |  |
| --- | --- |
| Состояние | Описание |
| LastCrc | CRC\_0 последнего отправленного PDU ведущего устройства безопасности (инициализируется значением 0 при включении питания) |
| OkJMasterCrc | CRC\_0 последнего отправленного PDU ведущего устройства безопасности (инициализируется значением 0 при включении питания) |
| OkJSIaveCrc | CRC\_0 последнего полученного PDU ведомого устройства безопасности (инициализируется значением 0 при включении питания) |
| MasterSeqNo | Порядковый номер ведущего устройства для испогъзовэния в CRC для следующего PDU ведущего устройства безопасности (инициализируется значением 0 при включении питания) |
| SlaveSeqNo | Ожидаемый порядковый номер ведомого устройства для использования в CRC следующего PDU ведомого устройства безопасности (инициализируется значением 0 при включении литания) |
| Sessionld | Произвольно генерируемый 10 сеанса (инициализируется значением 0 при включении питания) |
| DataCommand | Указывает на то. какая из команд *ProcessData* или *FatiSafeData* отправлена в состоя­ нии данных. Инициализируется с помощью *FaHSateData* при включении гытания |
| BytesToBeSent | Если несколько PDU блоков безопасности должно быть отправлено в состоянии сеанса, соединения или параметров, то эта переменная указывает на то. сколько еще октетов должно быть отправлено (инициализируется значение\*! 0 при включении питания) |
| ConnData | ConnData состоит из ID соединения и адреса ведомого устройства FSoE. Инициа­ лизируется конфигуратором безопасности при включении питания в соответствии с конфигурацией  ConnData.Connld: Connections соединения FSoE |

29

#### ГОСТ Р МЭК 617W-3-12—2016

*Окончание таблицы 32*

|  |  |
| --- | --- |
| Состояние | Описание |
| SafePara | SafePara состоит из параметров коммуникаций безопасности и приложения безопасности. Инициализируется конфигуратором безопасности при включении питания в соответствии с конфигурацией  SafePara.Watchdog: сторожевой таймер FSoE |
| SafeParaSize | Указывает на длину SafePara. Инициализируется конфигуратором безопасности при включении питания 8 соответствии с данными конфигурации |
| SafeOutputs | Содержит значения процесса выводов безопасности, отправленное ведомому устройству FSoE. Инициализируется с помощью  FS\_VALUE (Fail-safe Data ■ 0) при включении литания |
| Safelnputs | Содержит значения процесса вводов безопасности, полученных ведомым устройством FSoE. Инициализируется с помощью FS\_VALUE (Fail-safe Data = 0) при включении питания |
| CommFaultReason | Указывает на код ошибки в случае события возникновения коммуникационной ошибки |
| Second SessJonFrameSent | Если два блока PDU безопасности должны быть отправлены в состоянии Сеанса, то эта переменная указывает на то. был ли уже отправлен второй PDU.  Данная переменная устанавливается в значение FALSE макрокомандой CREATE SESSION JD |

* + - 1. Макрокоманды

Определенные функциональные возможности объединяются в макрокоманды для того, чтобы сохранять прозрачность таблиц состояний.

В таблице 33 перечислены макрокоманды из таблицы состояний ведущего устройства FSoE.

Таблица 33 — Макрокоманды в таблице состояний ведущего устройства FSoE

|  |  |
| --- | --- |
| Функция | Описание |
| IS\_CRC\_CORRECT(frame. lastCrc, seqNo. oldCrc. bNew) | Эта макрокоманда проверяет корректны ли CRC коды полученного PDU ведомого устройства безопасности.  Параметры:  Frame — принятый кадр:  lastCrc — CRC\_0 последнего отправленного PDU ведущего устройства безопасности, включенное в вычисления CRC для принятого PDU;  seqNo — порядковый номер ведомого устройства включается в вычисления CRC для принятого кадра. Возвращается приращенный (возможно неоднократно) seqNo;  oldCrc; указатель на CRC\_0 последнего принятого PDU ведомого устройства безопасности. Возвращается CRC.0 полученной телеграммы:  bNew: если bNew = TRUE и oldCrc равняется вычисленному сто. то вычисле­ ние CRC повторяется с приращенным seqNo до тех пор. пока вычисленный сто не перестанет быть равным oldCrc (процедура выполняется согласно 7.1.3.4) |
| UPDATE\_BYTES\_TO\_BE\_SENT  (bytesSent) | Данная макрокоманда проверят, сколько еще октетов в состояниях сеанса, со­ единения и параметров необходимо отправить перед изменением состояния. Параметры:  bytesSent — число октетов, ждущих отправления |
| IS\_SAFEDATA\_CORRECT  (frame. expectedData. bytesSent) | Данная макрокоманда проверяет, совпадает пи SafeData полученного PDU ведомого устройства безопасности с ожидаемыми данными.  Параметры:  Frame — принятый кадр:  expectedData — ссылка на ожидаемые данные: bytesSent — число отправленных октетов |

30

#### ГОСТ Р МЭК 61784-3.12—2016

*Окончание таблицы 33*

|  |  |
| --- | --- |
| Функция | Описание |
| START. WD (watchdog) | Данная макрокоманда перезапускает сторожевой таймер и запускает контролирующий таймер.  Параметры:  Watchdog — время контроля (мониторинга) в мс |
| CREATE.SESSION.ID | Данная макрокоманда генерирует произвольный 10 сеанса.  Переменная SecondSessionFrameSent (отправленный кадр второго сеанса) сбрасывается в значение FALSE |
| ADR | Данная макрокоманда генерирует ссылку (указатель) на переменную |

* + 1. Состояние сброса
       1. Событие принятия кадра

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее со­ стояние |
| RESET.OK | Frame.Command - Reset | Sesslonld CREATE\_SESSION\_ID(); SendFrame(Session,  ADR(Sesslonld*),* LastCrc, 0,  ADR(MasterSeqNo),  ADR(OldMasterCrc), FALSE*)}*  LastCrc - SendFsarae.CrcO; BytestoBesent u p d a t e \_b y t e s \_t o \_be \_ SENT(2);  START\_WD(SafePara.Watchdog); | Session (Сеанс) |
| RESET.STAY1 | Frame.Coemand <> Reset | LastCrc 0  OldMasterCrc 0;  OldSiaveCrc 0; MasterSeqNo :- l; SlaveSeqNo :- 1; DataCommand FallSafeData; CoemFaultReason :• 0; SendFrame(Reset,  ADR(CommFaultReason), LastCrc, 0,  ADR(MasterSeqNo),  ADR(OldMasterCcc), FALSE); MasterSeqNo :- 1; | Reset (Сброс) |

* + - 1. Событие истекшего сторожевого таймера

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состо­ яние |
| RESET.WD | Истек сторожевой тай\* мер | Sesslonld CREATE\_SESSION\_ID(); SendFrame(Session,  ADR(Sesslonld), LastCrc, 0,  ADR(MasterSeqNo),  ADR (OldMasterCrc), FALSE); LastCrc - SendFrame.CrcO;  BytesToBeSent UPDATE\_BYTES\_TO\_BE\_ SENT(2);  STARTWD(Sa fePat a.Watchdog); | Session (Сеанс) |

31

#### ГОСТ Р МЭК 61764.3-12—2016

* + - 1. Событие сброса соединения

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее со» стояние |
| RESET.START | Истек сторожевой таймер | LastCrc 0  OldMasterCrc :\*• 0;  OldSlaveCrc 0;  MasterSeqNo 1;  SlaveSeqNo 1;  DataCommand :- FallSafeData; CommFaultReason 0; SendFrame(Reset,  ADR(CommFaultReason), LastCrc, 0,  ADR(MasterSeqNo),  ADR(OldMasterCre), FALSE); MasterSeqNo 1; START\_WD(Sa Ге Pa ra. Watchdog); | Reset (Сброс) |

* + - 1. Событие Команды Set Data

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее со» стояние |
| RESET.STAY2 |  | DataCommand DataCmd; | Reset (Сброс) |

* + 1. Состояние сеанса
       1. Событие получения кадра

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| SESSJON.OK | Frame.Command - Session  AND  BytesToBeSent - 0 AND  IS\_CRC\_CORRECT(Frame,  LastCrc, AOR(SlaveSeqNo), ADR(OldSlaveCrc),  TRUE) - TRUE | LastCrc :• Frame.CrcO; SendFrame(Connection,  ADR(ConnData), LastCrc, ConnDa ta.Connld, ADR(MasterSeqNo), ADR(OldMasterCrc), TRUE);  LastCrc SendFrame.CrcO; BytesToBeSent UPDATE\_BYTES\_ TO\_BE\_SENT(4);  START\_WD(SafePara.Watchdog); | Connection (Соединение) |
| SESSION.FAILI | Frame.Command -  *Session*  AND IS\_CRC\_CORRECT(Frame,  LastCrc, ADR(SlaveSeqNo), ADR(OldSlaveCrc), TRUE) - FALSE  AND  SecondSessionFrameSent  - TRUE | LastCrc :• 0  OldMasterCrc :» 0;  OldSlaveCrc :• 0;  MasterSeqNo 1;  SlaveSeqNo 1; DataCoomand *FsiiSsieDsts;*  CommFaultReason INVALID\_CRC; SendFrame(Reset,  ADR(CommFauLtReason), LastCrc,  o, ADR(MasterSeqNo),  ADR(OldMasterCrc), FALSE); KasterSeqNo 1; START\_ WD(SafePara.Watchdog); | Reset (Сброс) |

32

#### ГОСТ Р МЭК 61784-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| SESSION.STAY2 | Frame.Cotroand -  *Session*  AND IS\_CRC\_CORRBCT(Frame,  LastCrc, ADR(SlaveSeqNo), ADR(OldSlaveCrc), TRUE) • FALSE  AND  SecondSessicnFrameSent  - FALSE | START\_WD(SaCePara.Watchdog); | Session (Сеанс) |
| SESSION\_STAY 1 | Frame.Coecnand -  *Session*  AND  BytesToBeSent <> 0  AND IS\_CRC\_CORREC7(Frame,  LastCrc, ADR(SlaveSeqNo), ADR(OidS1a veC r C), TRUE) - TRUE | LastCrc :» Frame.CrcO; SendFrame( Session,  ADR(SessionZd  (2-BytesToBeSent)), Frame. CrcO,  0,  ADR(MasterSeqNo), ADR(OldMasterCrc), TRUE);  LastCrc :« SendFrame.CrcO; BytesToBeSent UPDATE\_BYTES\_ TO\_BE\_SENT(  BytesToBeSent); SecondSessionFrameSent TRUE; START\_WD(SafePara.Watchdog); | Session (Сеанс) |
| SESSION RESET1 | Frame .Coemand - *Reset* | LastCrc 0  OldMasterCrc 0;  OldSlaveCrc 0;  MasterSeqNo 1;  SlaveSeqNo :- 1; Sessionld :- CREATE\_SESSI0N\_1D();  DataCommand *FailSafeD&ta;* SendFrame(Session, ADR(Sessionld), LastCrc,  0,  ADR(MastecSeqNo), ADR(OldMaSterCrC), FALSE);  LastCrc « SendFrame.CrcO; BytesToBeSent UPDATE\_BYTBS\_ TO\_BE\_SENT(2);  START~WD(SaCePara.Watchdog); | Session (Сеанс) |
| SESSION\_FAIL3 | Frame.Coecnand -  *Connection*  OR  Frame.Coetmand • Parameter  OR  Frame.Coecnand -  *PeocessDaia*  OR  Frame.Coomand •  *FailSateData* | LastCrc :• 0  OldMasterCrc 0;  OldSlaveCrc 0;  MasterSeqNo 1;  SlaveSeqNo :- 1; DataCommand *FailSateDaca;*  CommFaultReason 1NVAL1D\_CMD; SendFrame(Reset,  AOR(CommCaultReason), LastCrc,  0,  ADR(MasterSeqNo), ADR(OldMasterCrc), FALSE); MasterSeqNo 1; START\_ WD(SafePara.Watchdog); | Reset (Сброс) |

33

#### ГОСТ Р МЭК 61764.3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состоим не |
| SESSION FAIL4 | Frame.Command *О Reset* | LastCrc :- 0 | Reset (Сброс) |
| AND | OldMasterCrc :• D; |
| Frame.Command <> | OldSlaveCrc 0; |
| *Session* | MasterSeqNo 1; |
| AND | SlaveSeqNo :- 1; |
| Frame.Command <> | DataCoooand :• FailSafeData; |
| Connection | CommFau11Rea son UNKNOWN CMD; |
| AND | SendFrame(Reset, |
| Frame.Command <> |  |
| Parameter | ADR(CommFaultReason), |
| AND | LastCrc, |
| Frame.Command <> | 0, |
| ProcessData | ADR(MasterSeqNo), |
| AND | ADR (OldMasterCrc), FALSE) |
| Frame.Command <> | MasterSeqNo 1; START |
| *FsllShfeDsts* | WDlSafePara.Watchdog); |

* + - 1. Событие истекшего сторожевого таймера

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее со» стояние |
| SESSION.WD |  | LastCrc О OldMasterCrc 0;  OldSlaveCrc 0;  MasterSeqNo :• 1;  SlaveSeqKo :• 1,' DataCommand FailSateData;  CommFaultReason :• WD\_EXPIRED; SendFrame(Reset, ADR(CommFaultReason), LastCrc, 0,  ADR(MasterSeqNo), ADR(OldMasterCre),  FALSE*);*  MasterSeqNo 1; START\_WD(SafePaca. Watchdog); | Reset (Сброс) |

* + - 1. Событие сброса соединения

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее со­ стояние |
| SESStON\_RESET2 |  | LastCrc 0  OldMasterCrc 0; OldSlaveCrc О; MasterSeqNo 1;  SlaveSeqNo 1; DataCommand FailSafeData;  CommFaultReason 0; SendFramefReset, ADR(CommFaultReason), LastCrc,  0,  ADR(MasterSeqNo), ADR(OldMasterCrc), FALSE*)i*  MasterSeqNo 1; START\_MD(SafePara.Watchdog); | Reset (Сброс) |

34

#### ГОСТ Р МЭК 61784-3.12—2016

* + - 1. Событие Команды Set Data

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| SESSION\_STAY2 |  | DataCommand :• DataCmd; | Session (Сеанс) |

* + 1. Состояние соединения
       1. Событие получения кадра

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следую шее состояние |
| CONN OK | Frame.Coemand » *Connection* | LastCrc Frame.CrcO; | Parameter |
| AND | SendFrame(Parameter, | (Параметры) |
| BytesToBeSent - 0 | ADR(SafePara), |  |
| AND | Frame.CrcO, ConnData. |  |
| Frame.Connld - ConnData. | Connld, ADR(MasterSeqNo), |  |
| Connld | ADR(OldMasterCrc), TRUE); |  |
| AND | LastCrc SendFrame.CrcO; |  |
| IS SAFEDATA CORRECT (Frame, | BytesToBeSent UPDATE |  |
| ADR(ConnDaca), | BYTES TO BE SENT( |  |
| 4-BytesToBe Sent) - TRUE AND | SaCeParaSize); START |  |
| IS\_CRC\_CORRECT(Fr ame, | WD(Sa£ePara.Watchdog); |  |
| LastCrc, ADR(SlaveSe^No), ADR(OldS1aveC r C), TRUE) - TRUE |  |  |
| CONN FAJL1 | Frame.Coereiand - *Connection* | LastCrc :- 0 | Reset (Сброс) |
| AND | OldMasterCrc 0; |
| Frame.Connld - ConnData. | OldSiaveCrc :• 0; |
| Connld | MasterSeqNc 1; |
| AND | SlaveSeqNo I,- |
| IS SAFEDATA CORRECT(Frame, | DataCoemand *FailSateDa*ca; |
| ADR(ConnData), | ComnFaultReason :• INVALID |
| 4-BytesToBesent) - t r u e a nd | CRC; |
| IS CRC CORRECT(Frame, | SendFrame(Reset, |
| LastCrc,  ADR(SlaveSeqNo), | ADR(ComnFaultReason), |
| ADR(OldS1aveC r 0), | LastCrc, 0, |
| TRUE) - FALSE | ADR(MasterSeqNo), ADR(OldMasterCrc), FALSE); MaSterSeqNo :- 1; START\_ HD(SafePara.Watchdog); |
| CONN FAIL2 | Frame.Command - *Connection* | LastCrc :- 0 | Reset (Сброс) |
| AND | OldMasterCrc 0; |
| Frame.Connld - | OldSiaveCrc 0; |
| ConnData.Connld | MasterSeqNo :• 1; |
| AND | SlaveSeqNo 1,- |
| IS SAFEDATA CORRECT(Frame, | DataCoemand :- *FailSafeData;* |
| ADR(ConnData), | CoentFaultReason INVALID |
| 4-BytesToBeSent) - FALSE | DATA;  SendFrame(Reset, |
|  | ADR(Co«nFaultReason), LastCrc, 0, ADR(MasterSeqNo), ADR(OldMasterCrc), FALSE); MasterSeqNo :- 1; START\_ WD(Sa£ePara.Watchdog); |

35

#### ГОСТ Р МЭК 61784-3-12»2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| CONN.FAIL3 | Frame.Command - Connection AND  Frame.Connld <> ConnData. ConnId | LastCrc 0  OldMasterCrc 0;  OldSlaveCrc 0;  MasterSeqNo :- 1;  SlaveSeqNo 1;  DataCommand :• FailSafeOace; CommFaoltReason INVALID\_ CONNID;  SendFcame<Reset, | Reset (Сброс) |
| ADR(CommFaoltReason), LastCrc, 0, ADR(MasterSeqNo), ADR(OldMasterCrc), FALSE);  MasterSeqNo 1; START\_ WD(SafePara.Watchdog); |
| CONN.STAY1 | Frame.Command - Connection AND  BytesToBeSent <> 0 AND  Frame.Connld - ConnData. ConnId  AND IS\_SAFEDATA\_CORRECT(Frame,  ADR(ConnData),  4-BytesToBeSent) - TRUE AND IS\_CRC\_CORRECT < Frame,  LastCrc, ADR(SlaveSeqNo), ADR(OldSlaveCrc), TRUE) - TRUE | LastCrc :«• Frame.CrcO; SendFrame(Connection, ADR{ConnData(4- ByteaToBeSent]), Frame.CrcO, ConnData. Connld, ADR(KasterSeqNo), ADR(OldMasterCrc), TRUE); LastCrc SendFrame.CrcO; BytesToBeSent UPDATE\_ 8YTES\_T0\_BE\_SENT(  BytesToBeSent); START\_ WDfSafePara.Watchdog); | Connection (Соединение) |
| CONN.RESET1 | Frame.Command - Reset | LastCrc :«• 0  OldMasterCrc 0;  OldSlaveCrc :«• 0;  MasterSeqNo 1;  SlaveSeqNo 1; DataCommand FaiiSafeOata; sessionld CREATE SESSION ID<>;  SendFrame(Session, ADR(Sessionld), LastCRC, 0,  ADR(MasterSeqNo),  ADR(OldMasterCrc), FALSE); LastCrc - SendFrame.CrcO BytesToBeSent :- UP0ATE\_ 8YTES\_T0\_8E\_SENT(*2*);  START~WD(SafePara.Watchdog); | Session (Сеанс) |

36

#### ГОСТ Р МЭК 61784-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| CONN.FAIL4 | Frame.Coecnand • *Session*  OR  Frame.Coecnand - Parameter OR  Frame.Coecnand -  *PeocessData*  OR  Frame.Coacnand - FaiiSafeData | LastCrc 0  OldMasterCrc 0;  OldSlaveCrc 0,-  MasterSeqNo 1;  SlaveSeqHo 1; DataCoonand FaUSa/eData;  CocreiFaultReason :- INVALID\_ CHD;  SendFrame(Reset,  ADR(ComnFaultReason), LastCrc, 0, ADR(MasterSeqNo), ADR(OldMasterCrc), FALSE); MasterSeqNo :• 1; START\_ HDfSafePara.Watchdog),- | Reset {Сброс) |
| CONN.FAIL5 | Frame.Coecnand <> Reset AMD  Frame.Coecnand <> *Session*  AMD  Frame.Coecnand <>  *Connection*  AMD  Frame .Coecnand <> Parameter AMD  Frame .Coecnand <> Processesta  AMD  Frame.Coecnand <>  *FallSafeData* | LastCrc :- 0  OldMasterCrc 0;  OldSlaveCrc 0;  MasterSeqMo 1;  SlaveSeqHo 1,- DataCoercnand FaiiSa/eData;  ComnFaultReason UNKNOWN\_  CMD;  SendFrame(Reset,  ADR(CoeeiFaultReason), LastCrc, 0, ADR(MasterSeqNo), ADR(OidMasterCrc), FALSE); MasterSeqNo 1; START\_ WD(SafePara.Watchdoq); | Reset (Сброс) |

* + - 1. Событие истекшего сторожевого таймера

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее со­ стояние |
| CONN.WD |  | LastCrc :« 0  OldMasterCrc 0;  OldSlaveCrc 0,-  MasterSeqNo 1;  SlaveSeqNo :• 1,- DataCoonand FailSateData;  CoecnFaultReason :« WD\_EXPIRED; SendFraeie (Reset, ADR(CoeMFeultReason), LastCrc,  0,  ADR(MasterSeqNo), ADR(OldMasterCrc), FALSE);  MasterSeqMo 1,- S?ART\_WD(SafePara.Vatchdoq); | Reset (Сброс) |

37

#### ГОСТ Р МЭК 61764.3-12—2016

* + - 1. Событие сброса соединения

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее со\* стояние |
| CONN.RESET2 |  | LastCrc :• 0  OldMasterCrC 0;  OldSlaveCrc 0;  KasterSeqNo 1;  SlaveSeqNo :- 1; DataCommand FaLlSafeData;  CommFaultReason :• 0; SendFrame(Reset, ADR(CocmFaultReason), LastCrc,  0,  ADR(KasterSeqNo), ADR(OldMasterCrC), FALSE) ;  MastecSeqNo 1; START\_WD<SaCe?ara.Watchdogl; | Reset (Сброс) |

* + - 1. Событие Команды Set Data

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состоим ие |
| CONN.STAY2 |  | DataCocnnand DataCmd; | Connection (Соединение) |

* + 1. Состояние параметров (параметрическое)
       1. Событие получения кадра

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| PARA OK | Frame.Command - Parameter | LastCrc Frame.CrcO; | Data (Данные) |
| AND | SendFrame{DataCommand, |
| BytesToBeSent • 0 | ADR{SafeOutputs), Frame. |
| AND | CrcO, ConnData.Connld, |
| Fcame.Connld • ConnData. | ADR(KasterSeqNo), |
| ConnEd | ADR(OldKasterCrc), TRUE); |
| AND | LastCrc SendFrame.CrcO; |
| IS\_5AFBDATA\_C0RRECT <Frame,  ADR(SafePara), SafeParaSize-  ByteaToBeSenC) • TRUE AND IS\_CRC\_CORRECT{Frame,  LastCrc, ADR(SlaveSeqNo), ADR{OldSlaveCrc), TRUE) - TRUE | START\_WD(SafePara.Watchdog); |
| PARA FAIL1 | Frame.Command - Parameter | LastCrc :• 0 | Reset (Сброс) |
| AND | OldKasterCrc 0; |
| Frame.Connld - ConnData. | OldSlaveCrc :• 0; |
| Connld | KasterSeqNo :- 1; |
| AND | SlaveSeqNo :- 1; |
| IS SAFEDATA CORRECT{Frame, | DataCommand *FallSaCeData;* |
| ADR(SafePara), | CommFaultReason :• INVALID CRC; |
| SafeParaSlze- | SendFrame{Reset, |
| ByteaToBeSent) • TRUE  AND | ADR(CommFaultReason), |
| IS CRC CORRECT{Frame, | LastCrc, 0, |
| LastCrc, | ADR(KasterSeqNo), |
| ADR{SlaveSeqNo), | ADR(OldKasterCrc), FALSE); |
| ADR(OldSlaveCrc), | KasterSeqNo 1; START |
| TRUE) - FALSE | WD(SafePara.Watchdog); |

38

#### ГОСТ Р МЭК 61784-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| PARA\_FA!12 | Frame.Command - Parameter AND  Frame.Connld • ConnData. Connld  AND IS\_SAFEDATA\_CORRECT(Frame,  ADR(SafePara), SafeParaSize- BytesToBeSent) - FALSE | LastCrc 0  OldMasterCrc 0;  OldSlaveCrc 0;  HasterSeqNo :- 1; SlaveSeqNo : - 1,-  DataCommand :• *FailSaf&Data;* CommFaultReason INVAL1D\_ DATA;  SendFrame(Reset, | Reset (Сброс) |
| ADR(CommFaultReason), LastCrc, 0, ADR(MasterSeqNo), ADR(OldMasterCre), FALSE); MasterSeqNo 1; START\_ WO(SafePara.Watchdog); |
| PARA.FAIL3 | Frame.Command - Parameter ANO  Frame.Connld <> ConnData.Connld | LastCrc :« 0  OldMasterCrc 0;  OldSlaveCrc 0;  MasterSeqNo :• 1;  SlaveSeqNo :• 1;  DataCommand :<• PallSafeOata; CommFaultReason INVALID\_ CONNID;  SendFrame(Reset, | Reset (Сброс) |
| ADR(CommFaultReason), LastCrc, 0, ADR(MasterSeqNo), ADR(OldMasterCrc), FALSE); MasterSeqNo 1; START\_ WO(SaCePara.Watchdog); |
| PARA.STAY1 | Frame.Command - Parameter ANO  BytesToBeSent <> 0 ANO  Frame.Connld - ConnOata. Connld  ANO JS\_SAFEDATA\_CORRBCT(Frame,  ADR(SafePara), SafeParaSize- BytesToBeSent) • TRUE ANO  1S\_CRC\_CORRECT(Fr ame,  LastCrc, ADR(SlaveSeqNo), ADR(OldSlaveCrC), TRUE» - TRUE | LastCrc :• Frame.CrcO; SendFtame( Parameter, ADR(SaCePara[SateParaSize- BytesToBeSent)), Frame.CrcO, ConnData.Connld, ADR(MasterSeqNo), AOR(OldMasterCrc),  TRUE);  LastCrc :• SendFrame.CrcO; BytesToBeSent UPOATE\_BYTES\_ TO\_BE\_SENT(  BytesToBeSent); START\_ WD(SafePara.Watchdog); | Parameter (Параметры) |

39

#### ГОСТ Р МЭК 61784-3-12»2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| PARA.RESET1 | Frame.Command • Reset | LastCrc 0  OidMasterCrc 0;  OldSlaveCrc 0;  MasterSeqNo :- 1; SlaveSeqNo :- l; DataCommand *FaiiSafeData;*  Sessionld :• CREATE SESSION IDO;  SendFrame{Session, ADR(Sessionld), LastCRC, 0,  ADR(MasterSeqNo),  ADR(OidMasterCrc), FALSE); LastCrc • SendFrame.CrcO BytesToBeSent UPDATE\_BYTES\_ TO\_BE\_SENT(2);  START\_WD < Sa fePara.Watchdog); | Session (Сеанс) |
| PARA\_FAIL4 | Frame.Command - *Session*  OR  Frame.Command • *Connection*  OR  Frame.Command -  *ProcessData*  OR  Frame.Command -  *FailSateData* | LastCrc 0  OidMasterCrc 0;  OldSLaveCrc 0;  MasterSeqNo 1; SlaveSeqNo :• l;  DataCommand *i- FailSafeData;*  CommFaultReason INVALID\_CMD; SendFrame(Reset,  ADR(CommFaultReason), LastCrc, 0, ADR(MasterSeqNo), ADR(OidMasterCrc), FALSE); MasterSeqNo l; START\_ WD(SafePara.Watchdog); | Reset (Сброс) |
| PARA\_FAIL5 | Frame.Command <> Reset AMD  Frame.Command <> *Session*  AMD  Frame.Command <> Connection  AND  Frame.Command *<>* Parameter AND  Frame.Command *<> PeocessDa*ta  AND  Frame.Command <>  *FailSaCeData* | LastCrc 0  OidMasterCrc :» 0;  OldSLaveCrc 0;  MasterSeqNo 1;  SlaveSeqNo :\*• 1; DataCommand :• *FailSaCeData;*  CommFaultReason UNKNOWN\_CMD; SendFrame(Reset,  ADR(CommFaultReason), LastCrc, 0, ADR(MasterSeqNo), ADR(OidMasterCrc), FALSE); MasterSeqNo l; START\_ HD(SafePara.Watchdog); | Reset (Сброс) |

40

#### ГОСТ Р МЭК 61784-3.12—2016

* + - 1. Событие истекшего сторожевого таймера

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условно | Действие | Следующее со­ стояние |
| PARA.WD |  | LastCrc 0  OldMasterCrc 0;  OldSlaveCrc 0,-  MastetSeqNo 1;  SlaveSeqNo :- 1; DataCommand FailSaCeData;  ComfflFaultReason WO\_EXPIRED; SendFtame(Reset, ADR(CommFaultReason», LastCrc, 0,  ADR(MastetSeqNo», ADR(01dMasterCrc»,  FALSE»;  MasterSeqNo 1; START\_WD(SaTePara. Watchdog»; | Reset (Сброс) |

* + - 1. Событие сброса соединения

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| PARA\_RESET2 |  | LastCrc 0  OldMasterCrc 0;  OldSlaveCrc 0;  MasterSeqNo 1;  SlaveSeqNo :« 1; DataCommand FailSafeData; CommFaultReason 0; SendFrame(Reset, ADR(CommFaultReason», LastCrc,  0,  ADR(MasterSeqNo», AOR(OldMasterCrc», FALSE»;  MasterSeqNo 1; START\_WO(SafePara.Watchdog*»;* | Reset (Сброс) |

* + 1. Событие команды Set Data

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| PARA.STAY2 |  | OataCommand OataCmd; | Parameter (Параме­ тры) |

41

#### ГОСТ Р МЭК 61764.3-12—2016

* + 1. Состояние данных

7.4.6.1 Событие получения кадра

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| DATA ОК1 | Frame.Command - | Safelnputs Frame.SaleData; | Data (Данные) |
| Processed La | LastCrc Frame.CreO; |
| AMD | SendFrame (DataCocmand, |
| Frame.Conoid - ConnData. | ADR(SafeOutputs), Frame.CrcO, |
| Connld | ConnData.Connld, ADR(MasterSeqNo), |
| AMD | ADR(OldMascerCre), TRUE); |
| IS CRC CORRECT(Frame, | LastCrc SendFrame.CrcO; START |
| LastCrc, ADR(SlaveSeqNo), ADR(OldSIaveCrc),  TRUE) - TRUE | WDfSafePara.Watchdog); |
| DATA\_OK2 | Frame.Command • | Safelnputs FS\_VALUE; | Data (Данные) |
| FaiiSafeData | LastCrc :- Frame.CrcO; |
| AMD | SendFr ame (DataComnand, |
| Frame.Connld • ConnData. | ADR(SafeOutputs), Frame.CrcO, |
| Connld | ConnData.Connld, ADR(MasterSeqNo), |
| AMD | ADR(OldMasterCrc), TRUE); |
| IS CRC CORRECT(Frame, | LastCrc SendFrame.CrcO; START |
| LaatCrc, ADR(SlaveSeqMo), ADR(OldSIaveCrc), TRUE) - TRUE | WDfSafePara.Watchdog); |
| DATA.FAIH | (Frame.Command - | LastCrc :- 0 | Reset (Сброс) |
| Processedta | OldMasterCrc 0; |
| OR | OldSIaveCrc :• 0; |
| Frame.Command - | MasterSeqMo 1; |
| FaJlSa/eData) | SlaveSeqMo 1; |
| AMD | DataComnand :• *FailSaieData;* |
| Frame.Connld - ConnData. | Safelnputs FS VALUE; |
| Connld | CoetnFaultReason :• INVALID CRC; |
| AMD | SendFrame(Reset, |
| IS CRC CORRECT(Frame, | ADR(CoetnFaultReason), LastCrc, |
| LastCrc, | 0, |
| ADR(SlaveSeqMo), | ADR(MasterSeqMo), |
| ADR(OldSIaveCrc), | ADR(OldMasterCrc), FALSE); |
| TRUE) - FALSE | MasterSeqMo 1; START\_  WDfSafePara.Watchdog); |
| DATA\_FAiL2 | (Frame.Command - | LastCrc 0 | Reset (Сброс) |
| *ProceasDa*ta | OldMasterCrc 0; |
| OR | OldSIaveCrc 0; |
| Frame.Command - | MasterSeqMo :• 1; |
| *FailSa(eData)* | SlaveSeqMo 1; |
| AMD | DataCoemand *FailSafeData;* |
| Frame.Connld <> ConnData. | Safelnputs FS VALUE; |
| Connld | CoetnFaultReason INVALID\_CONNID SendFrame(Reset, ADR(ComeiFaultReason), LastCrc,  0,  ADR (MasterSeqMo), ADR(OldMasterCrc), FALSE); MasterSeqMo :• 1; START\_WD(SafePara.Watchdog); |

42

#### ГОСТ Р МЭК 61784-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| DATA | Frame.Command • *Reset* | LaatCrc 0 | Session |
| RESET1 | OldMastecCre 0;  OldSlaveCrc 0;  MasterSeqNo 1;  SlaveSeqNo 1;  OataCommand :» FailSafeOata; | (Сеанс) |
|  | Safelnputs FS\_VALUE;  Sesslonld CREATE SESSION IDO; |  |
|  | SendFrame(Session, ADR(Sessionld), LastCRC,  o, ADR(MasterSeqNo),  ADR(OldMasterCrc), FALSE); LastCrc - SendFrame.CrcO  BytesToBeSent UPDATE\_BYTES\_TO\_ BE\_SENT<2);  START\_WD(SafePara.Watchdog); |  |
| DATA FAILS | Frame.Command - Session | LastCrc 0 | Reset (Сброс) |
| OR | OldMastecCre :• 0; |
| Frame.Command - Connection | OldSlaveCrc 0; |
| OR | MasterSeqNo 1; |
| Frame.Command - Parameter | SlaveSeqNo 1; OataCommand FailSafeOata; CommFAultReason INVALID\_ CMC; Safelnputs FS\_VALU£; SendFrame(Reset,  ADR(CommFaultReason), LastCrc,  0,  ADR(MasterSeqNo), AOR(OldMasterCrc), FALSE); MasterSeqNo 1; START\_ HO(SafePara.Watchdog); |
| DATA FAIL4 | Frame.Command <> Reset | LastCrc 0 | Reset (Сброс) |
| AND | OldMastecCre *:•* 0; |
| Frame.Command <> Session | OldSlaveCrc 0; |
| AND | MasterSeqNo 1; |
| Frame.Command о Connection | SlaveSeqNo 1; |
| AND | OataCommand FailSafeOata; |
| Frame.Command <> Parameter | Safelnputs FS VALUE; |
| AND | CommFaultReason UNKNOWN CMD; |
| Frame.Command <> | SendFrame(Reset, |
| ProcessOata | ADR(CommFaultReason), LastCrc, |
| AND | 0, |
| Frame.Command <> | ADR(MasterSeqNo), |
| FailSafeOata | ADR(OldMasterCrc), FALSE); MasterSeqNo 1; START\_WD(SafePara.Watchdog); |

43

#### ГОСТ Р МЭК 61764.3-12—2016

7,4.6.2 Событие истекшего сторожевого таймера

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее со\* стояние |
| DATA.WD |  | LastCcc 0  OldMafiterCrc 0;  OldSlaveCrc 0;  MastacSeqNo 1;  SlaveSeqNo 1; DataCommand FailSafeData; Safelnputs FS\_VALUE;  CommFaultReaaon WD\_EXPIRED SendFcame(Reset, ADR(CommFaultReaaon), LdStCcc, 0,  ADR(MastecSeqNo), ADR(OldMasterCre),  FALSE);  MastecSeqNo 1; START\_WD(SafePara. Watchdog*);* | Reset (Сброс) |

* + - 1. Событие сброса соединения

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее со\* стоянке |
| DATA.RESET2 |  | LdStCcc 0  OldMasterCre 0;  OldSlaveCec 0;  MasterSeqNo 1;  SldveSeqNo 1; DataCommand FailSafeData; safelnputs FS\_VALUE; CommFaultReason 0; SendFrame(Reset,  ADR(ComaFaultReason), LastCrc, 0,  ADR(MasterSeqNo), ADR(OldMasterCre),  FALSE);  MasterSeqNo 1; START\_WD{SafePara. Watchdog); | Reset (Сброс) |

* + - 1. Событие Команды Set Data

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| DATA.STAY |  | DataCostnand DataCmd; | Data (Данные) |

* 1. Таблица состояний для ведомого устройства FSoE
     1. Машина состояний для ведомого устройства FSoE
        1. Обзор

В зависимости от коммуникационной процедуры, ведомое устройство FSoE может находиться в состояниях, перечисленных в таблице 34.

Таблица 34 — Состояния ведомого устройства FSoE

|  |  |
| --- | --- |
| Состояние | Описание |
| Сброс | Соединение FSoE сброшено (выводы в безопасном состоянии) |
| Сеанс | Передается ID сеанса (выводы в безопасном состоянии) |
| Соединение | Передается ID соединения (выводы в безопасном оостоянии) |

44

#### ГОСТ Р МЭК 61784-3.12—2016

*Окончание таблицы 34*

|  |  |
| --- | --- |
| Состояние | Описание |
| Параметры | Передаются параметры (выводы в безопасном состоянии) |
| Данные | Передаются данные промесса или отказоустойчивые данные (выводы активны, только ест получена команда *PmcassData)* |

Диаграмма состояний для ведомого устройства FSoE показана на рисунке 10.

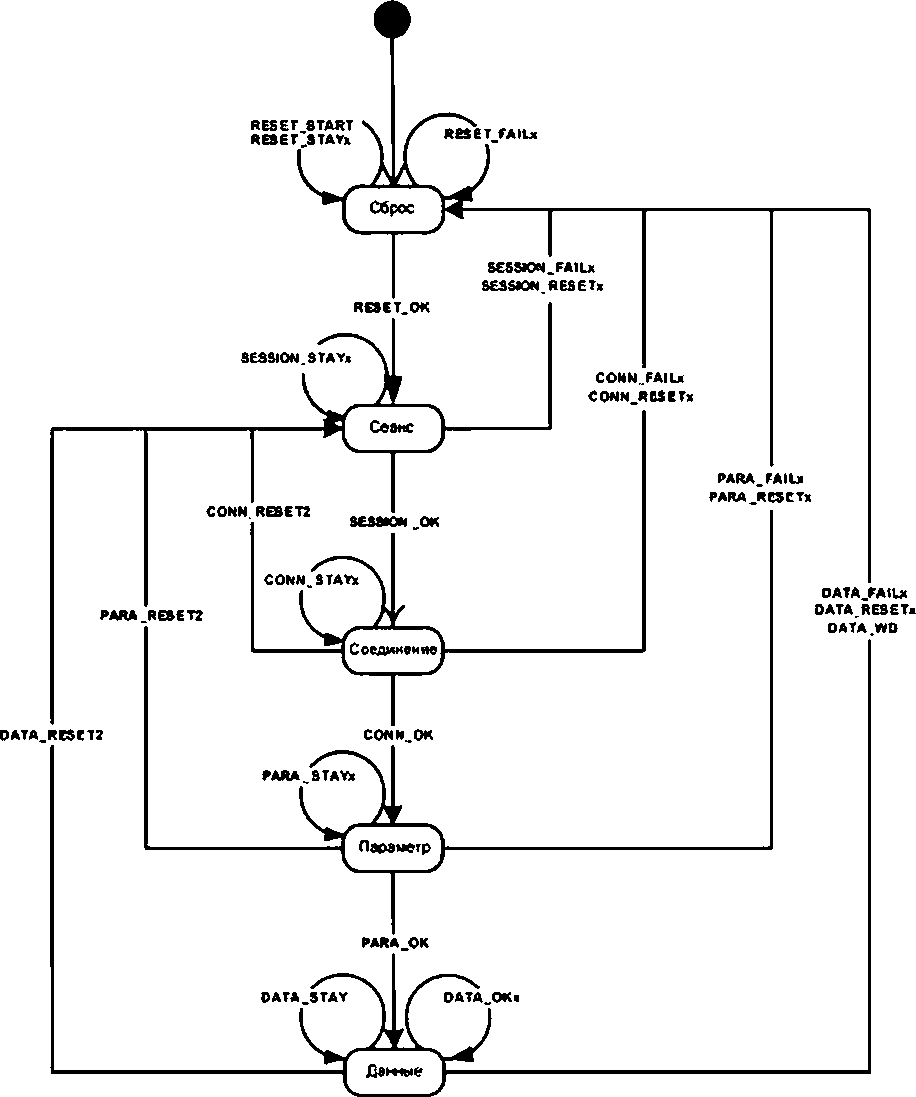


Рисунок 10 —Диаграмма состояний для ведомого устройства FSoE

8 следующих секциях проводится анализ событий, которые могут произойти на ведомом устройстве FSoE для каждого состояния.

Каждое событие рассматривается в условиях различных действий или проистекающих состояний.

45

#### ГОСТ Р МЭК 617W-3-12—2016

* + - 1. События

Событие может включать в себя различные параметры, на которые приводятся ссылки в таблицах состояний. В таблице 35 приведен список используемых событий.

Таблица 35 — События а таблице состояний ведомого устройства FSoE

|  |  |
| --- | --- |
| Событие | Описям ие |
| Получен кадр | Был получен PDU безопасности, т. е. хотя бы един бит в PDU безопасности был изменен . Параметры:  Frame — полученный PDU безопасности;  Frame.Command — команда полученного PDU безопасности: Frame.CrcO — CRC\_0 полученного PDU безопасности: Frame.Connld — ID соединения полученного PDU безопасности;  Frame.SafeOata — данные безопасности полученного PDU безопасности |
| Истек стороже­ вой таймер | Истек сторожевой таймер FSoE. т.е. за время сторожевого таймера не было получено никаких PDU безопасности.  Параметры: нет |
| Сброс Соедине­ ния | Запрос посредством локального интерфейса на сброс Соединения FSoE. Параметры: нет |
| Команда Set Data | Запрос посредством локального интерфейса на переключение Safelnputs в состояние без­ опасности или на выход из состояния безопасности.  Параметры:  DataCmd — *FatlSafeData* или *ProcessData* |

* + - 1. Действия

В зависимости от разных условий выполняются определенные действия, если происходит событие.

1. таблицах состояний действия показаны в виде вызовов функций или присваиваний переменных.

В таблице 36 перечислены функции, используемые в таблице состояний ведомого устройства FSoE.

Таблица 36 — Функции в табгмце состояний ведущего устройства FSoE

|  |  |
| --- | --- |
| Функция | Описание |
| SendFrame(cmd. safeData. lastCrc. connld. seqNo. oldCrc. bNew) | Отправлен кадр ведомого устройства FSoE. Параметры:  Crrtd — команда кадра:  SafeData — осыпка на данные безопасности, отправленные с кадром;  lastCrc — CRC\_0 последнего PDU ведущего устройства безопасности, включенный в вычисление CRC для кадра:  conntd — ID ооединешя. которое необходимо ввести е кадр и вклочить в вычисление CRC; seqNo—указательна порядковый номер ведомого устройства, включенный в вычисление CRC для кадров. Возвращается приращённый (возможно неоднократно) seqNo:  oldCrc — указатель на CRC\_0 последнего отправленного PDU ведомого устройства безопасности. Возвращается вычисленный CRC\_0.:  bNew — если bNew = TRUE и oldCrc равен вычисленному сгс, то вычисление CRC повторяется с приращенным seqNo до тех пор. пока вычисленный сгс не станет равен oldCrc (процедура соответствует 7.1.3.4) |

В таблице 37 перечислены переменные, используемые в таблице состояний ведомого устройства

FSoE.

Таблица 37 — Переменные, используемые в таблице состояний ведомого устройства FSoE

|  |  |
| --- | --- |
| Лереыенмая | Описание |
| LastCrc | CRC\_0 последнего PDU ведомого устройства безопасности (инициализируется значени­ ем 0 при вхлючении питания) |
| OldMasterCrc | CRC\_0 последнего полученного PDU ведущего устройства безопасности (инициализи­ руется значением 0 при включении питания) |

46

#### ГОСТ Р МЭК 61784-3.12—2016

*Окончание таблицы 36*

|  |  |
| --- | --- |
| Переменная | Описание |
| OldSlaveCrc | CRC\_0 последнего отправленного PDU ведомого устройства безопасности (инициали­ зируется значением 0 при включении питания) |
| MasterSeqNo | Ожидаемый порядковый номер ведущего устройства для использования в CRC для сле­ дующего PDU ведущего устройства безопасности (инициализируется значением 0 при включении питания) |
| SlaveSeqNo | Порядковый номер ведомого устройства для использования в CRC следующего PDU ве­ домого устройства безопасности (инициализируется значением 0 при включении питания) |
| InitSeqNo | Переменная, содержащая порядковый номер инициализации 1 |
| DataCommand | Указывает на то. какая из команд ProcessDafa или *FailSafeDala* отправлена в состоянии Данных. Ижциализируется с помощью *FailSafeDala* при включении питания |
| BytesToBeSent | Если несколько PDU блоков безопасности должно быть отправлено в состоянии сеанса, соединения или параметров, то эта переменная указывает на то. скогъко еще октетов  должно быть отправлено (инициализируется значением 0 при включении питания) |
| Connections | В состоянии соединения Connection! D принимается ведущим устройством FSoE (иници­ ализируется значением 0 при включении питания) |
| ConnectionData | В состоянии соединения ConnectionData принимается ведущим устройством FSoE (ини­ циализируется значением Опри включении питания) |
| SlaveAddress | Адрес ведомого устройства FSoE инициализируется посредством локального интер­ фейса при включении питания (как правило, внешний переключатель адресов) |
| SafePara | SafePara принимаются ведущим устройством FSoE 8 состоянии параметров и инициа­ лизируется в зависимости от устройства.  SafePara.Watchdog: сторожевой таймер FSoE (инициализируется значением 0 при вклю­  чении питания) |
| ExpectedSafeParaSize | Указывает на длину ожидаемого SafePara |
| SafeOutputs | Содержит значения процесса выводов безопасности, полученные ведущим устройством FSoE. Инициализируется с помощью  FS.VALUE (Fail-safe Data = 0) при включении питания |
| Safelnputs | Содержит значения процесса вводов безопасности, отправленных ведущему устройству FSoE. Инициализируется с помощью FS.VALUE (Fail-safe Data = 0) при включении питания |
| CommFaultReason | Указывает на код ошибки в случае события возникновения коммуникационной ошибки |

* + - 1. Макрокоманды

Определенные функциональные возможности объединяются в макрокоманды для того, чтобы сохранять прозрачность таблиц состояний.

В таблице 38 перечислены макрокоманды из таблицы состояний ведомого устройства FSoE.

Таблица 38 — Макрокоманды в таблице состояний ведомого устройства FSoE

Описание

IS\_CRC\_CORRECT{frame, Эта макрокоманда проверяет, корректны ли CRC коды полученного PDU lastCrc. seqNo. oldCrc. bNew) ведущего устройства безопасности.

Параметры:

Frame — принятый кадр:

lastCrc — CRC\_0 последнего отправленного PDU ведомого устройства безопасности, включенное в вычисления CRC для принятого PDU;

seqNo — указатель на порядковый номер ведущего устройства, используемый в вычислении CRC для принятого кадра. Возвращается приращенный (возможно неоднократно) seqNo:

otdCrc — ухазатель на CRC\_0 последнего принятого PDU ведущего устройства безопасности. Возвращается CRC\_0 полученной телеграммы:

bNew — если bNew = TRUE и oldCrc равняется вычисленному сгс. то вычисле­ ние CRC повторяется с приращенным seqNo до тех пор. пока вычисленный сгс не перестанет быть равным oldCrc (процедура выполняется согласно 7.1.3.4)

47

#### ГОСТ Р МЭК 61764-3-12—2016

*Окончание таблицы 38*

|  |  |
| --- | --- |
| Функция | Описание |
| UPDATE\_BYTES\_TO\_BE\_SENT  (bytesSent) | Данная макрокоманда проверят, сколько еще октетов в состояниях сеанса, со­ единения и параметров необходимо отправить перед изменением состояния. Параметры:  safePara: указатель на полученные SafePara |
| IS\_SAFE\_PARA\_CORRECT  (safePara) | Данная макрокоманда проверяет корректны ли SafePara полученные в состоянии параметров.  Параметры:  Frame — принятый кадр:  expectedData — ссылка на ожидаемые данные: bytesSent — число отправленных октетов |
| STORE\_DATA(dSl. src) | Данная макрокоманда хранит полученные данные безопасности PDU безопасности.  Параметры:  Dst — указатвгъ на целевой адрес: See — указатвгъ на адрес источника |
| GET\_PARA\_FAULT () | Данная макрокоманда возвращает код ошибки, если SafePara не действителен |
| START.WD (watchdog) | Данная макрокоманда перезапускает (сбрасывает) сторожевой таймер и запускает контролирующий таймер с указанным сторожевым таймером. Параметры:  Watchdog — время контроля (мониторинга) в мс |
| STOP.WDO | Данная макрокоманда останавливает контролирующий таймер и сбрасывает сторожевой таймер |
| ADR | Данная макрокоманда генерирует ссылку (указатель) на переменную |

7.5.2 Состояние сброса

7.5.2.1 Событие получения кадра

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие |  | Соедующее состояние |
| RESET OK | Frame.Command - | LastCrc :- Frame.CrcO; | Session (Ce- |
| *Session* | Sessionld CREATE SESSION ID|); | анс) |
| AMD | SendFrame(Session, |  |
| IS CRC CORRECT(Frame, | ADR(Sessionld), LastCrc, |  |
| LastCrc, | 0, |  |
| ADR(MasterSeqNo), | ADR(SlaveSeqMo), |  |
| ADR(OldMa s te rC г с 1, | ADR(OldSlaveCrc), FALSE); |  |
| FALSE) - TRUE | LastCrc SendFrame.CrcO;  BytesToBeSent |  |
|  | UPDATE\_BYTES\_TO\_BE\_SENT(2); |  |
| RESET FA1L1 | Frame.Command - | LastCrc :• 0 | Reset (Сброс) |
| *Session* | OldKasterCrc 0; |
| AND | OldSlaveCrc :• 0; |
| IS CRC CORRECT(Frame, | KasterSeqMo 1; |
| LastCrc, | SlaveSeqMo :- 1; |
| ADR(MasterSeqNo), | DataCommand :- *FailSaCeData;* |
| ADR(OldMaSteiCrc), | CommFaultReason :• INVALID CRC; |
| FALSE) - FALSE | SendFrame(Reset, ADR(CommFaultReason), LastCrc, 0,  ADR(SlaveSeqMo),  ADR(OldSlaveCrc), FALSE); SlaveSeqMo :» 1; |

46

#### ГОСТ Р МЭК 61784-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие |  | Следующее состояние |
| RESET\_STAY1 | Frame.Command - R*eset* | LastCrc :• 0  OldMasterCce 0;  OldSlaveCrc 0;  MasterSeaNo :- 1;  SlaveSeqNo 1;  InitSeqNo :•\* 1;  OataCommand :> FailSafeOata; CommFaultReason 0; SendFrame(Reset, ADR(CommFaultReason», LastCrc, 0,  ADR(SlaveSeqNo»,  AOR(OldSlaveCrc», FALSE»; SlaveSeqNo :- l; | Reset {Сброс) |
| RESET\_FAI12 | <Fcame.Command - | LastCrc :• 0 | Reset {Сброс) |
| *Connection* | OldMasterCrc :• 0; |
| OR | OldSlaveCrc :• 0; |
| Frame.Command - | MaStetSeqNO 1; |
| Parameter | SlaveSeqNo :• 1,- |
| OR | OataCommand :- *FeilSefeDete;* |
| Frame.Command \* | CommFaultReason :• INVALID CMO; |
| ProceasOeta | SendFrame(Reset, |
| OR | AOR(CommFaultReaaon>, LastCrc, |
| Frame.Command - | 0, |
| FailSafeDatd) | AOR(SlaveSeqNo», ADR(01dSlaveCrc», FALSE»;  SlaveSeqNo 1; |
| RESET\_FAIL3 | {Frame.Command о | LastCrc :• 0 | Reset {Сброс) |
| *Reset* | OldMasterCrc :• 0; |
| AND | OldSlaveCrc :• 0; |
| Frame.Command <> | MasterSeqNo 1; |
| *Session* | SlaveSeqNo :• 1; |
| AND | OataCommand :- *FeilSefeDete;* |
| Frame.Command <> | CommFaultReason UNKNOWN CMO; |
| *Connection* | SendFrame(Reset, |
| ANO | AOR(CommFaultReason|, LastCrc, |
| Frame.Command <> | 0, |
| Parameter | AOR(SlaveSeqNo», |
| AND | AOR(OldSlaveCrc», FALSE»; |
| Frame.Command <> ProceasOeta  AND  Frame.Command <> FailSafeOata) | SlaveSeqNo :• 1; |

7.5.2 2 Событие истекшего сторожевого таймера

Невозможно в данном состоянии, так как сторожевой таймер еще не был запущен.

49

#### ГОСТ Р МЭК 61764.3-12—2016

7.5.2.3 Событие сброса соединения

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| RESET.START | Frame.Command - | LastCrc D | Reset {Сброс) |
| *Session* | OldMasterCrc 0; |
| AND | OldSlaveCrc :• 0; |
| IS CRC CORRECT(Frame, | MasterSeqNo :- 1; |
| LastCrc, | SlaveSeqNo 1; |
| ADR(MasteiSeqNo), | InltSeqNo I; |
| ADR(OldMastexCrc), | DataCoranand :« FallSafeData; |
| FALSE) - TRUE | CommFaultReason 0; SendFrame(Reset, ADR(CommFaultReason), LastCrc, o,  ADR(SlaveSeqNo),  ADR(OldSlaveCrc), FALSE); SlaveSeqNo 1; |

7.S.2.4 Событие Команды Set Data

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| RESET\_STAY2 |  | DataCommand DataCmd; | Reset (Сброс) |

* + 1. Состояние сеанса
       1. Событие получения кадра

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| SESSION OK |  | STORE DATA(ADR(ConnectionData), | Connection |
| ADR(Frame.SafeData)); Connectionld  :» Frame.Connld; LastCrc Frame. CrcO; SendFrame(Connection, ADR(Frame.SafeData), LastCrc,  Connectionld, ADR(SlaveSeqNo), ADR(OldSLaveCrc), TRUE); LastCrc :• SendFrame.CrcO;  BytesToBeSent :• | (Соединение) |
| UPDATE\_BYTES\_TO\_BE\_SENT(4); |  |
| SESSION FAII1 | Frame.Command - | LastCrc :• 0; | Reset (Сброс) |
| *Connection* | OldKasterCrc :• 0; |
| AND | OldSlaveCrc :• 0; |
| BytesToBeSent • 0 AND | MasterSeqNo :- 1; |
| Frame.Connld <> 0 AND | SlaveSeqNo 1; |
| IS CRC CORRECT(Frame, | DataCommand :» *FallSafeDsts;* |
| LastCrc, | CommFaultReason INVALID CRC; |
| AOR(MasterSeqNo), | SendFrame(Reset, |
| ADR(OldMastexCrc), | ADR(CommFaultReason), LastCrc, |
| TRUE) • FALSE | 0,  ADR(SlaveSeqNo), ADR(OldSlaveCrc), FALSE);  SlaveSeqNo 1; |

50

#### ГОСТ Р МЭК 61784-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| SESSION\_FAIL2 | Frame.Command -  *Connection*  AND  BytesToBeSent - 0 AND Frame.Connld • 0 | LastCrc :• 0; OldMasterCrc :• Or OldSlaveCrc Or MasterSeqNo :• 1,- SlaveSeqNo lr  DataCommand *FailSafeData;* CommFaultReason INVALID\_CONNID; SendFrame(Reset, AOR(CommFaultReason>, LastCrc,  0,  ADR(SlaveSeqNo), AOR(OldSlaveCrc), FALSE)r  SlaveSeqNo lr | Reset {Сброс) |
| SESS!ON\_FAIL3 | Frame.Command - Connection  AND  BytesToBeSent <> 0 | LastCrc :- Or OldMasterCrc :• Or OldSlaveCrc Or MasterSeqNo :• 1;  SlaveSeqNo :•» 1 DataCommand :• *FailSafeData;*  CommFaultReason INVALID\_CMOr SendFrame(Reset, ADR(CommFaultReason), LastCrc, 0,  AOR(SlaveSeqNo), ADR(OldSlaveCrc),  FALSE)r  SlaveSeqNo lr | Reset (Сброс) |
| SESSION.STAY1 | Frame.Command -  *Session*  AND  SyteaToBeSent <> 0 AND IS\_CRC\_ CORRECT<Frame,  LastCrc, ADR(MasterSeqNo), ADR(OldMasterCre),  TRUE) - TRUE | LastCrc Frame.CrcOr SendFrame(Session, ADR(SessionId|2- BytesToBeSent]), LastCrc, 0,  ADR(SlaveSeqNo), ADR(OldSlaveCrc), TRUE)r  LastCrc SendFrame.CrcO; BytesToBeSent UPOATE\_BYTES\_TO\_ BE\_SENT(  BytesToBeSent)r | Session (Се­ анс) |
| SESSION.STAY2 | Frame.Command - Session  AND  IS CRC CORRECT(Frame,  o,~  ADR ( XnitSeqNo), ADR <01dMasterCrc), FALSE) - TRUE | LastCrc :• Frame.CrcOr MasterSeqNo InitSeqNor XnitSeqNo lr  SlaveSeqNo :- lr  DataCommand *FailSafeData;* Sessionld CREATE\_SESSION\_ID()r SendFrame(Session, AOR(SessionlD), LastCrc,  o,  AOR(SlaveSeqNo), ADR(OldSlaveCrc), FALSE)r  LastCrc SendFrame.CrcOr BytesToBeSent s- UPDATE\_BYTES\_TO\_ BE\_SENT(2)r | Session (Се­ анс) |

51

#### ГОСТ Р МЭК 61784-3-12»2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| SESSION FAIL4a| | Frame.Command - | LastCrc :• 0; | Reset (Сброс) |
| *Session* | OldMasterCrc 0; |
| AND | OldSlaveCrc 0; |
| IS CRC CORRECT(Frame, | MasterSeqNc :- l; |
| LastCrc, | SlaveSeqNo 1; |
| ADR(MasterSeqNo), | InitSeqNo 1; |
| ADR(OldMasterCrc!, | DataCocmand *FallSsieData;* |
| TRUE} - FALSE | CommFaultReason :• INVALID CRC; |
| AND IS CRC | SendFrame{Reset, |
| CORRECT(Fcame, 0, | ADR{CommFaultReason), LastCrc, |
| ADR(InitSeqNo!, | 0, |
| AOR(OldMasterCrc), | ADR{SlaveSeqNo), ADR{OldSlaveCrc), |
| FALSE» • FALSE | FALSE);  SlaveSeqNo 1; |
| SESSION FAILS»\* | Frame.Command - | LastCrc :• 0; | Reset (Сброс) |
| *Session* | OldMasterCrc 0; |
| AND | OldSlaveCrc :• 0; |
| BytesToBeSent • 0 AND | MasterSeqNo :«• 1; |
| IS CRC CORRECT(Frame, | SlaveSeqNo :• 1; |
| LastCrc, | DataCommand :» *FailSaleData;* |
| ADR(MasterSeqNo), | CommFaultReason :• INVALID CMD; |
| ADR(01dMasterCrc!, | SendFrame(Reset, |
| TRUE! - TRUE | ADR(CommFaultReason), LastCrc, |
| ANO IS CRC | 0, |
| CORRECT(Frame, 0, | ADR(SlaveSeqNo), ADR(OldSlaveCrc), |
| ADR(In i t SeqNo|, | FALSE); |
| ADR(OldMasterCrc), FALSE! \* FALSE | SlaveSeqNo 1; |
| SESSION | Frame.Coeeand - | LastCrc :«• 0; | Reset (Сброс) |
| RESET1 | *Reset* ANO IS CRC | OldMasterCrc D; |
|  | CORRECT(Frame, 0, | OldSlaveCrc :«• 0; |
|  | ADR(InitSeqNo), | MasterSeqNo :• 1; |
|  | ADR(OldMasterCrc), | SlaveSeqNo :- l; |
|  | FALSE) • TRUE | InitSeqNo l;  DataCommand *FailSaleData;* CommFaultReason :• 0; SendFrame(Reset, ADR(CommFaultReason), LastCrc, 0,  ADR(SlaveSeqNo), ADR(OldSlaveCrc),  FALSE); |
| SESSION FAIL6 | Frame.Command - | LastCrc :« 0; | Reset (Сброс) |
| Reset AND IS CRC | OldMasterCrc 0; |
| CORRECT (Frame, 0, | OldSlaveCrc 0; |
| ADR(InitSeqNo), | MasterSeqNo 1; |
| ADR(OldMasterCrc), | SlaveSeqNo 1,- |
| FALSE) - FALSE | InitSeqNo :\*• 1; DataCommand :• *FallSefeDais;*  CommFaultReason :- INVALID\_CRC; SendFrame{Reset, ADR(CommFaultReason), LastCrc, 0,  ADR(SlaveSeqNo), ADR(OldSlaveCrc),  FALSE) ;  SlaveSeqNo 1; |

52

#### ГОСТ Р МЭК 61784-3.12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| SESSION\_FAIL7 | Frame.Command - | LastCrc :• 0; | Reset {Сброс) |
| Parameter | OldMasterCrc 0,- |
| OR | OldSlaveCrc 0; |
| Frame.Command - | MasterSeqNo :• 1; |
| *ProcessDaia* | SlaveSeqNo :• 1; |
| CR | DataCommand *FsilSafeDeta;* |
| Frame.Command - | CommFaultReason INVALID CMD; |
| FdilSdfeDd ta | SendFrdine (Reset,  AOR(CofltmFaultReason>, LastCrc,  ,  ADR(SlaveSeqNo}, ADR(OldSlaveCrc}, FALSE};  SlaveSeqNo 1; |
| SESSION\_FAIL8 | {Frame.Command <> | LastCrc :- 0; | Reset (Сброс) |
| *Reset* | OldMasterCrc 0; |
| AND | OldSlaveCrc 0; |
| Frame.Command <> | MasterSeqNo :• 1; |
| *Session* | SlaveSeqNo :« 1; |
| AND | DataCommand :• *FailSafeData;* |
| Frame.Command <> | CommFaultReason L'NKNOHN CMD; |
| Connection | SendFrame(Reset, |
| AND | ADR(CommFaultReason}, LastCrc, |
| Frame.Command <> | , |
| *Parameter* | ADR(SlaveSeqNo}, ADR(OldSlaveCrc}, |
| AND | FALSE}; |
| Frame.Command <>  *ProcessData*  AND  Frame.Command <>  *FailSafeData)* | SlaveSeqNo l; |
| а> Два состояния SESS!ON\_FAIL4 и SESSlON\_FAIL5 являются единственными состояниями, в которых должны вычисляться две проверки CRC. Единственное отличие заключается в другом CommFaultReason. т. е. только диагностическая информация, не информация важная для безопасности. Разрешается сокращать эти состояния до одного: в этом состоянии только условие \*IS\_CRC\_CORRECT(Frame. 0. ADR(lnitSeqNo). ADR(Ok)- | | | |
| MasterCrc). FALSE) - | FALSE» должно проверяться с помощью CommFaultReason := INVAL1D\_CRC. | |  |

* + - 1. Событие истекшего сторожевого таймера

Невозможно в данном состоянии, так как сторожевой таймер еще не был запущен.

* + - 1. Событие сброса соединения

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| SESSION RESET2 |  | LastCrc 0;  OldMasterCrc 0;  OldSlaveCrc 0;  MasterSeqNo :- 1; SlaveSeqNo :- l; DataCommand FailSafeData; CommFaultReason 0; SendFrame{Reset,  ADR{CommFaultReason), LastCrc, t>,  ADR(SlaveSeqNo), ADR(OldSlaveCrc),  FALSE);  SlaveSeqNo :• l; | Reset (Сброс) |

53

#### ГОСТ Р МЭК 61764.3-12—2016

* + - 1. Событие Команды Set Data

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состоимне |
| SESStON\_STAY3 |  | DataCommand DataCmd; | Session (Сеанс) |

* + 1. Состояние соединения 7.5.4,1 Событие получения кадра

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| CONN OK | Frame.Command - Parameter | STORE DATA(ADR{SatePara), | Parameter |
| AND  BytesToBeSent - 0 | ADR(Fcame.SafeData)); | (Параметры) |
| AND | LastCrc Frame.CrcO; |  |
| Frame.Connld - | SendFrame{Parameter, |  |
| Connectionld AND  ConnectionData. | ADR(Frame.SaieData), |  |
| Connectionld | LastCrc, Connectionld, |  |
| - | ADR(SlaveSeqNo), |  |
| Connectionld | ADR(OldSiaveCrc), TRUE); |  |
| AND | LastCrc :• SendFrame.CrcO; |  |
| ConnectionData. | BytesToBeSent UPDATE BYTES |  |
| SlaveAddreaa | TO BE SENT< |  |
| - | ExpectedSafeParaSite); |  |
| SlaveAddreaa AND IS\_CRC\_CORRECT(Frame,  LastCrc, ADR(MdsterSeqNo), ADR(OldMasterCrC), TRUE) - TRUE |  |  |
| CONN FAIL1 | Frame.Command - Parameter | LastCrc 0; | Reset (Сброс) |
| AND | OldMasterCrC 0; |
| BytesToBeSent - 0 | OldSlaveCrc 0; |
| AND | MasterSeqNo 1; |
| Frame.Connld - | SlaveSeqNo 1; |
| Connectionld AND | DataCommand raiJSefeOata; |
| ConnectionData. | CommFaulLReason INVALID CRC; |
| Connectionld | SendFrame(Reset, |
| Connectionld | ADR(CommFaultReason), |
| AND | LastCrc, 0, |
| ConnectionData. | ADR(SlaveSeqNo), |
| SlaveAddreaa | ADR(OldSlaveCrc), FALSE); |
| - | SlaveSeqNo 1; |
| SlaveAddreaa AND  IS\_CRC\_CORRECT(Frame,  LastCrc, ADR(MasterSeqNo),  ADR(OldMasterCrC), TRUE) - FALSE |  |

54

#### ГОСТ Р МЭК 61784-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| CONN FAIL2 | Frame.Coecnand - Parameter | LastCrc 0; | Reset (Сброс) |
| AND | OldMasterCrc 0; |
| BytesToBeSent - 0 | OldSieveCrc 0; |
| AND | MasterSeqNo 1; |
| Frame.ConnXd - | SlaveSeqNo 1; |
| Connectionld AMD | DataCoetnand *FailSaSeData;* |
| ConnectionData. | CoatnFaultReeson INVALID |
| Connectionld | ADDR; |
| - | SendFrame(Reset, |
| Connectionld  AMD | ADR(CoetnFaultReason), |
| ConnectionData. | LastCrc, 0, |
| SlaveAddxess | ADR(SlaveSeqNo), |
| <> | ADR(OidSlaveCre), FALSE); |
| SlaveAddress | SlaveSeqNo 1; |
| CONN FAIL3 | Frame.Coarnand - Parameter | LastCrc 0,- | Reset (Сброс) |
| AMD | OldMasterCrc 0; |
| BytesToBeSent - 0 | OldSlaveCrc :« 0,- |
| MID | MasterSeqNo 1; |
| (Frame.ConnZd <> | SlaveSeqNo 1; |
| ConnectionId | DataCoetnand FaiiSafeData; |
| OR | ComnFaultReason :\*> INVALID\_  CONMID; |
| ConnectionData.  Connection Id | SendFrame(Reset, |
| <> | ADR(ComnFaultReason), |
| Connectionid) | LastCrc, 0, ADR(SlaveSeqNo), ADR(OldSlaveCrc), FALSE); SlaveSeqNo :• 1; |
| CONN FASL4 | Frame.Coatnand - Parameter | LastCrc 0,- | Reset (Сброс) |
| AMD | OldMasterCrc 0; |
| BytesToBeSent о 0 | OldSlaveCrc :« 0,-  MasterSeqNo 1;  SlaveSeqNo 1; DataCoetnand FaiiSafeData;  ComnFaultReason 2NVALXD\_CMD; SendFrame(Reset, |
|  | ADR(CoecnFaultReason), LastCrc, 0, ADR(SlaveSeqNo), ADR(OldSlaveCrc), FALSE),-  SlaveSeqNo 1; |

55

#### ГОСТ Р МЭК 61784-3-12»2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Услоеие | Дейстеие | Следующее состояние |
| CONN STAY1 | Frame.Command - Connection | STORE DATA{ | Connection |
| AND | ADR(Connect ion(4- | (Соединение) |
| BytesToBeSent <> 0 AND | BytesToBeSent]), |  |
| Frame.Conoid » | ADR(Frame.SafeData)); |  |
| Connectionld AND | LastCrc Frame.CrcO; |  |
| IS\_CRC\_CORRECT(Frame,  LastCrc, | SendFrame(Connection, |  |
| ADR(MdsterSeqNo), | ADR(Frame.SafeData), |  |
| ADR(OldMasterCrc), | LastCrc, Connectionld, |  |
| TRUE) - TRUE | ADR(SiaveSeqNo),  ADR (OldSlaveCrc), TRUE) ; LastCrc SendFrame.CrcO; BytesToBeSent UPDATE\_BYTES\_ |  |
|  | TO BE SENT< |  |
|  | BytesToBeSent); |  |
| CONN FAIL5 | Frame.Command - connection | LastCrc 0; | Reset (Сброс) |
| AND | OldMasterCrc 0; |
| BytesToBeSent <> 0 AND | OldSlaveCrc 0; |
| Frame.Connld - | MasterSeqNo 1; |
| Connectionld AND | SiaveSeqNo :- 1; |
| IS CRC CORRECT(Frame, | DataCommand *FoilSofeDato;* |
| LastCrc, | CommFauItReason INVALID CRC; |
| AOR(MasterSeqNo), | SendFrame(Reset, |
| ADR(OldMasterCre), TRUE) - FALSE | ADR(CommFaultReason), LastCrc, 0, ADR(SlaveSeqNo), ADR(OldSlaveCrc), FALSE); SiaveSeqNo 1; |
| CONN FAIL6 | Frame.Command - *Connection* | LastCrc 0; | Reset (Сброс) |
| AND | OldMasterCrc 0; |
| BytesToBeSent <> 0 | OldSlaveCrc 0; |
| AND | MasterSeqNo 1; |
| Frame.Connld <> | SiaveSeqNo :•» 1; |
| Connectionld | DataCommand FaiiSafeOaea; CommFauItReason INVALID\_ CONID;  SendFrame(Reset, |
|  | ADR(CommFaultReason), LastCrc, 0, ADR(SlaveSeqNo), ADR(OldSlaveCrc), FALSE); SiaveSeqNo 1; |

56

#### ГОСТ Р МЭК 61784-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| CONN.FAIL7 | Frame.Cotroand - *Connection*  AND  BytesToBeSent - 0 | LastCrc 0;  OldMasterCrc :• 0;  OldSlaveCrC 0;  MasterSeqNo 1;  SlaveSeqNo 1; DataCoenand *FailSaSeData;*  CoattFaultReason INVALZD\_CMD; SendFrame(Reset,  ADR(ComnFaultReason), LastCrc, 0, ADR(SlaveSeqHo), ADR(OldSlaveCzC), FALSE);  SlaveSeqNo 1; | Reset (Сброс) |
| CONN.RESET1 | Frame.Comnand - *Reset* AND IS\_CRC\_CORREC?(Frame, 0,  ADR(InltSeqNo), ADR(OldMasterCrc), FALSE) - TRUE | LastCrc 0;  OldMasterCrc 0;  OldSlaveCrC 0,-  MasterSeqNo 1,-  SlaveSeqNo 1;  InitSeqNo 1,-  DataCoetnand *FailSafeData;* ComnFaultReason :« 0; SendFrame(Reset,  ADR(CoemFaultReason), LastCrc, 0,  ADR (SlaveSeqNo), ADR(OldSlaveCrC), FALSE);  SlaveSeqNo 1,- | Reset (Сброс) |
| CONN.FA5L8 | Frame.Coomand - *Reset* AND IS\_CRC\_CORRECT(Frame, 0,  ADR(InltSeqNo), ADR(OldMasterCrc), FALSE) - FALSE | LastCrc 0;  OldMasterCrc 0;  OldSlaveCrC 0,-  MasterSeqNo 1;  SlaveSeqNo 1;  InitSeqNo 1,-  DataCoetnand *FailSafeData;* ComnFaultReason :«• 2NVALXD\_CRC; SendFrame(Reset,  ADR (CooeiFaultReascn), LastCrc, 0, ADR(SlaveSeqNo), ADR(OldSlaveCrC), FALSE);  SlaveSeqNo 1; | Reset (Сброс) |

57

#### ГОСТ Р МЭК 61764-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| CONN.RESET2 | Frame.Command - Session AND IS CRC CORRECT(Frame, 0,  ADRflnitSeqNo), ADR<01dMasterCrc), FALSE) - TRUE | LastCrc Frame.CrcO; MasterSeqNo 2;  InitSeqNo 1; SiaveSeqNo I;  DataCommand *FAllSafeD&ta;* Sessionld CREATE SESSION ID<);  SendFrame(Session, ADR(SessionlD), LastCrc, 0,  ADR(SiaveSeqNo),  AOR(OidSiaveCrc), FALSE); LastCrc SendFrame.CrcO; BytesToBeSent UPDATE\_BYTES\_ T0\_BE\_SENT<2>; | Session (Се­ анс) |
| CONN\_FAIL9 | Frame.Command - Session AND IS CRC CORRECT(Frame, 0,  ADR(InitSeqNo), ADR(OldMasterCcc), FALSE) - FALSE | LastCrc 0;  OldMasterCcc 0;  OldSiaveCrc 0;  MasterSeqNo 1;  SiaveSeqNo *:•* 1;  InitSeqNo :\*\* 1; DataCommand FaiiSafeOaea;  CommFaultReason :- INVALIO\_CRC; SendFrame(Reset, | Reset (Сброс) |
| ADR(CommFAuitReason), LastCrc, 0, ADR(SiaveSeqNo), ADR(OldSlaveCrc), FALSE); SiaveSeqNo *:•* 1; |
| CONN.FAIL10 | Frame.Command \_  ProoessOece OR  Frame.Command - faiiSafeOaea | LastCrc 0;  OldMasterCcc 0;  OldSiaveCrc 0;  MasterSeqNo 1;  SiaveSeqNo *:•* 1; DataCommand *FailSAfeDatA;*  CommFaultReason INVALID\_CMD; SendFrame(Reset, | Reset (Сброс) |
| ADR(CommFaultReason), LastCrc,  o, ADR(SlaveSeqNo),  ADR(OldSlaveCrc), FALSE); SiaveSeqNo :- 1; |

58

#### ГОСТ Р МЭК 61784-3.12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состоя и ие |
| CONN FAIL11 | (Frame.Command о *Reset* | LastCzc :\*• 0; | Reset (Сброс) |
| AND | OldMasterCrc 0; |
| Frame.Coereiand о *Session* | OldSlaveCre :• 0; |
| AND | MasterSeqNo 1; |
| Frame.Coereiand <> | SlaveSeqNo 1; |
| *Connection* | DataCoanand *FaiiSafeData;* |
| AND | CoawFauitReason unkn ow n CMD; |
| Frame.Coercnand <> Parameter | SendFrame(Reset, |
| AND |  |
| Frame.Comnand <> | ADR(CoeeiFaultReason), |
| *PeocessDa*ca | LastCrc, |
| AND | 0, |
| Frame.Coerekand о | ADR(SlaveSeqNo), |
| *FailSafeDsta]* | ADR(OidSlaveCre), FALSE); |
|  | SlaveSeqNC 1; |

* + - 1. Событие истекшего сторожевого таймера

Невозможно в данном состоянии, так как сторожевой таймер еще не был запущен.

* + - 1. Событие сброса соединения

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| CONN.RESET3 |  | LastCrc :«• 0;  OldMasterCxc 0;  OldSlaveCre :» 0;  MasterSeqNo :- 1;  SlaveSeqNo 1; DataCommand FallSaleData; CommFeultReason :\*• 0; SendFrame(Reset,  ADR(CommFeultReason), LastCrc, 0,  ADR(SlaveSeqNo), ADR(OldSlaveCre),  FALSE);  SlaveSeqNo :• 1; | Reset (Сброс) |

* + - 1. Событие Команды Set Data

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| CONN.STAY2 |  | DataCommand :- DataCmd; | Connection (Со\* единение) |

* + 1. Состояние параметров
       1. Событие получения кадра

59

#### ГОСТ Р МЭК 61784-3-12»2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условия | Действие | Следующее состояние |
| PARA.OKI | Frame.Command - | Watchdog :- SafePara.Watchdog; | Data (Данные) |
| *PeocesaData* | SafeOutputa :• Frame.SaleData; |
| AND | LaatCrc :« Frame.CrcO; |
| BytesToBeSent - 0 AND | SendFrame(DacaComaand, |
| Frame.Connld - | ADR(Safelnputa), LaatCrc, |
| Connectionld AND | Connectionld, ADR(SIaveSeqNo), |
| IS SAFE PARA CORRECT( | ADR(OldSlaveCrc), TRUE); |
| SafePara) - TRUE AND | LaatCrc :• SendFrame.CrcO; START |
| IS\_CRC\_CORRECT(Frame,  LaatCrc, ADR(MaaterSeqNo), ADR(OidMaaterCrc), TRUE) - TRUE | WD(Watchdog); |
| PARAJDK2 | Frame.Command - | Watchdog SafePara.Watchdog; | Data (Данные) |
| FaiJSafeOata AND | SafeOutputa :- FS VALUE; LaatCrc |
| ByteaToBeSent - 0 | Frame.CrcO; SendFrame(DataCommand, |
| AND | ADR(Safelnputa), LaatCrc, |
| Frame.Connld - | Connectionld, ADR(SIaveSeqNo), |
| Connectionld AND | ADR(OldSlaveCrc), TRUE); |
| IS SAFE PARA CORRECT( | LaatCrc :• SendFrame.CrcO; START |
| SafePara) - TRUE AND IS\_CRC\_CORRECT(Frame,  LaatCrc, ADR(MaaterSeqNo), ADR (OidMaaterCrc),  TRUE) - TRUE | WD(Watchdog); |
| PARA FAIL1 | (Frame.Command - | LaatCrc 0; | Reset (Сброс) |
| ProcesaDaca | OidMaaterCrc :• 0; |
| OR | OldSlaveCrc :• 0; |
| Frame.Command - | MaaterSeqNo :- I; |
| FaiJSafeoata) a nd | SlaveSeqNo :• 1; |
| ByteaToBeSent - 0 AND | DataCommand :- FaiiSafeOata; |
| Frame.Connld » | CommFaultReaaon :• INVALID CRC; |
| Connectionld AND | SendFrame(Reset, |
| IS SAFE PARA CORRECT( | ADR(CommFaultReaaon), LaatCrc, |
| SafePara) - TRUE AND | 0, |
| IS CRC CORRECT(Frame, | ADR(SlaveSeqNo), ADR(OldSlaveCrc), |
| LaatCrc, | FALSE); |
| ADR(MaaterSeqNo), ADR(OidMaaterCrc),  TRUE) - FALSE | SlaveSeqNo :• 1; |
| PARA.FAIL2 | (Frame.Command - | LaatCrc :• 0; | Reset (Сброс) |
| Processesta | OidMaaterCrc :• 0; |
| OR | OldSlaveCrc 0; |
| Frame.Command - | MaaterSeqNo :- 1; |
| FdiJSafeOdta) AND | SlaveSeqNo 1; |
| BytesToBeSent - 0 AND | DataCommand FaiiSafeOata; |
| Frame.Connld - | CommFaultReaaon GET PARA FAULT; |
| Conneotionld AND | SendFrame(Reset, |
| IS SAFE PARA CORRECT( | ADR(CommFaultReaaon), LaatCrc, |
| SafePara) - FALSE | 0,  ADR(SlaveSeqNo), ADR(OldSlaveCrc), FALSE);  SlaveSeqNo :• 1; |

60

#### ГОСТ Р МЭК 61784-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| PARA\_FA!L3 | (Frame.Command •  *ProcessData*  OR  Frame.Command • *FailSafaDat\*}* AND BytesToBeSent - 0 ANO Frame.Connld <> Connectionld | LastCrc :•» 0;  OldMasterCrc 0;  OldSlaveCEC 0;  MasterSeqNo :«■ 1;  SlaveSeqNo :• 1; DataCommand :w *FailSafeData;*  CommFaultReason INVALID\_CONNID;  SendFrame(Reset, AOR(CommFaultReason», LastCrc, 0,  ADR(SlaveSeqNo», A0R(01dSlaveCrc»,  FALSE»;  SlaveSeqNo :• 1; | Reset (Сброс) |
| PARA.FA1L4 | (Frame.Command -  *PzocessData*  OR  Frame.Command - *FailSafeData)* ANO BytesToBeSent <> 0 | LastCrc :«■ 0;  OldMasterCrc 0;  OldSlaveCrc :«■ 0;  MasterSeqNo :- 1;  SlaveSeqNo :- 1; DataCommand :- *FailSaiaData;*  CommFaul tReason :«■ 1NVAL1D\_CMD; SendFrame(Reset, ADRtCommFaultReason», LastCrc,  0,  AOR(SlaveSeqNo», AOR(OldSlaveCrc», FALSE»;  SlaveSeqNo :• 1,- | Reset (Сброс) |
| PARA.STAY1 | Frame.Command - Parameter  ANO  BytesToBeSent <> 0 ANO Frame.Connld - Connectionld ANO 3S\_CRC\_CORRECT(Frame,  LastCrc, AOR(MasterSeqNo», ADR(OldMasterCrc), TRUE» - TRUE | STORE\_DATA(  AOR(SatePara| ExpectedSaCeParaSize- BytesToBeSent)», ADR(Frame.  SateOata»»;  LastCrc Frame.CrcO; SendFrame(Parameter, AOR(Frame.SdCeOata», LastCrc, Connectionld, AOR(SlaveSeqNo», ADR(OldSlaveCrc», TRUE»; LastCrc :• SendFrame.CrcO;  BytesToBeSent UPDATE\_BYTES\_TO\_BE\_ SENT(  BytesToBeSent»; | Parameter (Парамегр) |
| PARA.FAIL5 | Frame.Command - Parameter  ANO  BytesToBeSent <> 0 ANO Frame.Connld • Connectionld ANO 1S\_CRC\_CORRECT(Frame,  LastCrc, ADR(MasterSeqNo», ADR(01dMasterCrc», TRUE» - FALSE | LastCrc :• 0;  OldMasterCrc :• 0;  OldSlaveCrc :• 0;  MasterSeqNo :- 1;  SlaveSeqNo :- 1; DataCommand *FailSafeData;*  CommFaultReason :• 1NVALID\_CRC; SendFrame(Reset*,* ADR(CommFaultReason», LastCrc,  0,  AOR(SlaveSeqNo», AOR(OldSlaveCrc», FALSE»;  SlaveSeqNo :- 1; | Reset (Сброс) |

61

#### ГОСТ Р МЭК 61784-3-12»2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| PARA.FAIL6 | Frame.Command - Parameter  AND  BytesToBeSent <> 0 AND Frame.Connld <> Connections | LastCrc 0;  OldMastecCro 0;  OldSlaveCrc 0;  MasterSeqNo 1;  SlaveSeqNO 1;  DataCommand :» FaiJSafeData;  CoemFa u11 Rea son Г NVALID\_CONN ID; SendFrame{Reset, ADR{CommFaultReason), LastCrc,  0,  ADR(SlaveSeqNO), ADR(OldSlaveCrc), FALSE);  SlaveSeqNO 1; | Reset {Сброс) |
| PARA.FAIL7 | Frame.Command - Parameter  AND  BytesToBeSent - 0 | LastCrc :• 0;  OldMasterCrc 0;  OldSlaveCrc :• 0;  MasterSeqNo :- 1;  SlaveSeqNO 1;  DataCommand :• FaiiSafeOaca; CommFaultReason INVALID\_CMD; SendFrame(Reset, ADR(CommFaultReason), LastCrc,  0,  ADR(SlaveSeqNO), ADR(OldSlaveCrc), FALSE*);*  SlaveSeqNO 1,- | Reset {Сброс) |
| PARA.RESET1 | Frame.Command - Reset AND IS\_CRC\_ CORRECT(Frame, 0,  ADR <In i t SeqNo), ADR(OldMasterCrc), FALSE) • TRUE | LastCrc 0;  OldMasterCrc :• 0;  OldSlaveCrc 0;  MasterSeqNo 1;  SlaveSeqNo :• 1;  InltSeqNo :«■ 1;  DataCommand :• FaJiSafeOata; CommFaultReason :• 0; SendFrame(Reset, ADR(CommFaultReason), LastCrc, 0,  ADR(SlaveSeqNO), ADR(OldSlaveCrc),  FALSE*);*  SlaveSeqNO 1; | Reset {Сброс) |
| PARA.FAIL8 | Frame.Command - Reset AND IS\_CRC\_ CORRECT{Frame, 0,  ADR{Ini tSeqNo), ADR(OldMasterCrc), FALSE) - FALSE | LastCrc 0;  OldMasterCrc :» 0;  OldSlaveCrc 0;  HasterSeqNo 1;  SlaveSeqNO :<• 1;  InitSeqNo 1;  DataCommand *FailSaieData;* CommFaultReason INVALID\_CRC; SendFrame(Reset, ADR(CommFaultReason), LastCrc, 0,  ADR(SlaveSeqNO), ADR(OldSlaveCrc), FALSE);  SlaveSeqNO :• 1; | Reset {Сброс) |

62

#### ГОСТ Р МЭК 61784-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| PARA.RESET2 | Frame.Command -  *Session*  AN О IS\_CRC\_CORRECT(Frame, 0,  ADR(InitSeqNO), ADR(OldMas terCrc», FALSE» • TRUE | LastCrc :- Frame.CrcO; MasterSeqNo  :» *2;*  InitSeqNO I; SlaveSeqNo 1;  DataCommand *FatlSsfeData;*  Sessionld CREATE\_SESSION\_ID(»; SendFrame(Session, ADR(SessionID», LastCrc,  0,  ADR(SlaveSeqNo», ADR(OldSlaveCrc», FALSE»;  LastCrc :«■ SendFrame.CrcO; BytesToBeSent UPDATE\_BYTES\_TO\_BE\_ SENT(2»; | Session (Се­ анс) |
| PARA\_FAIL9 | Frame.Command «  *Session*  AND  IS CRC CORRECT(Frame, 0,  ADR(InitSeqNo»,  ADR(OldMasterCrc), FALSE» - FALSE | LastCrc 0;  OldMasterCrc 0;  OldSlaveCrc :« 0;  MasterSeqNo :«■ 1;  SlaveSeqNo 1;  InitSeqNO 1;  DataCommand :« *FailSaieData;* CommFaultReason :- INVAL10\_CRC; SendFrame(Reset, ADR(CommFaultReason», LastCrc,  0,  ADR(SlaveSeqNo», ADR(OldSlaveCrc», FALSE);  SlaveSeqNo :• 1; | Reset (Сброс) |
| PARA.FAtHO | Frame.Command -  *Connection* | LastCrc :• 0,-  OldMasterCrc 0,-  OldSlaveCrc 0;  MasterSeqNo :• 1;  SlaveSeqNo :• 1; DataCommand *FailSaieData;*  CommFaultReason 1NVAL1D\_CMD; SendFrame(Reset, ADR(CommFaultReason», LastCrc, 0,  ADR(SlaveSeqNo», ADR(OldSlaveCrc),  FALSE);  SlaveSeqNo :• 1; | Reset (Сброс) |

63

#### ГОСТ Р МЭК 61764-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| PARA FAIL11 | {Frame.Command <> | LastCrc 0; | Reset {Сброс) |
| Reset | OldMasterCzc 0; |
| AND | OldSlaveCrc :« 0; |
| Frame.Command <> | MasterSeqNo :« 1,- |
| *Session* | SlaveSeqNo 1; |
| AND | DataCommand :» *FailSsCeDeta;* |
| Frame.Command <> | CommFa и11Rea son UNKNOWN CMD; |
| Connection | SendFrame{Reset, |
| AND | ADRJCommFaultReason), LastCrc, |
| Frame.Command <> | 0, |
| Parameter | ADR(SlaveSeqNo), ADR(OldSlaveCrc), |
| AND | FALSE); |
| Frame.Command <> FailSafeOata  AND  Frame.Command <> Processesta) | SlaveSeqNo 1; |

* + - 1. Событие истекшего сторожевого таймера

Невозможно в данном состоянии, так как сторожевой таймер еще не был запущен.

* + - 1. Событие сброса соединения

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состоялне |
| PARA.RESET3 |  | LastCrc 0;  OldMasterCrc :• 0;  OldSlaveCrc 0,-  MasterSeqNc 1,-  SlaveSeqNo :• 1; DataCocmand FailSafeData; CoeeiFaultReason 0; SendFrame(Reset,  ADR(CoeeiFaultReason), LastCrc, 0,  ADR(SlaveSeqNo), ADR(OldSlaveCrc),  FALSE);  SlaveSeqNo :• 1; | Reset (Сброс) |

* + - 1. Событие Команды Set Data

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состоял\*\* |
| PARA.STAY2 |  | DataCoemand :« DataCmd; | Parameter {Па- раме гры) |

* + 1. Состояние данных
       1. Событие получения кадра

64

#### ГОСТ Р МЭК 61784-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| DATA.OK1 | Frame.Command - | SafeOutputs Frame.SafeData; | Data (Данные) |
| *ProcessData* | LastCrc :» Frame.CrcO; |
| AND | SendFrame(DataCommand, |
| Frame.ConnId - | ADR(Safelnputs), LastCrc, |
| Connectionld AND | Connectionld, ADR(SlaveSeqNo), |
| IS CRC CORRECT <Frame, | ADR{01dSlaveCrc), TRUE); |
| LaatCcc, ADR{MasterSeqNo), | LastCrc SendFrame.CrcO; START |
| ADR{01dMasterCrc), TRUE)  - TRUE | HD{Hatchdog); |
| DATA.OK2 | Frame.Command • | SafeOutputs FS\_VALUE; | Data (Данные) |
| *FailSafeData* AND | LastCrc Frame.CrcO; |
| Frame.Connld - | SendFrame <DataCommand, |
| Connectionld AND | ADR(Safelnputs), LastCrc, |
| IS CRC CORRECT<Frame, | Connectionld, AOR{SlaveSeqNo), |
| LaSLCrc, ADRfMasterSeqNo), | ADR (OldSlaveCrc), TRUE); |
| ADR(OldMasterCrc), TRUE) | LastCrc SendFrame.CrcO; START |
| - TRUE | WD{Watchdog); |
| DATA.FAIL1 | {Frame.Command - | LastCrc 0; | Reset (Сброс) |
| *ProcessData* | OldMaaterCrc 0; |
| OR | OldSlaveCrc 0; |
| Frame.Command - | MasterSeqNo 1; |
| raiiSa/eData) AND | SlaveSeqNo : - 1; |
| Frame.Connld - | DataCommand *FailSafeData;* |
| Connectionld AND | SafeOutputs FS VALUE; STOP |
| IS CRC CORRECT(Frame, | HDO; |
| LastCrc, ADR(MasterSeqNo), | CommFaultReason INVALID CRC; |
| ADR<OidMasterCrc), TRUE) | SendFrame{Reset, |
| - FALSE | ADR{CommFauitReason), LastCrc, 0,  ADR{SlaveSeqNo),  ADR{OldSlaveCrc), FALSE); SlaveSeqNo :- 1; |
| DATA.FAIL2 | {Frame.Command - | LastCrc 0; | Reset (Сброс) |
| ProcessOata | OldMaaterCrc :» 0; |
| OR | OldSlaveCrc 0; |
| Frame.Command • | MasterSeqNo 1; |
| *FallSateData)* AND | SlaveSeqNo :- 1; |
| Frame.Connld *<>* | DataCommand *FallSafeData;* |
| Connectionld | SafeOutputs FS VALUE; STOP WD<);  CommFaultReason INVALID\_ CONNIO;  SendFrame{Reset, ADR{CommFaultReason), LastCrc, 0,  ADR{SlaveSeqNo), ADR{OldSlaveCrc), FALSE); SlaveSeqNo :• 1; |

65

#### ГОСТ Р МЭК 61784-3-12»2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| DATA.RESET1 | Frame.Cooaand - Resec AMD IS\_CRC\_CORRECT(Frame, 0,  ADR (InitSeqNo), ADR(OldMasterCrc), FALSE)   * TRUE | LastCrc 0;  OldMasterCrc 0;  OldSiaveCrc *z- 0;*  HasterSeqNo 1;  SlaveSeqNo 1;  InltSeqNo 1;  DataComnand :• FaiJSafeDaea; SaTeOutputs FS VALUE; STOP  «DO;  CoecnFaultReason 0; SendFrame(Reset, ADR(CoecaFaultReason), LastCrc, 0,  ADR(SlaveSeqNo), ADR(OldSiaveCrc), FALSE); SlaveSeqNo 1; | Reset (Сброс) |
| OATA\_FA!L3 | Frame.Coemand - Resec AND IS\_CRC\_CORRECT(Frame, 0,  ADR(InitSeqNo), ADR(OldMasterCre), FALSE)  - FALSE | LastCrc :• 0;  OldMasterCrc 0;  OldSiaveCrc :• 0;  MasterSeqNo 1;  SlaveSeqNo 1;  InitSeqNo 1;  DataCoamand FailSafeDaca; SafeOutputs FS VALUE; STOP  «DO;  CoeeiFaultReason INVALID\_CRC; SendFrame(Reset, ADR(CoecnFaultReason), LastCrc, 0,  ADR(SlaveSeqNo),  ADR(OldSlaveCrC), FALSE);  SlaveSeqNo 1; | Reset (Сброс) |
| DATA\_RESET2 | Frame. Coemand - Session  AND IS\_CRC\_CORRECT(Frame, 0,  ADR(InitSeqNo), ADR(OldMasterCrc), FALSE)  - TRUE | LastCrc :• Frame.CrcO; MasterSeqNo 2;  InitSeqNo :• 1;  SlaveSeqNo 1; DataCoenand FaiiSafeData; saeeoutputs f s v a l u e ,- s t o p WD();  Sessionld CREA7E\_SESSION\_ID();  SendFrame(Session, ADR(SesslonlD), LastCrc, 0,  ADR(SlaveSeqNo),  ADR(OldSiaveCrc), FALSE); LastCrc SendFrame.CrcO;  BytesToBeSent UPDATE\_BYTES\_TO\_ BE\_SENT(2); | Session (Се­ анс) |

66

#### ГОСТ Р МЭК 61784-3-12—2016

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| DATA.FAIL4 | Frame.Command -  *Session*  AND  IS\_CRC\_CORRECT (Frame, 0,  ADR(InitSeqNo), ADR(OldMaaterCrc), FALSE)  - FALSE | LastCrc 0;  OldMastecCrc :• 0;  OldSlaveCcc 0;  MasterSeqNo 1;  SlaveSeqNo 1;  InltSeqNo 1;  DataCommand *FailSafeData;* SafeOutputs FS VALUE; STOP ND<);  CommFaultReason INVALID\_CRC; SendFrame(Reset, ADR(CommFaultReaaon), LastCrc, 0,  ADR(SlaveSeqNo),  ADR(OldSlaveCcc), FALSE); SlaveSeqNo 1; | Reset (Сброс) |
| DATA.FAIL5 | Frame.Command - Connection  OR  Frame.Command -  *Parameter* | LastCrc 0;  OldMastecCrc 0;  OldSlaveCcc 0;  MasterSeqNo 1;  SlaveSeqNo :- 1; DataCommand FailSafeData; SafeOutputs FS VALUE; STOP WD<);  CommFaultReason INVALID\_CMD; SendFrame(Reset, ADR(CommFaultReason), LastCrc, 0,  ADR(SlaveSeqNo),  ADR(OldSlaveCcc), FALSE); SlaveSeqNo :- 1; | Reset (Сброс) |
| DATA.FAIL6 | (Frame.Command <> Reset  AND  Frame.Command <>  *Session*  AND  Frame.Command <>  *Connection*  AND  Frame.Command <> Parameter  AND  Frame.Command <> FailSafeOata  AND  Frame.Command <> ProcessOata) | LastCrc 0;  OldMastecCrc 0;  OldSlaveCrc 0;  MasterSeqNo 1;  SlaveSeqNo :- 1; DataCommand *FallSaFeData;* SafeOotputs FS VALUE; STOP WD<);  CommFaultReason UNKNOWN\_CMD; SendFrame(Reset, ADR(CommFaultReason), LastCrc,  0,  ADR(SlaveSeqNo), ADR(OldSlaveCcc), FALSE); SlaveSeqNo 1; | Reset (Сброс) |

67

#### ГОСТ Р МЭК 61784.3-12—2016

* + - 1. Событие истекшего сторожевого таймера

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее со\* стояние |
| DATA.WD |  | LdStCrc 0;  OldMeSterCrc 0;  OldSlaveCrc 0;  M&SterSeqNO 1;  SlAveSeqNo 1; DAtACommend Fa IISa feDAtA;  SAfeOutputS FS\_VALUE; STOP\_WD(); CoounFAultReason WD\_EXPIRED; SendFrome(Reset, ADR(CommFAultReason), LestCrc,  0,  ADR(SlaveSeqNo), ADR(OldSleveCrc), FALSE);  SlaveSeqNo 1; | Reset (Сброс) |

* + - 1. Событие сброса соединения

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее со­ стояние |
| DATA.RESET3 |  | LAStCrC 0;  OldMasterCrc :« 0;  OldSleveCrc 0; MesterSeqNo :- l; SlaveSeqNo :- 1; DetaCoamand FallSafeData;  SafeOutputs FS\_VALUE; STOP\_WD(); CommFaultReeson :» 0; SendFrame(Reset, ADR(CommFaultReason), LastCro,  0,  ADR(SlaveSeqNo), ADR(OldSlaveCrc), FALSE);  SlaveSeqNo :• l; | Reset (Сброс) |

7.S.6.4 Событие Команды Set Data

|  |  |  |  |
| --- | --- | --- | --- |
| Переход | Условие | Действие | Следующее состояние |
| DATA.STAY |  | DataCoeaumd Datacard; | Data (Данные) |

## Управление коммуникационным уровнем безопасности

* 1. Обработка параметров FSCP 12/1

Протокол FSCP 12/1 поддерживает встроенную возможность скачивания параметров ведомого устройства FSoE у ведущего устройства FSoE в состоянии параметров.

* 1. Параметры коммуникаций FSoE

Коммуникации FSoE между ведущим устройством FSoE и ведомым устройством FSoE используют коммуникационные параметры, заданные в таблице 39.

68

#### ГОСТ Р МЭК 61784-3.12—2016

Таблица 39 — FSoE Communication parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Диапа\* эон | Описание |
| ID Сеанса FSoE | UINT16 | 0..2\* | Произвольно генерируемый ID сеанса FSoe |
| ID соединения FSoE | UINT16 | 1 ... 21в | Уникальный ID соединения между ведущим устройством FSoE и ведомым устройством FSoE |
| Порядковый номер FSoE | UINT16 | Init: 0  1 ... 21в | Увеличивается в каждом цикле FSoE |
| Адрес ведомого устрой­ ства FSoE | UINT16 | 1 ... 21в | Уникальный адрес ведомого устройства FSoE для каждого устройства FSoE |
| Время сторожевого тай­ мера FSoE | UINT16 | 1... 2,в | Время сторожевого таймера для соединения FSoE в мс |

## Системные требования

* 1. Индикаторы и коммутаторы
     1. Состояния индикаторов и частота вспышек

Состояния индикаторов и частота вспышек определены в таблице 40 и на рисунке 11.

Перечисленные в этих таблицах времена должны соблюдаться с отклонением меньше чем ± 20%.

Таблица 40 — Состояния индикаторов

|  |  |
| --- | --- |
| Состояние индикатора | Определение |
| включен | Индикатор должен быть постоянно включен |
| выключен | Индикатор должен быть постоянно выключен |
| единичная вспышка | Одно короткое мерцание индикатора (50 мс). за которым следует фаза «выключен» длительностью минимум 200 мс |
| мерцание | Индикатор должен включаться и выключаться через равные интервалы с частотой в 10 Гц. т. 6. 50 мс включен и 50 мс выключен |
| мигание | Индикатор должен включаться и выключаться через равные интервалы с частотой в  2.5 Гц. г. е. 200 мс включен и 200 мс выключен |
| мерцание с 1 вспышкой | Сначала индикатор должен мерцать 500 мс. затем одна фаза «выключен» (500 мс). затем короткая вспышка (200 мс). за которой следует длинная фаза «вьключен» (1 000 мс) |
| мерцание с 2 вспышками | Сначала индикатор должен мерцать 500 мс. затем одна фаза «выключен» (500 мс). затем последовательность из двух коротких вспышек (200 мс). разделенных фазой  «выключен» (200 мс). за которой следует длинная фаза «выключен» (1 000 мс) |
| мерцание с п вспышками | Сначала индикатор должен мерцать 500 мс. затем одна фаза «выключен» (500 мс). затем последовательность из л коротких вспышек (200 мс). разделенных фазой  «выключен» (200 мс). за которой следует длинная фаза «выключен» (1 000 мс) |

69

#### ГОСТ Р МЭК 61764-3-12—2016

ил

*€лт\** ~~н~~нктим

:\_50не

П

арчи a»m\*fli4«ifi«»n

**М Л** я я л

М~~ЦЩ~~я~~Н~~

кя ЪМРММО

КПМ\*»\*

МА

с?

И\*(Ч«М«СИ

KfWtflU

.аомс

# 11Ш1Ш1™ШШЖ1ШШШШПЛ1ШШ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tow 200и: |  |  |  |  |  |
| 600 ИС | SDOwc | 200 МС |  | ' КЮ« | *X1* |
| **ППППП** |  |  |  |  | **1ЛЛЛЛ** |
| 600 Л | SO0HC |  |  |  | 1 000 ис |
| **ППППП** |  |  |  |  |  |
| 000 ис | 500и£ | 200\* |  |  | 1000м: |
| **лшшл** |  |  |  |  | **"U1** |

тля

* + 1. Индикаторы

Рисунок 11 — Частота вспышек индикаторов

* + - 1. Требующиеся индикаторы

Устройство, поддерживающее протокол FSCP12/1 должно иметь индикатор статуса для поддерж- ки при визуальном осмотре и выявлении неисправностей соединения FSoE. Если устройство поддер­ живает какие-либо из описанных здесь индикаторов, то они должны придерживаться спецификаций.

Могут быть задействованы дополнительные индикаторы.

* + - 1. Индикатор FSoE STATUS

Индикатор STATUS (статус) FSoE должен отображать статус FSoE соединения. Он должен гореть зеленым цветом.

Для того чтобы не нарушать пространственные ограничения, можно опустить маркировку инди­ катора STATUS для FSoE. Если же индикатор промаркирован, то маркировка должна быть следующей (не чувствительна к регистру):

-FS:

* FSoE;
  + Статус FSoE.

Состояния индикатора STATUS для FSoE описаны в таблице 41.

Таблица 41 — Состояния индикатора STATUS для FSoE

|  |  |  |
| --- | --- | --- |
| Состояние индикатора | Определение | Состояние FSoE |
| выключен | Осуществляется инициализация | Перед сбросом (Pre-Reset) |
| мигает | Готов к параметризации | Сброс, сеанс, соединение, параме­ тры |
| включен | Нормальное функционирование | Данные Процесса |
| единичная вспышка | Отказоустойчивые данные | Отказоустойчивые данные |
| мерцание | Неустановленная ошибка в соединении FSoE | Все |
| мерцание с 1 вспышкой | Ошибка в F-парамегре | Параметры |
| мерцание с 2 вспышками | Ошибка в параметре приложения | Параметры |

70

#### ГОСТ Р МЭК 61784-3.12—2016

*Окончание таблицы 41*

|  |  |  |
| --- | --- | --- |
| Состояние индикатора | Определение | Состояние FSoE |
| мерцание с 3 вспышками | Неверный адрес безопасности | Соединение |
| мерцание с 4 вспышками | Неверная команда | Все |
| мерцание с 5 вспышками | Ошибка сторожевого таймера | Все |
| мерцание с 6 вспышками | CRC-Ошибка | Все |

Индикация ошибки (мерцание) должна длиться до тех пор. пока соединение FSoE не сменит состояние сброс на состояние сеанс.

Должка быть показана хотя бы одна индикация в виде полной последовательности мерцаний с л

числом вспышек.

* 1. Руководство по установке

8 настоящем стандарте описан протокол и услуги для коммуникационной системы безопасно­ сти. основанные на МЭК 61158. Тип 12. Тем не менее, применение устройств безопасности вместе с протоколом безопасности, описанным в настоящем стандарте, требует их надлежащую установку. Все устройства, соединенные с коммуникационной системой безопасности, определенной в настоящем стандарте, должны выполнять требования SELV/PELV. описанные в соответствующих стандартах МЭК. таких как МЭК 60204-1. Другие важные руководящие указания по установке приведены в МЭК 61918.

* 1. Время реакции функции безопасности
     1. Общие положения

Для определения времени реакции функции безопасности, функция безопасности декомпозирует­ ся на несколько компонентов, как это показано на рисунке 12.

т S т I т Сот

Т L Т Сот

ТО ТА

Датчик

Васю

Коммуни­ кации

Лосиха

Коммут­

ации ,

|  |  |  |
| --- | --- | --- |
|  |  | Исполни- |
| Вывод |  | тельное устройство |

время реакции функции безопасности

Рисунок 12 — Компоненты функции безопасности

Не все компоненты должны быть представлены в системе. Датчик (например, световая завеса или кнопка аварийной остановки) преобразует физический сигнал в электрический. Этот сигнал может быть подключен к устройству ввода (например, вводу безопасности), который преобразует сигнал в логическую информацию. Эта логическая информация передается по коммуникационной сети безопас­ ности в логический узел безопасности. Логический узел безопасности (например. PLC безопасности) объединяет эту и/или другую входную информацию в логическую выходную информацию. Выходная информация передается устройству вывода (например, выводу безопасности) и преобрауется в элек­ трический сигнал. Данный сигнал соединяется с исполнительным устройством, которое выполняет фи­ зическую реакцию, например, отключение питания привода.

Основные допущения в случае коммуникационных ошибок следующие:

* + все компоненты работают асинхронно:
* обработка входных(ой) сигналов/икформации не зависит от обработки выходных(ой) сигналов/ информации. Это означает, что каждая сторона может обладать своим поведением во времени;
  + для того чтобы вычислить время реакции функции безопасности, должна допускаться только одна ошибка или один отказ во всей системе. Предполагается, что эта ошибка или отказ произошел в той части пути сигнала, которая формирует наибольшую разницу во времени между его временем

71

#### ГОСТ Р МЭК 61764-3-12—2016

задержки е худшем случае и временем сторожевого таймера. Это означает, что одновременные отказы не рассматриваются.

В таблице 42 определены времена для компонентов.

Таблица 42 — Определение времен

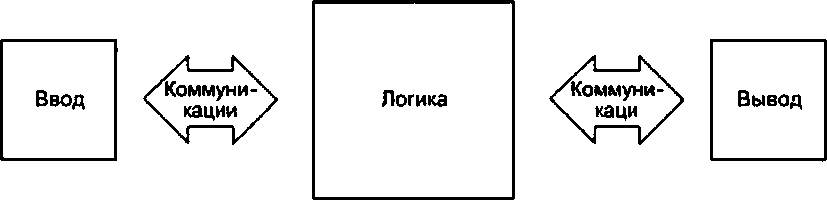
|  |  |  |
| --- | --- | --- |
| Врем\* | Название | Описание |
| T.SFR | Время реакции функции безопас­ ности | Время реакции функции безопасности от физического ввода до реакции на исполнительном устройстве |
| TJnCon | Время соединения ввода | Время передачи сигнала физического ввода в логический узел безопасности |
| T\_OutCon | Время соединения вывода | Время передачи расчитайного сигнала вывода от логического узла безопасности исполнигвгъному устройству |
| T\_S | Время датчика | Время преобразования на датчике безопасности |
| TJ | Время ввода | Время задержки вводного устройства безопасности |
| T\_Com | Время коммунжаций | Время коммуникационного цикла для коммуникационной сети |
| T\_L | Время логического узла | Время задержки логического узла (цикл) |
| T.O | Время вывода | Время задержки выводного устройства безопасности |
| T.A | Время исполнительного устрой­ ства | Время преобразования на исполнительном устройстве безопасности |
| T.WDJn | Время сторожевого таймера вво­ да | Время сторожевого таймера FSoE для соединения ввода |
| T.WD.Out | Время сторожевого таймера вы­ вода | Время сторожевого таймера FSoE для соединения вывода |
| ДТ | запас времени сторожевого тай­ мера | Дополнительный запас времени для минимального сторожевого таймера |

Так как принято считать, что все компоненты работают асинхронно, наихудшее время для каж- дого компонента является удвоенным временем задержки каждого компонента. Это происходит, если

«двигающаяся» информация становится доступной сразу после запуска процесса. Времена наихудших случаев помечаются суффиксом a\_wc.

* + 1. Установление времени сторожевого таймера FSoE

На рисунке 13 показана базовая схема для вычисления времени сторожевого таймера FSoE для соединения ввода и вывода.

TJ T\_Com TL Т\_Сот Т\_0

Соединение ввода

4-----------------------------------------»

TWDJn

Соединение вывода

4-----------------------------------------►

Т WD Out

Рисунок 13 — вычисление времени сторожевого таймера FSoE для соединений вводе и вывода

72

#### ГОСТ Р МЭК 61784-3.12—2016

Для того чтобы определить время сторожевого таймера соединения ввода T\_WD\_ln. можно ис­ пользовать уравнение (1):

T\_WD\_ln = T\_l\_wc ♦ T\_Com\_wc + T L\_wc ♦ T\_Com\_wc + ДТ (1)

= 2 x TJ ♦ 4 x T\_Com ♦ 2 \* T\_L +ДТ.

Аналогично уравнение <2) позволяет вычислить время сторожевого таймера соединения вывода T\_WO\_Out:

T\_WD\_Out = T\_Com\_wc \* T\_L\_wc + T\_Com\_wc + T\_0\_wc + ДТ (2)

= 4 x T Com + 2\*T L\*2\*T О + ДТ.

* + 1. Вычисление наихудшего времени реакции функции безопасности

На рисунке 14 показана базовая схема для вычисления наихудшего времени реакции функции безопасности.

Т\_$ TJ T\_Com T\_L Т.сот Т\_0 Т\_А

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | /1—J\ |  | И— |  |  | Исполни- |
|  | /комму\*\*ь\ Ч кации */* |  | /Хоммунм\  N. кации */* |
| Ввод | Лоте | Вывод |  | тельное |
|  |  |  |  | устройство |

Датчик

Т loCoon T\_OutConn

----------------------------------------------------------------- ► --------------------------------------------------------------------------------►

Рисунок 14 —Вычисление наихудшего времени реакции функции безопасности

Время передачи информации сигнала датчика логическому узлу безопасности, то есть время TJnConn, может быть вычислено как:

TJnConn = T\_S\_wc + T\_l\_wc ♦ T\_Com\_wc + T\_L\_wc (3)

s 2 x T\_S ♦ 2 x T\_l + 2 x T\_Com + 2\* T\_L.

Наихудшее время получения информации безопасного состояния от сигнала датчика логическим узлом безопасности или время T\_lnConn\_wc — это время в том случае, когда входные коммуникации прерываются и истекает время сторожевого таймера соединения ввода. В этом случае в логике безопасности используются отказоустойчивые значения входных сигналов. Это время может быть вычислено как:

T\_lnConn\_wc *-* T\_S\_wc + T\_WD Jn (4)

г 2 x t \_S ♦ T.WDJn.

Время на передачу вычисленного выходного сигнала от логического узла безопасности исполнительному устройству или время T\_OutConn может быть вычислено как:

TJnConn = T\_L\_wc + T\_Com\_wc + T\_0\_wc + T\_A\_wc (5)

= 2\* T\_L + 2 x T\_Com + 2\* T\_0 + 2\* T\_A.

Наихудшее время на передачу вычисленного выходного сигнала от логического узла безопасности исполнительному устройству или время T\_OutConn\_wc, это время в случае, когда выходные коммуникации прерываются и истекает время сторожевого таймера соединения вывода. В таком случае активируются отказоустойчивые значения в устройстве вывода. Это время может быть вычислено как:

T\_OutConn\_wc = TJ wc + T\_WD\_Out + T\_A\_wc (6)

= 2 x TJ. ♦ T\_WD\_ Out + 2 x T\_A.

Для того чтобы вычислить время реакции функции безопасности, должна допускаться только одна ошибка или один отказ на том пути сигнала, который создает максимальную разницу во времени между его наихудшим временем задержки и временем сторожевого таймера. Это означает, что одновременные отказы не рассматриваются.

73

#### ГОСТ Р МЭК 617W-3.12—2016

Для определения наихудшего времени реакции функции безопасности или времени T\_SFR\_wc используется уравнение (7):

T\_SFR\_wc = max{T\_lnConn\_wc ♦ T\_OutConn; T\_OutConn\_wc + TJnConn). (7)

При необходимости изготовители системы должны предоставлять индивидуальный адаптирован\* ный метод вычисления.

* 1. Длительность запросов на обслуживание

Длительность запросов на обслуживание приложения, связанного с безопасностью, коммуникационным уровнем безопасности может быть равна времени процесса безопасности или таймауту FSCP 12/1 (сторожевой таймер FSoE) или превышать их.

* 1. Ограничения на вычисление характеристик системы
     1. Общие положения

FSCP 12/1 не накладывает никаких ограничений на:

* минимальное время цикла коммуникаций;
* количество данных безопасности, приходящихся на одно устройство FSoE:
  + основную коммуникационную систему.

Все устройства должны быть электробезопасными. используя SELV/PELV.

Все устройства спроектированы для нормальных промышленных сред в соответствии с МЭК 61000\*6\*2 или МЭК 61131\*2 и обеспечивают повышенную защищенность в соответствии с МЭК 61326\*3\*1 или МЭК 61326-3\*2.

Коммуникационный путь выбирается произвольно. Он может быть системой полевых шин. Eth\* егпеГом или реализован другими подобными средствами, оптоволоконными кабелями, медными кабелями или даже путем беспроводной передачи. Нет никаких ограничений или требований для использования шик или других устройств в коммуникации.

Дополнительное внесение трех нулевых октетов в вычисление CRC. вместе с наследованием CRC гарантирует независимость основной коммуникации, даже если используется один CRC полином.

Коммуникационный интерфейс в устройствах безопасности может быть одноканальным интерфейсом. Такой интерфейс может быть избыточным для обеспечения готовности.

Соединение между устройствами FSoE является соединением ведущего устройства с ведомым.

Ведущее устройство FSoE обладает одним или несколькими соединениями FSoE с одним или несколькими ведомыми устройствами FSoE. Ведомое устройство FSoE реагирует только на ведущее устройство FSoE. Система может различать до 65 535 соединений.

* + 1. Ограничения, полученные в результате вероятностного анализа

Каждая обнаруженная ошибка в коммуникациях безопасности должна инициировать переход в состояние сброса, т. е. в безопасное состояние. Такой переход не должен происходить более одного раза за 5 часов, т. е. частота возникновения остаточной ошибки должна быть лучше, чем 1СГ2/ч.

Доказано, что CRC полином с включением трех нулевых октетов (так называемых виртуальных битов) гарантирует независимость для стандартной проверки основной коммуникационной системы.

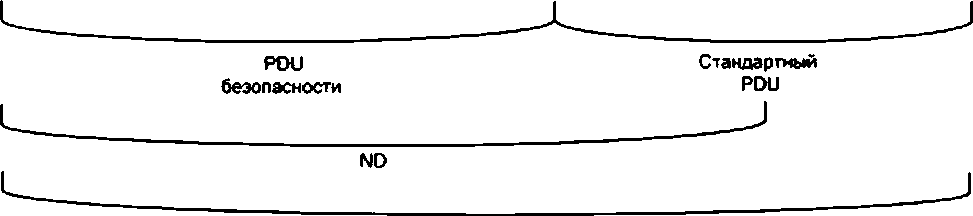
РОиТипа12состоитиэчасти.связаннойсбезопасностью.истандартной части. Часть безопасности встраивается в стандартную часть. На рисунке 15 показан POU. состоящий из SafetyData NDeefely4 виртуальных битов с длиной би(в|у = 24 бита. FCSsefely безопасности, стандартного NDslandard полезных данных и стандартного FCSslanMrd.

Safety — безопасность.

74

#### ГОСТ Р МЭК 61784-3.12—2016

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NDsafey | 0x000000 | FCSgafeiy | NDgfcndard | FCSsiandanj |



PDU

Рисунок 15 — PDU безопасности, встроенный в стандартный POU

Были получены следующие требования:

* + xdssfe(y4i и генератор полиномов основные друг для друга:

-число виртуальных битов dsafety меньше или равно числу битов для стандартной части

^safety ^ ^elandafd)\*

* + частота возникновения остаточных ошибок ниже 10'9/ч.

При использовании примитивного полинома безопасности 139B7h зти требования выполняются при следующих условиях:

* + число битов данных безопасности — 8 или 16 (ND8afe1y = 8 или NDsafely = 16):
* число виртуальных битое равно 24 {d8afely = 24);
* минимальное число стандартных битов — 16 (NOstandard а16);
* максимальное число стандартных битов — 12 144 (NOstan<jafd *S* 12 144).

Примечание —Было доказано, что до 12 144 битое (1 518 октетов) используется для Ethernet в качестве максимальной длины DPDU:

-стандартные биты могут снова содержать блоки данных безопасности, состоящие из данных безопасности и FCSsa(ety.

На рисунке 16 показана частота возникновения остаточных ошибок для данных безопасности из

1. 16 и 24 битов. При максимальной вероятности возникновения битовой ошибки, равной 10’2/ч. ве­ роятность возникновения остаточной ошибки ниже 10-\*/ч для 8- и 16-битовых данных безопасности. 24-битовые данные безопасности не используются в рамках данного протокола.

75

#### ГОСТ Р МЭК 61784-3-12—2016

Пом<ои1Э9вП| Ммсиийт»4я<1(к?о1а«ст8г!миса1 оьибм мн да 6<i6i24 битоа мчсормашм Смом&ест. W ал»1уда«и« б\*м бехпасмсети

16 t2i«46\*rte<iaM»w\*>«eiH-«0CM»uw er\*e«e0cr»wea»tw»eiei>arbMi4»C«rt6e

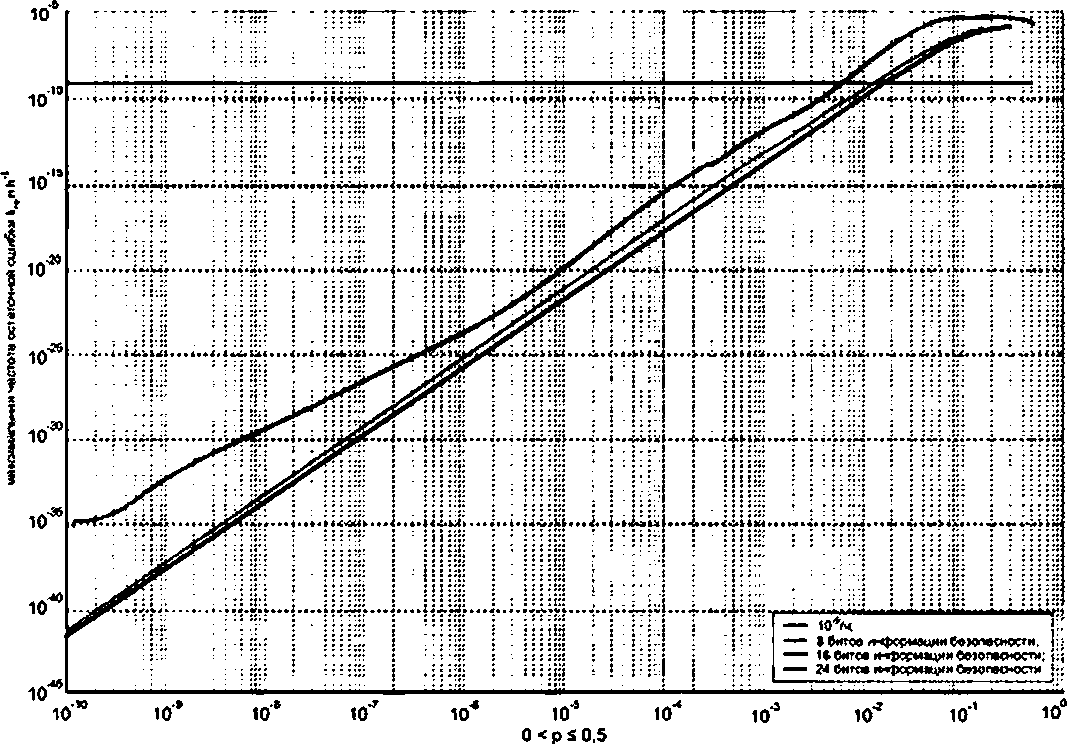


Рисунок 16 — Частота возникновения остаточной ошибки доя 6/16/24-битовых данных безопасности и стандарт­ ных данных до 12 144 битов

* 1. Техническое обслуживание

Этот протокол не содержит специальных требований к техническому обслуживанию.

* 1. Руководство по безопасности

Специалисты по внедрению настоящего стандарта должны предоставить руководство по безопасности, содержащее, как минимум, следующую информацию:

* + руководство по безопасности должно оповещать пользователей об ограничениях на вычисления системных характеристик, см. 9.5:

- руководство по безопасности должно оповещать пользователей об ответственностях, связанных

с надлежащей параметризацией устройства.

Кроме требований данного раздела, руководство по безопасности должно удовлетворять всем  требованиям МЭК 61508.

## Оценка

Настоятельно рекомендуется, чтобы специалисты по внедрению FSCP 12/1 получили результаты верификации всех аспектов функциональной безопасности изделия от независимого компетентного органа оценки, включая протокол и все приложения. Настоятельно рекомендуется, чтобы специалисты по внедрению FSCP12/1 получили доказательство того, что независимым компетентным органом оцен­ ки был выполнен подходящий тест на соответствие.

76

#### ГОСТ Р МЭК 61784-3.12—2016

ПриложениеА (справочное)

Дополнительная информация для профилей коммуникаций функциональной безопасности

CPF12

А.1 Вычисление хэш функции

Нижеприведенный код для PDU безопасности представляет из себя пример того, как вычисляются CRC ходы PDU безопасности. В таблицах учитываются три нулевых младших разряда.

Параметр: psPackeL - FSCP12/1 Safety PDU

startCrc - Начальное значение вычисления CRC seqNo - SeqMo

oldCRC - CRC\_0 последнего принятого/отлраьленного PDU ведомого

- устройства безопасности

bRcvDlr - bRcvDit - True: вычисления CRC принятого кадра

bRcvDir - False: вычисления CRC для отправленного кадра size - размер PDU безопасности

Возвращает: bSuCcess - TRUE: CRC верен

............................................................................................................................................... /

UIHT8 CalcCrc(SAFETY\_PDU «psPacket, UIHT16 startCrc, UIN716 • aeqNo, UINT16 OldCrc, UINT8 bRcvDir, UIH78\_size)

(

UIHT16 и1,и2; // временные значения UIHT16 crc;

UXHT16 сгс\_сое\*воп; // общая часть вычисления CRC,

// включает CRC\_0, Conn-ZD, Sequence-Но., Cmd UINT8 ’pCrc - apsPacket->au8Data(2]; // указатель на младший байт CRC

U1HT8 \*pSafeData // указатель на младший байт SafeData

if ( size > 6 )

pCrc\*-;

do (

// \*.e. 2 или кратное двум данным безопасности

//-> младший байт СгсО при байтовом

// смешении 3 вместо 2

ere « 0; // сброс сгс

// Последовательность для вычисления:

// старый CRC-Lo, старый CRC-H1, Connld-Lo, Connld-Hi, SeqNo-Lo, SeqMo-Hi, Comnand,

// (Индекс,) Данные

// CRC-LO

Hi - aCRCTabl(((U1HT8 •) Acre)(HI\_BY?EJ); // Обратитесь к CRC-таблице

и2 - aCRCTab2(((UINT8 •) astartCrc)(0)J; // Обратитесь к CRC-таблице Hi - Hi XOR \*2;

( (UINT8 \*1 4СГС)(H1\_BYTEI - ( (UXNT8 \*) 4Hl){HI\_BYTEJ XOR ((U1HT8 •)

acre)lLO\_BYTE); ( (UZKT8 ») acre) (LO BYTE] - ( (UZHT8 •) anl) (LO BYTE);

// CRC-H1

Hi - aCRCTabl(((UIHT8 •) acre)(HX\_BYTEJ);

h 2 - aCRCTab2(((UINT8 \*) astartCrc)(1J];

Hi - Hi XOR h 2;

( (UINT8 \*) acre)fHI\_BYTEJ - *(*(UINT8 \*1 anl) (HI\_BY?EJ XOR ((UIHT8 •)

acre)|LG\_BY?E); ( (UINT8 \*1 acre) (LO\_BYTE) - ((UINT8 •) anl) [LO\_BYTE*];*

*77*

#### ГОСТ Р МЭК 61784-3-12»2016

*U* Connld-Lo

wl - aCRCTabl(< {UINT8 ♦) (crc)(HI\_8YTE)|; w2 - aCRCTab2[psPacket->au8Data[aize-2||; wl - wl XOR w2;

<(UINT8 •) SCCC)(HI\_BYTEJ - <<UINT8 •) (wl)|HI\_8YTE| XOR <<UINT8 \*)

(СГС) |LO\_BYTEJ

<(UINT8 •) SCCC)(LO\_BYTEJ - <<U1NT8 \*) (wl)IL0\_3YTE|;

*U* Connld-Hi

wl - aCRCTabl(<{UINT8 ♦) (ere)|HI\_8YTE)|; w2 - aCRCTab2[psPacket->auSData[ai2e-l]|; wl - wl XOR w2;

<(UINT8 •) (cec)(HI\_BYTE] - <<U1NT8 ♦) (wl)IHI\_3YTE| XOR {{UINT8 ♦)

(CrC)(L0\_8YTE);

<(UINT8 \*) (CCC)IL0\_8YTE| - <<U1NT8 ♦> (wl)(L0\_3YTE|;

// SeqNo-Lo

wl - aCRCTdbl[< {UINT8 •> (CCC)IH1\_8YTE)|; w2 - aCRCTab2[< <UINT8 \*) aeqNO)[L0\_3YTE]);

wl - wl XOR w2;

<{UINT8 \*) (crc)(HI\_BYTEJ - <<UINT8 ♦) (wl)IHI\_3YTE| XOR <{UINT8 ♦)

(СГС)|L0\_8YTE); (<UINT8 \*) SCCC)(LO\_BYTEJ - <<U1NT8 ♦> (wl)(L0\_3YTE|;

*П* SeqNo-Hi

wl - aCRCTabl(<{UINT8 ♦) (crc)(HI\_8YTE)|;

w2 - aCRCTab2|{<UINT8 \*) aeqNO)[H1\_3YTE|);

wl - wl XOR w2;

{<UINT8 •) SCCC)(HI\_BYTEJ - <<UINT8 ») (wl)IHI\_3YTE) XOR <<UINT8 ♦)

(Crc)|L0\_8YTE J;

<(UINT8 •) SCCC)(L0\_BYTE1 - <<UINT8 \*7 (wl)IL0\_8YTE);

// Команда

wl - aCRCTabl (< (UINT8 \*> (crc)1HI\_8YTE));

w2 - aCRCTab2(pa Packet->au8Data(OFFS\_COMMAND]]; wl - wl XOR w2;

<(UINT8 ♦) (crc)[H1\_BYTE] - <<UINT8 \*> (wl)(HI\_3YTE| XOR <{UINT8 ♦)

(ere)lLO\_BYTE);

{<UINT8 ♦) (crc)[L0\_8YTE| - <<UINT8 \*) (wl)(L0\_3YTE|;

*II* часть CRC общая для всех других ссс-вычислений сохранена cre\_common - ссе;

### // Да**н**ые|0]

wl - aCRCTablп<UINT8 \*) (СГС)|HI\_BYTEJ);

w2 - aCRCTab2[psPacket->au8Data[0PPS\_0ATA]); wl - wl XOR w2;

{{UINT8 \*) (СГС)|HI\_BYTE| - <<UINT8 •) (wl)(HI\_8YTE| XOR <<UINT8 •)

ecce>IL0\_BYTE];

{{UINT8 •) (CEC)(LO\_BYTE| - <<UINT8 ») (wl)fLO\_BYTEl;

*II* если 2 байта данных безопасности — вычислить следующий байт ссс if { size > 8 )

I

### // Да**н**ые(1)

wl - aCRCTabl(<<UINT8 •) (ere)|HI\_BYTEJ);

w2 - aCRCTab2(psPacket->au8Data[OFFS\_DATA+l|); wl - wl XOR w2;

<<U1NT8 •) (СГС)[HI\_BYTE| - ((U2NT8 •) (wl)(HI\_BYTE| XOR <<0INT8 •)

(CCC)|LO\_BYTE*)}*

{{UINT8 \*) (crc)[LO\_BYTE| - <<UINT8 •) (wl)[LO\_BYTE|; I

78

#### ГОСТ Р МЭК 61784-3-12—2016

// UPDATE\_SEQ\_>iO seqNo(0]\*+;

if (seqNo(O) — 0] seqNo(0]\*+;

) while ( crc — oldccc AA (bRevDir A t(EK\_CRCJ !- 0 );

// до rex nop пока результирующий crc такой же, как и oldCrc

if (bRevDir) // для направления приема

< if ( ((UINT8 \*) АСГС)IHI\_BYTE| — pCrc[OFFS\_CRC\_HI-OFFS\_CRC\_LO)

*kU* ((UXNT8 \*) АСГС))LO\_BYTEl —pCielOJ I

{ // для направления приема

// CRC верен bSuccess - TRUE;

)

)

else // для направления передачи

<

// введите контрольную сушу Checksum

### pCrc(OFFS\_CRC\_HI-OFFS\_CRC\_LOJ - ((UIM78 \*) АСГС){HI\_BYTE); pCrC[0} - ((U1HT8 \*) ACIC)|LO\_BYTE);

)

// если передано дольше, чем 2 байта данных безопасности,

// CRC\_1 и остальное должно быть вычислено if ( size > 10 )

*i*

UINT16 i - 1;

pSafeData - pcrc\*2

pCrc ♦ - 4;

size — 7,-

while ( size >- 4 )

// Start-CRC

etc • crc СОШПОП;

// установить pSafeData в младший байт SafeData

// следующей части - SafeData(2)

// установить pCrc в младший байт CRC\_i

// вычесть первую часть кадра

// пока другие части следуют

// данная часть, уже вычислена в«дэе

// 1 (Bit 0-7) // вычислить индекс

wi - aCRCTabi(((UIMT8 \*) а с г с ) [h i\_b y t e )1; w2 - aCRCTab2( ((UIN78 \*) AilllO\_BYTE));

wl - wl XOR м2;

((UIH78 \*) ACIC){HX\_BYTE) - ((UIH78 \*) Awl)IHI\_BY?EJ XOR ((UZHT8 \*)

АСГС)lLO\_BYTE);

### ((UZN78 \*) АСГС) ILO BYTE} - ((UINT8 \*) Awl)[LO BYTE);

// 1 (Bit 8-15)

wl - aCRCTabiU(UIMT8 •) Acre)IHI\_BYTE)); w2 - aCRCTab2(((UINT8 \*) Ai){HI\_BYTE)J;

wl - wl XOR w2;

### ((UZN78 •) Acre){HI\_BYTE} - ((UIH78 \*) Awl)IHI\_BYTE] XOR ((UINT8 \*)

Acre)|LO\_BYTE); ((UZN78 \*) АСГС) [LO BYTE) - ((UINT8 \*] Awl)|LO BYTE);

// Данные 2\*1

wl - aCRCTabiH(UINT8 \*) Acre)|HI\_BYTE)J; w2 - aCRCTab2(pSafeData|0));

wl - wl XOR w2;

((UZN78 \*J ДСГС)(HI\_BYTEJ - (tUIM78 \*) Awl>tHl\_BYTE) XOR ((UZHT8 \*)

АСГС)lLO\_BYTE|; ((UZN78 \*) АСГС)IL0\_BYTE) - ((UIWT8 ♦) Awl)ILO\_BY?EJ;

79

#### ГОСТ Р МЭК 61784-3-12»2016

// Данные 2\*1+1

wl - aCRCTabl[{{UINT8 \*) (ссс)]HI\_BYTE]];

w2 - aCRCTab2(pSafeData]1]); wl - wl XOR w2;

### <<UINT8 ♦) SCCC)[HI\_BYTE] - <<UINT8 ♦) (wl)(HI\_BYTE1 XOR <<UINT8 \*)

SCCC)|LO\_BYTE);

(<UINT8 \*) tore)(LO BYTE] - <<UINT8 \*) Swl)|LO BYTE];

If < {{UINT8 \*) SCCC)(HI\_BYTE) -- pCee fl]

SS ({UINTS \*) SCCC)]LO\_BYTE] — pCCC ]0] )

I

I

else I

*ll* CRC верен

bSuccess - FALSE;

if { bRcvOir -- 0) *It* для направления передачи

I

### *it* **в**едитеконтрольнуюсу**м**уChecksum рСсеfl] - ({UINT8 \*) SCCC)]HI\_8YTE]; pCcC [0] - <(U1NT8 \*) SCCC)]L0\_BYTE1;

1

I

ычесть данную часть кадра становить в следующий младший становить в следующий младший ндекс приращения

|  |  |
| --- | --- |
| size 4; | *//* в |
| pSafeData \*■- 4; | *и* у |
| рСссО \*— 4; | *П* у |
|  | *а* и |

I I

байт SafeData байт CRC 1

cetucn bSuccess;

I

Используются следующие две таблицы: aCccTabl: ARRAY (О..255] OF WORD

1600000,16039В7

1608761,160BED6

1603775,160ОЕС2

,160736Е,1604АО9,

,1601203,1602В64,

,160441В,1607DAC,

160E6DC,160DF6B,

16061BD,16058ОА,

16025А6,1601С11,

16095В2,

160D1A9,

16056С8,

160АСО5,160F4OF,

160Е81Е,160А2С7,

1606F7F,160С37А,

160CDB8,

1609В7О,

160FACD,

1608014,16089АЗ

, 160 9АЕ5,160А352, 160Е98В,16000ЗС,

1607С39,

160458Е,160OF57,

16036ЕО,

1606ЕЕА,160575D

1603822,1600195

160599F,1606028

,1601084,1602433,

,160АО90,1609427,

,1602AF1,1601346,

1608836,1608181,

160OSFE,160Е749,

160OC7D,16035СА,

160FB58,

160BF43,

1607F13,

160C2EF,1604В4С,

16086F4,160CC2D,

16046А4,160ЕАА1,

16072FB,

160F59A,

1600316,

160 99CF,160АО78 ,160F872,160С1С5,

1608В1С,160В2АВ,

1601ЕАЕ, 1602719,1606DCO,

1605477,

1600004,160Е463

1605АВ5,1606302

1607044,16049F3

,160АЕВА,1609700,

,160CFO7,160F6BO,

,160О32А,1603A9D,

1603ВО8,160O2BF,

160ВС69,16085ОЕ,

1606297,1605В2О,

1604866,

1609698,

16011F9,

1607101,16029DB,

160AF2F,160E5F6,

160284Е,160844В,

1601О6С,

160DC41,

160BDFC,

160F725,160СЕ92

,1604731,1607Е86,

160345F,160ООЕ8,

160А1ЕО, 160 985А,160D283, 160ЕВ34,

160ВЗЗЕ,1608А89

,160СО5О,160F9E7,

16055Е2,1606CS5,

160268С,

1601F3B,16018FA,

160214D,

1606894,1605223

160ОА29,160339Е

1605СЕ1,1606556

,160FE26,160С791,

,1607947,1604OFO,

,1603D5C,160О4ЕВ,

1608048,160B4FF,

160С953,160FOE4,

1604Е32,1607785,

160ECF5, 1600542,1609F9B,

160ВАЗО, 160838А,1602F8F,

160ОВ8О, 160Е237,160А8ЕЕ,

160А62С,

1601638,

1609159,

160821F,160BSA8

160О57Е,1603СС9

,160F171,160С8С6,

,1609ОСС,160А978,

16064СЗ,1605074,

160ЕЗА2,160DA15,

16017AD,

16053В6,

1602Е1А,1607610,

1606А01,1602008,

1604FA7,

160196F,

160856А,1608CDD

,160С6О4,160FFB3,

160А7В9,1609ЕОЕ,

160O4D7, 160ED6O,1604165,

1607802,

160320В,160ОВВС

160ЕО88,160D93F

160В64О,1608FF7

,1601487,1602D3O,

,16093Е6,160АА51,

,16023F2,1601А45,

16067Е9,1605Е5Е,

1600654,1603FE3,

160509С,160692В,

160F258,

160753А,

1603121,

160СВЕС,1608135,

1604C8D,160С52Е,

1600896,160424F,

160В882,

160FC99,

1607ВР8,

160D7FD,160ЕЕ4А

,160А493,1609024,

1608Е62,160B7D5,

160FDOC, 160С4ВВ,160688Е,

1605109,

1601BDO,1602267

,1607A6D, 1604 ЗОА, 1600903,1603ОВ4,

1609СВ1,

160А5О6,160EFDF,

1600668,

1605FCB,160667С

160D8AA,160Е110

80

,1602CAS,1601512,

,1604D18,16074AF,

1608917,1608ОАО,

1603Е76,160О7С1;

160СА79,

160F3CE,160АВС4,

1609273,

#### ГОСТ Р МЭК 61784.3-12—2016

аСгсТаЬ2: ARRAY[0..255J OF WORD 16\*0000,16\*7648,16»EC90,16\*9AD8,16»E097,

16\*96DF,16\*0007,16\*7A4F,16\*F899,16\*8ED1,

16\*1409, 16\*6241, 16\*180E,16\*6E46, 16\*F49E, 16\*8206, 16\*0885, 16»BECD, 16\*2415, 164S25D,

16\*2812,16\*5E5A,16\*C482,16\*B2CA,16\*301C, 16\*4654,16»DC8C,16\*AAC4,16\*D08B,16«A603,

16\*3C1B,16\*4A53,16\*A8BD,16\*DEF5, 16\*442D, 16\*3265, 16»482A,16»3E62, !6\*A4BA, 16\*D2F2,

16\*5024,16\*266C,16\*BCB4,16\*CAFC,16\*B0B3,

16\*8CA8,16\*FAE0,16\*80AF,16\*F6E7,16\*6C3F,

16\*C6FB,16\*5023,16»2A6B,16\*6038,16\*1670,

16\*1A77,16\*98A1,16»EEE9, 16\*7431, 16\*0279,

16\*7836,16\*0B7E, 16\*94A6,16\*E2EE,16\*68CD, 16«1E85,16\*845D,16\*F215,16»885A, 16»FE12,

16\*64CA,16\*1282,16\*9054,16\*E61C,16\*7CC4,

16\*A048,16\*D600,16\*4CD8,16\*3A90,16\*40DF,

1640A80,16\*7003, 16\*068B, 16\*9053, 16\*EA1B,

16\*3697,16»A04F,16\*DA07,16\*5801,16\*2E99,

16\*B441, 16\*0209, 16\*B846, 16\*OEOE, 16\*5406, 16\*229B,16\*0070,16»B638,16\*2OE0,16\*5AA8,

16\*20E7,16\*56AF,16\*0077,16\*BA3F,16C38B9, 16\*4EA1,16\*0479,16\*A231,16»087E,16\*AE36,

16\*34EE,16\*42A6,16\*08FS, 16\*7EBO, 16\*E46S, 16\*9220, 16«E862, 16«9E2A, 16»04F2, 16\*72BA,

16\*F06O,16\*8624,16\*1CFC,16\*6AB4,16»10FB,

16»3D0A,16\*4B42,16\*3100,16\*4745,16\*0090,

16\*0994,Z6»BFDC,16\*2504,16\*5340,164191F,

16\*1518,16\*6350,16\*E186,16»970E,16»0D16,

16»66B3,16\*FC6B,16\*8A23,16\*D19A,16\*A7D2,

16\*ABD5, 16\*2903, 16\*5F4B, 16»CS93, 16»B30B,

16»6FS7,16\*F58F,16\*8307,16\*F988,16»8FC0,

16«7B5E,16\*0111,16\*7759,16«E081,16\*9BC9,

16\*7927,16\*0F6F,16\*95B7,16\*E3FF, 16\*99B0, 16\*EFF8,16\*7520,16\*0368, 16\*81BE, 16\*F7F6,

16»602E, 16»1B66, 16\*6129,16\*1761, 16\*80B9, 16«FBF1,16»B1A2,16\*C7EA,16\*5032,16»2B7A,

16\*5135,16\*2770,16\*BDA5,16\*CBED,16\*493B,

16\*4530,16\*3374,16\*B9S7,16»CF1F,16\*5507,

16»3F73,16\*A5AB, 16»03E3,16\*A9AC,16»0FE4,

16\*238F, 16\*5900, 16\*2F88,16»B550, 16\*0318,

16\*4 ICE,16\*3786,16\*AD5E, 16\*DB16, 16\*A159, 16\*0711,16\*4009,16»3B81,16\*7102, 16I079A,

16\*9042, 16\*EB0A, 16\*9145, 164Б70О, 16\*7005, 16\*0B9O, 16\*894B,16\*FF03,16»650B, 16\*1393,

16\*6900,16\*1F94,16\*8540,16\*F304,16\*11BA,

16\*10E0,1646BA5,164E973,16\*9F3B,16405E3,

16\*096F,16\*AF27,16\*35FF,16»43B7,16\*39F8,

16»67A2,16»FD7A, 16»8B32,16\*F17D, 16\*8735,

16\*73AB, 16409E4, 16\*7FAC, 16»E574, 16\*9330,

16»4FB0,16\*0568,164A320,16\*21F6,16\*57BE,

16\*0066,16\*BB2E, 16\*0161, 16»B729, 16\*20F1, 16»5BB9;

81

#### ГОСТ Р МЭК 61784-3-12—2016

ПриложениеВ (справочное)

Информация для оценки профилей коммуникаций функциональной безопасности CPF 12

Информация о тестовых лабораториях, в которых испытывают и подтверждают соответствие изделий FSCP 8/1 стандарту МЭК 61784-3-12. может быть получена в национальных комитетах МЭК или в следующих организациях:

EtherCAT Technology Group Ostendstrasse 196

90482 Nuremberg GERMANY

Телефон: +49-911-54056-20 Факс: \*49-911-54056-29

E-matf: [info@ethercat.org](mailto:info@ethercat.org) URL: wwiv.ethercaLorg

82

#### ГОСТ Р МЭК 61784-3.12—2016

ПриложениеДА (справочное)

Сведения о соответствии ссылочных международных стандартов национальным стандартам

Таблица ДА.1

|  |  |  |
| --- | --- | --- |
| Обозначение ссылочного международного стандарта | Степень соответствия | Обозначение и наныеиоеание соответствующего национального стан­ дарта |
| IEC 60204\*1 | IDT | ГОСТ Р МЭК 60204-1—2007 «Безопасность машин. Электрообо­ рудование машин и механизмов. Часть 1. Общие требования» |
| IEC 61000-6-2 | MOD | ГОСТ Р 51317.6.2—2007 (МЭК 61000-6-2—2005) «Совмести­  мость технических средств электромагнитная. Устойчивость к электромагнитным помехам технических средств, применяемых в промышленных зонах. Требования и методы испытаний» |
| IEC 61131-2 | ГОТ | ГОСТ IEC 61131-2—2012 «Контроллеры программируемые. Часть 2. Требования к оборудованию и испытания» |
| IEC 61158-2 | — | В |
| IEC 61158-3-12 | — | в |
| IEC 61158-4-12 | — | в |
| IEC 61158-5-12 | — | • |
| IEC 61158-5-10 | — | в |
| IEC 61158-6-12 | — | в |
| IEC 61326-3-1 | — | в |
| IEC 61326-3-2 | — | в |
| IEC 61508 (все части) | ГОТ | ГОСТ Р МЭК 61508—2012 (все части). «Функциональная безопас­ ность систем электрических, электронных, программируемых электронных, связанных с безопасностью» |
| IEC 61784-2 | — | • |
| IEC 61784-3 | — | в |
| IEC 61918 | — | в |
| \* Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать | | |
| перевод на русский язык данного международного стандарта. | | |
| Примечание — В настоящей таблице использованы следующие условте обозначения степени соответствия стандартов:   * IDT — идентичные стандарты: * MOD — модифицированный стандарт. | | |

83

#### ГОСТ Р МЭК 61784-3-12—2016

Библиография

1. IEC 60050 (al parts). International Electrotechnical Vocabulary

Примечание — См. также IEC Multilingual Dictionary — Electricity, Electronics and Telecommunications (до­ ступен на CD-ROM и no адресу <[http://www.electropedia.org](http://www.electropedia.org/)>).

1. IEC/TS 61000-1-2, Electromagnetic compatibility (EMC)— Part 1-2: General —Methodology tor the achievement of the functional safety of electrical and electronic equipment with regard to electromagnetic phenomena
2. IEC 61131-6, Programmable controllers — Part 6: Functional safety
3. IEC 61156 (all parts), Industrial communication networks — Fiel&us specifications
4. IEC 61496 (al parts). Safety of machinery — Electro-sensitive protective equipment
5. IEC 61508-1:2010, Functional safety of electricat/electronic/programmable electronic safety-related systems —

Part 1: General requirements

1. IEC 61508-4:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems —

Part 4: Definitions and abbreviations

1. IEC 61508-5:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems —

Part 5: Examples of methods for the determination of safety integrity levels

1. IEC 61511 (all parts), Functional safety — Safety instrumented systems for the process industry sector
2. IEC 61784-1. Industrial communication networks — Profiles — Pari 1: Fietdbus profiles
3. IEC 61784-4, Industrial communication networks — Profiles — Part 4: Secure communications for fieldbuses
4. IEC 61784-5 (all parts). Industrial communication networks — Profiles — Part 5: Instalation of fieldbuses — Instal­ lation profiles for CPF x
5. IEC 61800-5-2, Adjustable speed electrical power drive systems — Part 5-2: Safety requirements — Functional
6. IEC/TR 62059-11, Electricity metenng equipment — Dependability — Part 11: General concepts
7. IEC 62061, Safety of machinery — Functional safety of safety-related electrical, electronic and programmable elec­ tronic control systems
8. IEC/TR 62210, Power system control and associated communications — Data and communication security
9. IEC 62280-1, Railway applications — Communication, signalling and processing systems— Part 1: Safety-related communication in closed transmission systems
10. IEC 62280-2. Railway applications — Communication, signalling and processing systems — Part 2: Safety-related communication in open transmission systems
11. IEC 62443 (al parts). Industrial communication networks — Network and system security
12. ISQ/IEC Guide 51:1999, Safety aspects — Guidelines for the«r inclusion in standards
13. ISO/IEC 2382-14, Information technology — Vocabulary — Part 14: Reliability, maintainability and availability
14. ISO/IEC 2382-16, Information technology — Vocabulary — Pari 16: Information theory
15. ISO/IEC 7498 (all parts). Information technology — Open Systems Interconnection — Basic Reference Model
16. ISO/IEC 19501, Information technology — Open Distributed Processing — Unrfied Modeling Language (UML) Ver­ sion 1.4.2
17. ISO 10218-1. Robots for industrial environments— Safety requirements — Pari 1: Robot
18. ISO 12100-1. Safety of machinery — Basic concepts, general principles for design — Part 1: Basic terminology, methodology
19. ISO 13849-1. Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design
20. ISO 13849-2. Safety of machinery — Safety-related parts of control systems — Pari 2: Validation
21. ISO 14121. Safety of machinery — Principles of risk assessment
22. EN 954-1:1996, Safety of machinery — Safety related parts of control systems — General principles for design
23. ANSl/ISA-84.00.01-2004 (ail parts). Functional Safety: Safety Instrumented Systems for the Process Industry Sector
24. VDI/VDE 2180 (aU parts). Safeguarding of industrial process plants by means of process control engineering
25. GS-ET-26. Grundsatz fur die Priifung und Zertifizierung von Bussystemen fur die Obertragung sicherheitsrelevanter Nachnchten. May 2002. HVBG. Guslav-Heinemann-Ufer 130. D-50968 Koln (’Principles for Test and Certification of Bus Systems for Safety relevant Communication’)
26. ANDREW S. TANENBAUM. Computer Networks. 4th Edition. Prentice Hal. N.J.. ISBN-10:0130661023. ISBN-13: 978-0130661029
27. W. WESLEY PETERSON. Error-Correcting Codes. 2nd Edition 1981, MIT-Press. ISBN 0-262-16-039-0
28. BRUCE P. DOUGLASS. Doing Hard Time. 1999. Addison-Wesley. ISBN 0-201-49837-5
29. New concepts for safety-related bus systems. 3rd International Symposium ’Programmable Electronic Systems in Safety Related Applications\*. May 1998, from Dr. Michael Schdfer. BG-lnstitute for Occupational Safety and Health.
30. DIETER CONRADS. Datenkommunikation. 3rd Edition 1996. Vieweg. ISBN 3-528-245891
31. German IEC subgroup DKE AK 767.0.4: EMC and Functional Safety. Spnng 2002
32. NFPA79 (2002). Electrical Standard for Industrial Machinery
33. GUY E. CASTAGNOLl, On the Minimum Distance of Long Cyclic Codes and Cyclic Redundancy-Check Codes. 1989, Dissertation No. 8979 of ETH Zurich. Switzerland

84

#### ГОСТ Р МЭК 61784-3-12—2016

(42] GUY Е. CASTAGNCHJ. STEFAN BRAUER. and MARTIN HERRMANN. Optimization of Cyclic Redundancy-Check Codes with 24 and 32 Parity Bits. June 1993. IEEE Transactions On Communications. Volume 41. No. 6

(43] SCHILLER F and MATTES T: An Efficient Method to Evaluate CRC-Polynomials for Safety-Critical Industrial Com­ munication. Journal of Applied Computer Science. Vol. 14. No 1. pp. 57-80. Technical University Press. Lodz,Poland. 2006

(44] SCHILLER F and MATTES T: Analysis of CRC-potynomials for Safety-critical Communication by Deterministic and Stochastic Automata. 6th IFAC Symposium on Fauft Detection. Supervision and Safety for Technical Processes. SAFEPROCESS 2006. pp. 1003-1008. Beijing. China. 2006

85

#### ГОСТ Р МЭК 61784-3-12»2016

УДК 62-783:614.8:331.454:006.354 ОКС 13.110 Т51

Ключевые слова: промышленные сети, профили, функциональная безопасность полевых шин. спе­ цификации для CPF 12

Редактор *АН. Рубин*

Технический редактор *В.Н. Прусакова*

Корректор *М.В. Бучмая*

Компьютерная верстка *Е.*О. *Асташина*

Сдано в набор 22.12.2016. Подписано в печать 16.01.2017. Формат 60«64’/g. Гарнитура Ариел Уел. печ. п. 10.70. Уч.-иэд. п. 0.68. Тираж 27 кз. Эак. 76

Подготоапено иа основе электронной версии, предоставленной разработчиком стандарта

Издано и отпечатано во ФГУП «СТАНДЛРТИКФОРМ», 12399S Москва. Гранатный пер.. 4 [www.gostinfb.m](http://www.gostinfb.m/) [info@gostinfo.iu](mailto:info@gostinfo.iu)