ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 24709-2–2011

Информационные технологии
БИОМЕТРИЯ
Испытания на соответствие биометрическому программному интерфейсу (БиоАПИ)
Часть 2
Тестовые утверждения для поставщиков биометрических услуг

ISO/IEC 24709-2:2007
Information technology – Conformance testing for the biometric application programming interface (BioAPI) – Part 2:Test assertions for biometric service providers

(IDT)

Издание официальное

Москва
Стандартинформ
2014

gost.e-ecolog.ru

Предисловие

Цели и принципы стандартизации в Российской Федерации установлены Федеральным законом от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании», а правила применения национальных стандартов Российской Федерации – ГОСТ Р 1.0 – 2004 «Стандартизация в Российской Федерации. Основные положения».

Сведения о стандарте

1 ПОДГОТОВЛЕН Научно-исследовательским и испытательным центром биометрической техники Московского государственного технического университета имени Н.Э. Баумана (НИИЦ БТ МГТУ им. Н.Э. Баумана) на основе собственного аутентичного перевода на русский язык стандарта, указанного в пункте 4, при консультативной поддержке Ассоциации автоматической идентификации «ЮНИСКАН/ГС1 РУС»

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

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 21.09.2011 г №324-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 24709-2:2007 «Информационные технологии. Испытания на соответствие биометрическому программному интерфейсу (БиоАПИ). Часть 2. Тестовые утверждения для поставщиков биометрических услуг» (ISO/IEC 24709-2:2007 Information technology – Conformance testing for the biometric application programming interface (BioAPI). Part 2: Test assertions for biometric service providers»).

Наименование настоящего стандарта изменено относительно наименования указанного международного стандарта для приведения в соответствие с ГОСТ Р 1.5–2004 (подраздел 3.5).

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

5 ВВЕДЕН ВПЕРВЫЕ

6 Следует обратить внимание на то, что некоторые элементы настоящего стандарта могут быть объектами получения патентных прав. ИСО и МЭК не несут ответственность за установление подлинности каких–либо или всех подобных патентных прав

Информация об изменениях к настоящему стандарту публикуется в ежегодно издаваемом информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячно издаваемых информационных указателях «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии

© Стандартинформ, 2012

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

Содержание

1 Область применения
2 Соответствие
3 Нормативные ссылки
4 Термины и определения
5 Обозначения и сокращения
6 Общие положения
7 ИспытаниеПБУ на соответствиеБиоАПИ
7.1 Основные положения
7.2 Испытание ПБУ подкласса ПБУ верификации
7.3 Испытание ПБУ подкласса ПБУ идентификации
7.4 Испытание ПБУ подкласса ПБУ получения данных
7.5 Испытание ПБУ подкласса механизм верификации
7.6 Испытание ПБУ подкласса механизм идентификации
8 Тестовые утверждения
8.1 Основные положения
8.2 Общие процессы
8.3 Утверждение 1a BioSPI_BSPLoad_InvalidUUID
8.4 Утверждение 1b BioSPI_BSPLoad_ValidParam
8.5 Утверждение 2a BioSPI_BSPUnload_ValidParam
8.6 Утверждение 2b BioSPI_BSPUnload_InvalidUUID
8.7 Утверждение 2c BioSPI_BSPUnload_UnmatchedLoad
8.8 Утверждение 2d BioSPI_BSPUnload_Confirm
8.9 Утверждение 3a BioSPI_BSPAttach_ValidParam
8.10 Утверждение 3b BioSPI_BSPAttach_InvalidUUID
8.11 Утверждение 3c BioSPI_BSPAttach_InvalidVersion
8.12 Утверждение 3d BioSPI_BSPAttach_InvalidBSPHandle
8.13 Утверждение 4a BioSPI_BSPDetach_ValidParam
8.14 Утверждение 4b BioSPI_BSPDetach_InvlidBSPHandle
8.15 Утверждение 4c BioSPI_BSPDetach_Confirm

8.16 Утверждение 5a BioSPI_FreeBIRHandle_ValidParam
8.17 Утверждение 5b BioSPI_FreeBIRHandle_InvalidBSPHandle
8.18 Утверждение 5c BioSPI_FreeBIRHandle_InvalidBIRHandle
8.19 Утверждение 6a BioSPI_GetBIRFromHandle_ValidParam
8.20 Утверждение 6b BioSPI_GetBIRFromHandle_InvalidBSPHandle
8.21 Утверждение 6c BioSPI_GetBIRFromHandle_InvalidBIRHandle
8.22 Утверждение 7a BioSPI_GetHeaderFromHandle_ValidParam
8.23 Утверждение 7b BioSPI_GetHeaderFromHandle_InvalidBSPHandle
8.24 Утверждение 7c BioSPI_GetHeaderFromHandle_InvalidBIRHandle
8.25 Утверждение 7d BioSPI_GetHeaderFromHandle_BIRHandleNotFreed
8.26 Утверждение 8a BioSPI_EnableEvents_ValidParam
8.27 Утверждение 8b BioSPI_EnableEvents_InvalidBSPHandle
8.28 Утверждение 9a BioSPI_Capture_AuditData
8.29 Утверждение 9b BioSPI_Capture_ReturnQuality
8.30 Утверждение 9c BioSPI_Capture_IntermediateProcessedBIR
8.31 Утверждение 9d BioSPI_Capture_InvalidBSPHandle
8.32 Утверждение 10a BioSPI_CreateTemplate_PayloadSupported
8.33 Утверждение 10b BioSPI_CreateTemplate_BIRHeaderQuality
8.34 Утверждение 10c BioSPI_CreateTemplate_OutputBIRDataType
8.35 Утверждение 10d BioSPI_CreateTemplate_OutputBIRPurpose
8.36 Утверждение 10e BioSPI_CreateTemplate_InputBIRDataType
8.37 Утверждение 10f BioSPI_CreateTemplate_Inconsistent_Purpose
8.38 Утверждение 11a BioSPI_Process_ValidParam
8.39 Утверждение 11b BioSPI_Process_BIRHeaderQuality
8.40 Утверждение 11c BioSPI_Process_OutputBIRPurpose
8.41 Утверждение 11d BioSPI_Process_BuildsProcessedBIR
8.42 Утверждение 11e BioSPI_Process_InputBIRDataType
8.43 Утверждение 12a BioSPI_VerifyMatch_ValidParam
8.44 Утверждение 12b BioSPI_VerifyMatch_Payload
8.45 Утверждение 12c BioSPI_VerifyMatch_Inconsistent_Purpose

8.46 Утверждение 13a BioSPI_Enroll_ValidParam
8.47 Утверждение 13b BioSPI_Enroll_Payload
8.48 Утверждение 13c BioSPI_Enroll_AuditData
8.49 Утверждение 13d BioSPI_Enroll_BIRHeaderQuality
8.50 Утверждение 14a BioSPI_Verify_ValidParam
8.51 Утверждение 14b BioSPI_Verify_Payload
8.52 Утверждение 14c BioSPI_Verify_AuditData
8.53 Утверждение 15a BioSPI_DbOpen_ValidParam
8.54 Утверждение 15b BioSPI_DbOpen_InvalidBSPHandle
8.55 Утверждение 16a BioSPI_DbClose_ValidParam
8.56 Утверждение 16b BioSPI_DbClose_InvalidBSPHandle
8.57 Утверждение 17a BioSPI_DbCreate_DbProtected
8.58 Утверждение 17b BioSPI_DbCreate_ValidParam
8.59 Утверждение 17c BioSPI_DbCreate_InvalidBSPHandle
8.60 Утверждение 18a BioSPI_DbDelete_InvalidBSPHandle
8.61 Утверждение 18b BioSPI_DbDelete_OpenDbProtected
8.62 Утверждение 18c BioSPI_DbDelete_ValidParam
8.63 Утверждение 19a BioSPI_DbSetMarker_ValidParam
8.64 Утверждение 19b BioSPI_DbSetMarker_InvalidBSPHandle
8.65 Утверждение 19c BioSPI_DbSetMarker_RecordNotFound
8.66 Утверждение 20a BioSPI_DbFreeMarker_ValidParam
8.67 Утверждение 20b BioSPI_DbFreeMarker_InvalidBSPHandle
8.68 Утверждение 20c BioSPI_DbFreeMarker_InvalidMarker
8.69 Утверждение 21a BioSPI_DbStoreBIR_ValidParam
8.70 Утверждение 21b BioSPI_DbStoreBIR_InvalidBSPHandle
8.71 Утверждение 22a BioSPI_DbGetBIR_ValidParam
8.72 Утверждение 22b BioSPI_DbGetBIR_InvalidBSPHandle
8.73 Утверждение 22c BioSPI_DbGetBIR_RecordNotFound
8.74 Утверждение 23a BioSPI_DbGetNextBIR_ValidParam
8.75 Утверждение 23b BioSPI_DbGetNextBIR_InvalidBSPHandle

8.76 Утверждение 24a BioSPI_DbDeleteBIR_ValidParam .....
8.77 Утверждение 24b BioSPI_DbDeleteBIR_InvalidBSPHandle .....
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации .....

Введение

Настоящий стандарт устанавливает несколько тестовых утверждений, написанных на языке утверждений, определенном в ИСО/МЭК 24709-1:2007 «Информационные технологии. Испытания на соответствие биометрическому программному интерфейсу (БиоАПИ). Часть 1: Методы и процедуры». Данные утверждения позволяют пользователю настоящего стандарта (например, испытательной лаборатории) проводить испытания любого поставщика биометрических услуг (ПБУ) на соответствие ИСО/МЭК 19784-1 «Информационные технологии. Биометрический программный интерфейс. Часть 1. Спецификация биометрического программного интерфейса» (спецификации БиоАПИ 2.0).

Структура тестовых утверждений, приведенных в настоящем стандарте, соответствует указанной в приложении А ИСО/МЭК 19784-1, в котором определены параметры соответствия БиоАПИ для различных типов реализаций (ПБУ, инфраструктуры и приложения) и для ПБУ, принадлежащих отдельным подклассам соответствия.

Настоящий стандарт устанавливает тестовые утверждения, предназначенные для проверки на соответствие ПБУ всех подклассов соответствия. В дальнейшем тестовые утверждения могут быть сгруппированы в соответствии с подклассами (если ПБУ принадлежит к одному или нескольким подклассам) и конкретной функциональностью.

Каждое тестовое утверждение использует одну или более (как можно более простую) функцию испытуемой реализации. Тестовые утверждения помещают в пакеты (в пакете должно быть одно или более утверждений) в соответствии с требованиями языка тестовых утверждений.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные технологии
БИОМЕТРИЯ
Испытания на соответствие
биометрическому программному интерфейсу (БиоАПИ)
Часть 2

Тестовые утверждения для поставщиков биометрических услуг

Information technology.
Biometrics.
Conformance testingfor the biometric application programming interface.
Part 2.
Test assertions for biometric service providers

Дата введения – 01-07 -2013

1 Область применения

Настоящий стандарт устанавливает ряд тестовых утверждений, написанных на языке, определенном в ИСО/МЭК 24709-1. Настоящий стандарт устанавливает по пять подмножеств тестовых утверждений для каждого из пяти подклассов соответствия ПБУ, соответствующих приведенным в ИСО/МЭК 19784-1. Кроме того, в настоящем стандарте приведены дополнительные тестовые утверждения, которые должны быть выполнены в зависимости от наличия дополнительных функций, определенных в спецификации БиоАПИ 2.0, поддержку которых должна обеспечить испытуемая реализация.

Перечень утверждений проверки, установленный в настоящем стандарте, не является исчерпывающим (см. также ИСО/МЭК 24709-1, раздел 6). Реализации ПБУ, подвергаемые испытанию с использованием метода, определенного в ИСО/МЭК 24709-1, и на соответствие тестовым утверждениям, установленным в настоящем стандарте, соответствуют только тем требованиям ИСО/МЭК 19784-1, для которых в настоящем стандарте приведены тестовые утверждения.

2 Соответствие

Реализации (наборы тестов при испытании на соответствие БиоАПИ), соответствующие требованиям настоящего стандарта, должны обеспечивать воз-

Издание официальное

можность обработки всех тестовых утверждений, указанных в разделе 8, с использованием методики, приведенной в ИСО/МЭК 24709-1, а также правил и условий, установленных в разделах 6 и 7 настоящего стандарта.

3 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие международные стандарты. В случае ссылок на документы, у которых указана дата утверждения, необходимо использовать только указанную редакцию. В случае, когда дата утверждения не указана, следует использовать последнюю редакцию ссылочных стандартов, включая любые поправки и изменения к ним:

ИСО/МЭК 19784-1:2006, Информационные технологии – Биометрический программный интерфейс – Часть 1: Спецификация биометрического программного интерфейса

ИСО/МЭК 24709-1:2007, Информационные технологии – Испытания на соответствие биометрическому программному интерфейсу (БиоАПИ) –Часть 1: Методы и процедуры

4 Термины и определения

В настоящем стандарте использованы термины и определения, установленные в ИСО/МЭК 24709-1.

5 Обозначения и сокращения

В настоящем стандарте использованы обозначения и сокращения, установленные в ИСО/МЭК 24709-1.

6 Общие положения

6.1 Тестовые утверждения, указанные в разделе 7 и определенные в разделе 8, основаны на методике испытания на соответствие, установленной в ИСО/МЭК 24709-1, и могут применяться только в соответствии с данной методикой.

6.2 В разделе 6 ИСО/МЭК 24709-1 приведены три модели испытаний на соответствие:

а) модель испытания на соответствие приложения БиоАПИ;

б) модель испытания на соответствие инфраструктуры БиоАПИ;

в) модель испытания на соответствие ПБУ БиоАПИ.

6.3 Каждую модель используют для испытания одного из трех стандартных компонентов БиоАПИ (см. ИСО/МЭК 24709-1, раздел 6). В разделе 8 настоящего стандарта приведены тестовые утверждения, применяемые в модели испытания на соответствие ПБУ БиоАПИ. Настоящий стандарт не распространяется на модели испытания на соответствие для приложения БиоАПИ и инфраструктуры БиоАПИ, а также соответствующие тестовые утверждения.

7 ИспытаниеПБУ на соответствие БиоАПИ

7.1 Основные положения

7.1.1 Для проведения испытания ПБУ на соответствие БиоАПИ изготовитель ПБУ должен предоставить подтверждение соответствия биометрическому программному интерфейсу (ПСБПИ), указанное в подразделе 6.1 ИСО/МЭК 24709-1 и содержащее информацию, приведенную в таблице 1.

Таблица 1 – Подтверждение соответствия биометрическому программному интерфейсу

Контактные данные предприятия-изготовителя:	
наименование	
Адрес:	
улица	
город	

Продолжение таблицы 1

Контактные данные предприятия-изготовителя:	
штат или область (край)	
почтовый индекс	
страна	
Телефон	
Биометрический продукт:	
наименование	
серийный номер	
описание	
Класс соответствия БиоАПИ (один из следующих):	
приложение БиоАПИ	
инфраструктура БиоАПИ	
ПБУ(должен быть выбран этот вариант)	X
Подкласс соответствия ПБУ (один из следующих):	
ПБУ верификации	
ПБУ идентификации	
ПБУ получения данных	
механизм верификации	
механизм идентификации	
Дополнительные функции, поддерживаемые ПБУ (одна или несколько из следующих):	
BioSPI_Capture	
BioSPI_CreateTemplate	
BioSPI_Process	
BioSPI_VerifyMatch	
BioSPI_Enroll	
BioSPI_Verify	
BioSPI_Identify	
Дополнительные свойства, поддерживаемые ПБУ (одно или несколько из следующих):	
база данных, контролируемая ПБУ	
адаптация шаблона	
генерирование событий SOURCE PRESENT	
установка значения поля «Качество» в промежуточной ЗБИ	

Окончание таблицы 1

Контактные данные предприятия-изготовителя:	
установка значения поля «Качество» в обработанной ЗБИ	
Дополнительная информация:	
УУИД ПБУ	
версия спецификации БиоАПИ	
версия продукта	
поддерживаемые форматы блока биометрических данных (владелец формата и тип формата)	
поддерживаемые биометрические типы (один или более)	
максимальный поддерживаемый объем полезных данных	

7.1.2 В таблице 2 приведен перечень тестовых утверждений, которые должны быть подтверждены для определения того, противоречит ли испытуемая реализация ПБУ БиоАПИ требованиям, указанным в ИСО/МЭК 19784-1, приложение А, раздел А.4.

Примечание – Успешная обработка всех применимых тестовых утверждений является свидетельством соответствия испытуемой реализации требованиям ИСО/МЭК 19784-1, однако не устанавливает соответствия, поскольку тестовые утверждения не являются исчерпывающим испытанием на соответствие (см. ИСО/МЭК 24709-1, раздел 6).

7.1.3 Данные утверждения используют модель испытания на соответствие ПБУ БиоАПИ (см. ИСО/МЭК 24709-1, раздел 6) и определены в разделе 8 настоящего стандарта.

7.1.4 В подразделах 7.2 – 7.6 приведены особые условия для каждого подкласса соответствия и установлен набор тестовых утверждений, которые должны быть подтверждены при проведении испытания на соответствие ПБУ определенного подкласса.

Таблица 2 – Тестовые утверждения для ПБУ БиоАПИ

Номер	Наименование утверждения	Номер пункта настоящего стандарта	Назначение
1a	*BioSPI_BSPLoad_InvalidUUID*	9.3.1.1, 11.2.3	Проверка возвращения кода ошибки BioAPIERR_H_FRAMEWORK_ INVALID_UUID при вызове функции *BioSPI_BSPLoad* с недостоверным значением входного параметра УУИД
1b	*BioSPI_BSPLoad_ValidParam*	9.3.1.1, A.4	Проверка возвращения значения BioAPI_OK при вызове функции *BioSPI_BSPLoad* с достоверными значениями входных параметров
2a	*BioSPI_BSPUnload_ValidParam*	9.3.1.2, A.4	Проверка возвращения значения BioAPI_OK при вызове функции *BioSPI_BSPUnload* с достоверными значениями входных параметров
2b	*BioSPI_BSPUnload_InvalidUUID*	9.3.1.2, 11.2.3	Проверка возвращения кода ошибки BioAPIERR_INVALID_UUID при вызове функции *BioSPI_BSPUnload* с недостоверным значением входного параметра УУИД
2c	*BioSPI_BSPUnload_Unmatched Load*	9.3.1.2, 8.1.6.1	Проверка возвращения кода ошибки BioAPIERR_BSP_NOT_LOADED при вызове функции *BioSPI_BSP-Unload* без соответствующего вызова функции *BioSPI_BSPLoad*
2d	*BioSPI_BSPUnload_Confirm*	9.3.1.2	Проверка выгружения ПБУ при вызове функции *BioSPI_BSP-Unload*
3a	*BioSPI_BSPAttach_ValidParam*	9.3.1.3, A.4	Проверка возвращения значения BioAPI_OK при вызове функции *BioSPI_BSPAttach* с достоверными значениями входных параметров

Продолжение таблицы 2

Номер	Наименование утверждения	Номер пункта настоящего стандарта	Назначение
3b	*BioSPI_BSPAttach_InvalidUUID*	9.3.1.3, 11.2.3	Проверка возвращения кода ошибки BioAPIERR_INVALID_UUID при вызове функции *BioSPI_BSPAttach* с недостоверным значением входного параметраУУИД
3c	*BioSPI_BSPAttach_InvalidVersion*	9.3.1.3, 11.2.3	Проверка возвращения кода ошибки BioAPIERR_INCOMPATIBLE_VERSION при вызове функции *BioSPI_BSPAttach*, если вызывающая функция определяет несовместимую версию спецификации БиоАПИ
3d	*BioSPI_BSPAttach_InvalidBSP-Handle*	9.3.1.3	Проверка возвращения сообщения об ошибке при вызове функции *SPI_BSPAttach*, если вызов выполняется с недостоверным дескриптором ПБУ
4a	*BioSPI_BSPDetach_ValidParam*	9.3.1.4, А.4	Проверка возвращения значения BioAPI_OK при вызове функции *BioSPI_BSPDetach* с достоверным дескриптором модуля
4b	*BioSPI_BSPDetach_InvalidBSP-Handle*	9.3.1.4, 8.1.8.1	Проверка возвращения сообщения об ошибке при вызове функции *BioSPI_BSPDetach* с недостоверным дескриптором модуля
4c	*BioSPI_BSPDetach_Confirm*	9.3.1.4, 8.1.8.4	Проверка прерывания сессии присоединения при вызове функции *Bio-SPI_BSPDetach*
5a	*BioSPI_FreeBIRHandle_Valid Param*	9.3.2.1, 8.2.1, А.4	Проверка высвобождения дескриптора ЗБИ при вызове функции *BioSPI_FreeBIRHandle* с достоверным дескриптором ЗБИ

Продолжение таблицы 2

Номер	Наименование утверждения	Номер пункта настоящего стандарта	Назначение
5b	*BioSPI_FreeBIRHandle_Invalid BSP-Handle*	9.3.2.1, 8.2.1	Проверка возврата ошибки при вызове функции *BioSPI_FreeBIRHandle* с недостоверным дескриптором ПБУ
5c	*BioSPI_FreeBIRHandle_Invalid BIR-Handle*	9.3.2.1, 8.2.1	Проверка возвращения ошибки при вызове функции *BioSPI_FreeBIR-Handle* с недостоверным дескриптором ЗБИ
6a	*BioSPI_GetBIRFromHandle_ Valid-Param*	9.3.2.2, 8.2.2, А.4	Проверка возврата значения BioAPI_OK при вызове функции *BioSPI_GetBIRFromHandle* с достоверными значениями входных параметров
6b	*BioSPI_GetBIRFromHandle_ Invalid-BSPHandle*	9.3.2.2, 8.2.2	Проверка возврата сообщения об ошибке при вызове функции *BioSPI_GetBIRFromHandle* с недостоверным дескриптором ПБУ
6c	*BioSPI_GetBIRFromHandle_ Invalid-BIRHandle*	9.3.2.2, 8.2.2	Проверка возврата сообщения об ошибке при вызове функции *BioSPI_GetBIRFromHandle* с недостоверным дескриптором ЗБИ
7a	*BioSPI_GetHeaderFromHandle_ ValidParam*	9.3.2.3, 8.2.3, А.4	Проверка возврата значения BioAPI_OK при вызове функции *BioSPI_ GetHeaderFromHandle* с достоверными значениями входных параметров
7b	*BioSPI_GetHeaderFromHandle_ In-validBSPHandle*	9.3.2.3, 8.2.3	Проверка возврата сообщения об ошибке при вызове функции *BioSPI_GetHeaderFromHandle* с недостоверным дескриптором модуля
7c	*BioSPI_GetHeaderFromHandle_ In-validBIRHandle*	9.3.2.3, 8.2.3	Проверяет, возвращается ли сообщение об ошибке при вызове функции *BioSPI_GetHeaderFrom-Handle* с недостоверным дескриптором ЗБИ

Продолжение таблицы 2

Номер	Наименование утверждения	Номер пункта настоящего стандарта	Назначение
7d	*BioSPI_GetHeaderFromHandle_ BIR-HandleNotFreed*	9.3.2.3, 8.2.3	Проверка невысвобождения дескриптора ЗБИ после вызова функции *BioSPI_GetHeaderFromHandle*
8a	*BioSPI_EnableEvents_ValidParam*	9.3.3.1, 8.3.1	Проверка функции *BioSPI_ EnableEvents* с достоверными значениями входных параметров
8b	*BioSPI_EnableEvents_Invalid BSP-Handle*	9.3.3.1, 8.3.1	Проверка функции *Bio-SPI_Enable Events* с недостоверным дескриптором модуля
9a	*BioSPI_Capture_AuditData*	9.3.4.1, 8.4.1.1, 7.47, A.4.6.2.1	Проверка функции *Bio-SPI_Capture* с установленным значением параметра AuditData
9b	*BioSPI_Capture_ReturnQuality*	9.3.4.1, 8.4.1, 7.49, 7.47, A.4.6.2.2	Проверканаличия в заголовке полученной ЗБИ достоверного значения качества (в диапазоне от 0 до 100)
9c	*BioSPI_Capture_Intermediate-ProcessedBIR*	9.3.4.1, 8.4.1.1, 8.4.1.2	Проверка степени обработки (INTERMEDIATE – промежуточная или PROCESSED – обработана) у ЗБИ, возвращенной функцией *BioSPI_Capture*
9d	*BioSPI_Capture_InvalidBSPHandle*	9.3.4.1, 8.4.1.2	Проверка возвращения кода ошибки BioAPIERR_INVALID_BSP_HANDLE при вызове функции *BioSPI_Capture* с недостоверным дескриптором ПБУ
10a	*BioSPI_CreateTemplate_ Payload-Supported*	9.3.4.2, 8.4.5.1, 7.47, A.4.6.2.6	Проверка функции *BioSPI_Create Template* с достоверными значениями параметров и данных
10b	*BioSPI_CreateTemplate_BIR-HeaderQuality*	9.3.4.2, 7.49.3, 7.47, A.4.6.2.2	Проверка функции *Bio-SPI_CreateTemplate* с достоверными значениями параметров и возвращенным значением качества

Продолжение таблицы 2

Номер	Наименование утверждения	Номер пункта настоящего стандарта	Назначение
10c	*BioSPI_CreateTemplate_Output BIR-DataType*	9.3.4.2, 8.4.2.1	Проверка функции *Bio-SPI_Create Template* с достоверными параметрами. Разрабатываемая ЗБИ шаблона должна иметь степень обработки PROCESSED
10d	*BioSPI_CreateTemplate_Output BIR-Purpose*	9.3.4.2, 8.4.2.1	Проверка совпадения назначения созданного шаблона с назначением полученной ЗБИ
10e	*BioSPI_CreateTemplate_InputBIR-DataType*	9.3.4.2, 8.4.2.1	Проверка функции *BioSPI_Create Template* с входной ЗБИ с недостоверной степенью обработки
10f	*BioSPI_CreateTemplate_ Inconsistent_Purpose*	9.3.4.2, 8.4.2.1	Проверка функции *Bio-SPI_Create Template* с недостоверным назначением входной ЗБИ
11a	*BioSPI_Process_ValidParam*	9.3.4.3, 8.4.3.1	Проверка функции *BioSPI_Process* с достоверными параметрами и возвращенным значением степени обработки
11b	*BioSPI_Process_BIRHeaderQuality*	9.3.4.3, 8.4.3.1, 7.49.3, 7.47, A.4.6.2.2	Проверка возвращаемого значения качества при вызове функции *BioSPI_Process* с достоверными значениями параметров
11c	*BioSPI_Process_OutputBIRPurpose*	9.3.4.3, 8.4.3.1, 7.12.3	Вызов функции *BioSPI_Process* с достоверными параметрами и проверка совпадения назначения обработанной ЗБИ с назначением захваченной ЗБИ
11d	*BioSPI_Process_BuildsProcessedBIR*	9.3.4.3, 8.4.3.1	Проверка функции *BioSPI_Process* с достоверными значениями параметров и возвращенным значением степени обработки

Продолжение таблицы 2

Номер	Наименование утверждения	Номер пункта настоящего стандарта	Назначение
11e	*BioSPI_Process_InputBIRData Type*	9.3.4.3, 8.4.3.1	Проверка функции *BioSPI_Process* с входной ЗБИ со степенью обработки PROCESSED. Такой вызов функции BioSPI_Process должен вызвать ошибку
12a	*BioSPI_VerifyMatch_ValidParam*	9.3.4.5, 8.4.5.1	Проверка возвращения значения BioAPI_OK при вызове функции *BioSPI_VerifyMatch* с достоверными значениями входных параметров
12b	*BioSPI_VerifyMatch_Payload*	9.3.4.5, 8.4.5.1, 7.47	Проверка поддержки наполнения данными функции *BioSPI_Verify Match*. Возвращаемым значением данной функции должно быть BioAPI_OK
12c	*BioSPI_VerifyMatch_Inconsistent_ Purpose*	9.3.4.5, 8.4.5.1, 11.2.3	Проверка функции *BioSPI_VerifyMatch* с ЗБИ, назначение которой является недопустимым для данной функции. Код ошибки должен быть значением BioAPIERR_BSP_INCONSISTENT_ PURPOSE
13a	*BioSPI_Enroll_ValidParam*	9.3.4.7, 8.4.7.1, A.4	Проверка возвращения значения BioAPI_OK при вызове функции *BioSPI_Enroll* с достоверными значениями входных параметров
13b	*BioSPI_Enroll_Payload*	9.3.4.7, 8.4.7.1, A.4.6.2.6, 7.47	Проверка возвращения значения BioAPI_OK при вызове функции *BioSPI_Enroll* с данными
13c	*BioSPI_Enroll_AuditData*	9.3.4.7, 8.4.7.1, 7.47, A.4.6.2.1	Проверка функции *BioSPI_Enroll* с установленным значением параметра AuditData. Функция должна возвратить контрольные данные в том случае, если ПБУ поддерживает возвращение контрольных данных

Продолжение таблицы 2

Номер	Наименование утверждения	Номер пункта настоящего стандарта	Назначение
13d	*BioSPI_Enroll_BIRHeaderQuality*	9.3.4.7, 8.4.7.1, 7.49.3, 7.47, A.4.6.2.2	Проверка возвращения достоверного значения качества в заголовке нового шаблона ЗБИ при вызове функции *BioSPI_Enroll*
14a	*BioSPI_Verify_ValidParam*	9.3.4.8, 8.4.8.1, A.4	Проверка возвращения значения BioAPI_OK при вызове функции *BioSPI_Verify* с достоверными значениями входных параметров
14b	*BioSPI_Verify_Payload*	9.3.4.8, 8.4.8.1, 7.47, A.4.6.2.6	Проверка возвращения значения BioAPI_OK при вызове функции *BioSPI_Verify* с данными
14c	*BioSPI_Verify_AuditData*	9.3.4.8, 8.4.8.1, 7.47, A.4.6.2.1	Проверка возвращения значения BioAPI_OK при вызове функции *BioSPI_Verify* с контрольными данными
15a	*BioSPI_DbOpen_ValidParam*	9.3.5.1, 8.5.1.1	Проверка функции *BioSPI_DbOpen* с достоверными значениями входных параметров
15b	*BioSPI_DbOpen_InvalidBSP Handle*	9.3.5.1, 8.5.1.1	Проверка функции *BioSPI_DbOpen* с недостоверным дескриптором ПБУ
16a	*BioSPI_DbClose_ValidParam*	9.3.5.2, 8.5.2.1	Проверка функции *BioSPI_DbClose* с достоверными значениями входных параметров
16b	*BioSPI_DbClose_InvalidBSP Handle*	9.3.5.2, 8.5.2.1	Проверка функции *BioSPI_ DbClose* с недостоверным дескриптором ПБУ
17a	*BioSPI_DbCreate_DbProtected*	9.3.5.3, 8.5.3.1	Двойной вызов функции *BioSPI_ DbCreate* и проверка возвращения сообщения об ошибке BioAPIERR_ DATABASE_ALREADY_EXISTS при повторном вызове
17b	*BioSPI_DbCreate_ValidParam*	9.3.5.3, 8.5.3.1	Проверка функции *BioSPI_ DbCreate* с достоверными значениями входных параметров

Продолжение таблицы 2

Номер	Наименование утверждения	Номер пункта настоящего стандарта	Назначение
17c	*BioSPI_DbCreate_InvalidBSP Handle*	9.3.5.3, 8.5.3.1	Проверка функции *BioSPI_ DbCreate* с недостоверным дескриптором ПБУ
18a	*BioSPI_DbDelete_InvalidBSP Handle*	9.3.5.4, 8.5.4.1	Проверка функции *BioSPI_ DbDelete* с недостоверным дескриптором ПБУ
18b	*BioSPI_DbDelete_OpenDb Protected*	9.3.5.4, 8.5.4.1	Проверка функции *BioSPI_ DbDelete*, при открытой базе данных
18c	*BioSPI_DbDelete_ValidParam*	9.3.5.4, 8.5.4.1	Проверка функции *BioSPI_ DbDelete* с достоверными значениями входных параметров
19a	*BioSPI_DbSetMarker_ValidParam*	9.3.5.5, 8.5.5.1	Проверка функции *BioSPI_DbSetMarker* с достоверными значениями входных параметров
19b	*BioSPI_DbSetMarker_InvalidBSP-Handle*	9.3.5.5, 8.5.5.1	Проверка функции *BioSPI_DbSetMarker* с недостоверным дескриптором ПБУ
19c	*BioSPI_DbSetMarker_Record Not-Found*	9.3.5.5, 8.5.5.1	Проверка функции *BioSPI_DbSet Marker* со значением первичного ключа, не существующим в базе данных
20a	*BioSPI_DbFreeMarker_ValidParam*	9.3.5.6, 8.5.6.1	Проверка функции *BioSPI_DbFree Marker* с достоверными значениями входных параметров
20b	*BioSPI_DbFreeMarker_Invalid BSP-Handle*	9.3.5.6, 8.5.6.1	Проверка функции *BioSPI_DbFree Marker* с недостоверным дескриптором ПБУ
20c	*BioSPI_DbFreeMarker_Invalid-Marker*	9.3.5.6, 8.5.6.1	Проверка функции *BioSPI_DbFree Marker* с недостоверным дескриптором маркера
21a	*BioSPI_DbStoreBIR_ValidParam*	9.3.5.7, 8.5.7.1	Проверка функции *BioSPI_DbStore BIR* с достоверными значениями входных параметров

Окончание таблицы 2

Номер	Наименование утверждения	Номер пункта настоящего стандарта	Назначение
21b	*BioSPI_DbStoreBIR_InvalidBSP-Handle*	9.3.5.7, 8.5.7.1	Проверка функции *BioSPI_DbStore BIR* с недостоверным дескриптором ПБУ
22a	*BioSPI_DbGetBIR_ValidParam*	9.3.5.8, 8.5.8.1	Проверка функции *BioSPI_DbGetBIR* с достоверными значениями входных параметров
22b	*BioSPI_DbGetBIR_InvalidBSP-Handle*	9.3.5.8, 8.5.8.1	Проверка функции *BioSPI_DbGetBIR* с недостоверным дескриптором ПБУ
22c	*BioSPI_DbGetBIR_RecordNot Found*	9.3.5.8, 8.5.8.1	Проверка функции *BioSPI_DbGetBIR* со значением первичного ключа, не входящим в базу данных
23a	*BioSPI_DbGetNextBIR_Valid Param*	9.3.5.9, 8.5.9.1	Проверка функции *BioSPI_DbGet NextBIR* с достоверными значениями входных параметров
23b	*BioSPI_DbGetNextBIR_Invalid BSP-Handle*	9.3.5.9, 8.5.9.1	Проверка функции *BioSPI_DbGet NextBIR* с недостоверным дескриптором ПБУ
24a	*BioSPI_DbDeleteBIR_ValidParam*	9.3.5.10, 8.5.10.1	Проверка функции *BioSPI_Db DeleteBIR* с достоверными значениями входных параметров
24b	*BioSPI_DbDeleteBIR_Invalid BSP-Handle*	9.3.5.10, 8.5.10.1	Проверка функции *BioSPI_Db DeleteBIR* с недостоверным дескриптором ПБУ

7.2 Испытание ПБУ подкласса ПБУ верификации

7.2.1 В настоящем подразделе приведены тестовые утверждения, которые следует использовать для испытания ПБУ верификации на соответствие требованиям ИСО/МЭК 19784-1, приложение А, Раздел А.4

7.2.2 Все ПБУ верификации должны быть подвергнуты испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 3, в указанном порядке.

Таблица 3 – Тестовые утверждения для испытания ПБУ верификации

Номер	Наименование утверждения	Пакет
1a	*BioSPI_BSPLoad_InvalidUUID*	020e90c8-0c19-1085-ab54-0002a5d5fd2e
1b	*BioSPI_BSPLoad_ValidParam*	01f6c6f0-0c19-1085-97fe-0002a5d5fd2e
2a	*BioSPI_BSPUnload_ValidParam*	01661010-0c22-1085-8688-0002a5d5fd2e
2b	*BioSPI_BSPUnload_InvalidUUID*	01c2e5b0-0c3b-1085-b31d-0002a5d5fd2e
2c	*BioSPI_BSPUnload_UnmatchedLoad*	02f6c618-0c23-1085-ba89-0002a5d5fd2e
2d	*BioSPI_BSPUnload_Confirm*	03daf040-0c3b-1085-a9fd-0002a5d5fd2e
3a	*BioSPI_BSPAttach_ValidParam*	00ae6488-0c3d-1085-9912-0002a5d5fd2e
3b	*BioSPI_BSPAttach_InvalidUUID*	049cc170-0c5f-1085-981f-0002a5d5fd2e
3c	*BioSPI_BSPAttach_InvalidVersion*	0052ac10-0c60-1085-9883-0002a5d5fd2e
3d	*BioSPI_BSPAttach_InvalidBSPHandle*	03826830-0c57-1085-bfb0-0002a5d5fd2e
4a	*BioSPI_BSPDetach_ValidParam*	00e0d2b0-0c7a-1085-b8ac-0002a5d5fd2e
4b	*BioSPI_BSPDetach_InvalidBSPHandle*	0434c458-0c79-1085-9f2c-0002a5d5fd2e
4c	*BioSPI_BSPDetach_Confirm*	002e7e58-0c78-1085-9e1d-0002a5d5fd2e
5a	*BioSPI_FreeBIRHandle_ValidParam*	0280a7d0-0c80-1085-a9a0-0002a5d5fd2e
5b	*BioSPI_FreeBIRHandle_InvalidBSPHandle*	047aed48-0c80-1085-898b-0002a5d5fd2e
5c	*BioSPI_FreeBIRHandle_InvalidBIRHandle*	018e6c18-0c9c-1085-afdf-0002a5d5fd2e
6a	*BioSPI_GetBIRFromHandle_ValidParam*	0460b658-0cb4-1085-a304-0002a5d5fd2e
6b	*BioSPI_GetBIRFromHandle_ InvalidBSPHandle*	02445668-0cc5-1085-a3ac-0002a5d5fd2e
6c	*BioSPI_GetBIRFromHandle_ InvalidBIRHandle*	0194a9c0-0cc7-1085-8780-0002a5d5fd2e
7a	*BioSPI_GetHeaderFromHandle_ ValidParam*	027a7db0-0cc7-1085-9391-0002a5d5fd2e
7b	*BioSPI_GetHeaderFromHandle_ InvalidBSPHandle*	057e0d38-0ccd-1085-83b8-0002a5d5fd2e

Окончание таблицы 3

Номер	Наименование утверждения	Пакет
7c	*BioSPI_GetHeaderFromHandle_ InvalidBIRHandle*	02195e68-0cce-1085-a46f-0002a5d5fd2e
7d	*BioSPI_GetHeaderFromHandle_ BIRHandleNotFreed*	01cc0988-0ccf-1085-a367-0002a5d5fd2e
8a	*BioSPI_EnableEvents_ValidParam*	0333f628-0ccf-1085-aceb-0002a5d5fd2e
8b	*BioSPI_EnableEvents_InvalidBSPHandle*	04ed0838-0ccf-1085-b64e-0002a5d5fd2e
13a	*BioSPI_Enroll_ValidParam*	0b5ebb60-eefb-11d9-990c-0002a5d5c51b
13b	*BioSPI_Enroll_Payload*	e8969d40-ef05-11d9-9098-0002a5d5c51b
13c	*BioSPI_Enroll_AuditData*	b40a5260-ef14-11d9-a4fe-0002a5d5c51b
13d	*BioSPI_Enroll_BIRHeaderQuality*	6f727320-ef1a-11d9-9143-0002a5d5c51b
14a	*BioSPI_Verify_ValidParam*	b78e5be0-efcb-11d9-b2c7-0002a5d5c51b
14b	*BioSPI_Verify_Payload*	32969ec0-eff8-11d9-9831-0002a5d5c51b
14c	*BioSPI_Verify_AuditData*	89719700-f218-11d9-b028-0002a5d5c51b

7.2.3 Если испытуемая реализация поддерживает функцию *BioSPI_ Capture*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 4, в указанном порядке.

Таблица 4 – Тестовые утверждения для испытания ПБУ верификации, поддерживающего функцию *BioSPI_Capture*

Номер	Наименование утверждения	Пакет
9a	*BioSPI_Capture_AuditData*	02704c50-0cd8-1085-96cb-0002a5d5fd2e
9b	*BioSPI_Capture_ReturnQuality*	03f601f0-0cd8-1085-bd59-0002a5d5fd2e
9c	*BioSPI_Capture_IntermediateProcessedBIR*	055ddb08-0cd6-1085-a6d3-0002a5d5fd2e
9d	*BioSPI_Capture_InvalidBSPHandle*	0244f2a8-0cf2-1085-8443-0002a5d5fd2e

7.2.4 Если испытуемая реализация поддерживает функцию *BioSPI_CreateTemplate*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 5, в указанном порядке.

Таблица 5 – Тестовые утверждения для испытания ПБУ верификации, поддерживающего функцию *BioSPI_CreateTemplate*

Номер	Наименование утверждения	Пакет
10a	*BioSPI_CreateTemplate_PayloadSupported*	04a01118-0cf9-1085-96d4-0002a5d5fd2e
10b	*BioSPI_CreateTemplate_BIRHeaderQuality*	00b5c728-0cfb-1085-8969-0002a5d5fd2e
10c	*BioSPI_CreateTemplate_OutputBIRDataType*	0193c730-0cf9-1085-b0a3-0002a5d5fd2e
10d	*BioSPI_CreateTemplate_OutputBIRPurpose*	03dbdaa0-0cf2-1085-99ed-0002a5d5fd2e
10e	*BioSPI_CreateTemplate_InputBIRDataType*	6d543ea0-2ce9-11d9-9669-0800200c9a66
10f	*BioSPI_CreateTemplate_Inconsistent_Purpose*	28ec1620-e995-11d9-b1d1-0002a5d5c51b

7.2.5 Если испытуемая реализация поддерживает функцию *BioSPI_Process*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 6, в указанном порядке.

Таблица 6 – Тестовые утверждения для испытания ПБУ верификации, поддерживающего функцию *BioSPI_Process*

Номер	Наименование утверждения	Пакет
11a	*BioSPI_Process_ValidParam*	4ec34700-e9a0-11d9-8fc8-0002a5d5c51b
11b	*BioSPI_Process_BIRHeaderQuality*	211668e0-e9a6-11d9-bcc8-0002a5d5c51b
11c	*BioSPI_Process_OutputBIRPurpose*	e1bb4f20-ed61-11d9-9344-0002a5d5c51b
11d	*BioSPI_Process_BuildsProcessedBIR*	f2ce6540-ed66-11d9-9618-0002a5d5c51b
11e	*BioSPI_Process_InputBIRDataType*	3cf96080-ed6b-11d9-9acf-0002a5d5c51b

7.2.6 Если испытуемая реализация поддерживает функцию *BioSPI_VerifyMatch*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 7, в указанном порядке.

Таблица 7 – Тестовые утверждения для испытания ПБУ верификации, поддерживающего функцию *BioSPI_VerifyMatch*

Номер	Наименование утверждения	Пакет
12a	*BioSPI_VerifyMatch_ValidParam*	688aad60-ee30-11d9-a62c-0002a5d5c51b
12b	*BioSPI_VerifyMatch_Payload*	692ebe20-ee47-11d9-bd34-0002a5d5c51b
12c	*BioSPI_VerifyMatch_Inconsistent_Purpose*	9108ec70-2e9b-11d9-9669-0800200c9a66

7.2.7 Если испытуемая реализация поддерживает базу данных контролируемой ПБУ (см. ИСО/МЭК 19784-1, раздел А.4), то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 8, в указанном порядке.

Таблица 8 – Тестовые утверждения для испытания ПБУ верификации, поддерживающего функцию контроля базы данных

Номер	Наименование утверждения	Пакет
15a	*BioSPI_DbOpen_ValidParam*	e68ff9a0-e506-11d9-a6a1-0002a5d5c51b
15b	*BioSPI_DbOpen_InvalidBSPHandle*	bfd44400-e5de-11d9-bdb9-0002a5d5c51b
16a	*BioSPI_DbClose_ValidParam*	39aa9560-e5f1-11d9-89f3-0002a5d5c51b
16b	*BioSPI_DbClose_InvalidBSPHandle*	6e3f5c00-e5f3-11d9-b663-0002a5d5c51b
17a	*BioSPI_DbCreate_DbProtected*	7b6c2f40-e650-11d9-812f-0002a5d5c51b
17b	*BioSPI_DbCreate_ValidParam*	1421ec38-1db6-49d4-873d-03e2de17598b
17c	*BioSPI_DbCreate_InvalidBSPHandle*	ef4bb862-79f6-4f01-8f5d-af5c3abf23c0
18a	*BioSPI_DbDelete_InvalidBSPHandle*	678e5d12-3d51-41ec-a672-13f34ea24545

Окончание таблицы 8

Номер	Наименование утверждения	Пакет
18b	*BioSPI_DbDelete_OpenDbProtected*	9e4d0c6d-4d59-479c-9f58-160ecde99aad
18c	*BioSPI_DbDelete_ValidParam*	499d9cc3-4269-4671-9d69-29a31bc1a08f
19a	*BioSPI_DbSetMarker_ValidParam*	94271080-e723-11d9-898c-0002a5d5c51b
19b	*BioSPI_DbSetMarker_InvalidBSPHandle*	69f7ce20-e72e-11d9-b4e0-0002a5d5c51b
19c	*BioSPI_DbSetMarker_RecordNotFound*	b9c90f40-e7ec-11d9-a435-0002a5d5c51b
20a	*BioSPI_DbFreeMarker_ValidParam*	2e1c9520-e7f1-11d9-a011-0002a5d5c51b
20b	*BioSPI_DbFreeMarker_InvalidBSPHandle*	0f735140-e800-11d9-8a8a-0002a5d5c51b
20c	*BioSPI_DbFreeMarker_InvalidMarker*	a9007b60-e802-11d9-8a5d-0002a5d5c51b
21a	*BioSPI_DbStoreBIR_ValidParam*	da953680-e806-11d9-90d3-0002a5d5c51b
21b	*BioSPI_DbStoreBIR_InvalidBSPHandle*	e39027e0-e80b-11d9-85eb-0002a5d5c51b
22a	*BioSPI_DbGetBIR_ValidParam*	67512700-e811-11d9-a5e0-0002a5d5c51b
22b	*BioSPI_DbGetBIR_InvalidBSPHandle*	f62947e0-e8b7-11d9-9dad-0002a5d5c51b
22c	*BioSPI_DbGetBIR_RecordNotFound*	37457440-e8ba-11d9-87da-0002a5d5c51b
23a	*BioSPI_DbGetNextBIR_ValidParam*	e3396400-e8c9-11d9-990f-0002a5d5c51b
23b	*BioSPI_DbGetNextBIR_InvalidBSPHandle*	f0ef5320-e8d3-11d9-bbc1-0002a5d5c51b
24a	*BioSPI_DbDeleteBIR_ValidParam*	ed4afbe0-e8d6-11d9-9ed0-0002a5d5c51b
24b	*BioSPI_DbDeleteBIR_InvalidBSPHandle*	72eca940-e8d9-11d9-aa0d-0002a5d5c51b

7.3 Испытание ПБУ подкласса ПБУ идентификации

7.3.1 В настоящем подразделе приведены тестовые утверждения, которые следует использовать для испытания ПБУ идентификации на соответствие требованиям ИСО/МЭК 19784-1, приложение А, раздел А.4

7.3.2 Все ПБУ идентификации должны быть подвергнуты испытаниям путем выполнения всех тестовых утверждений, приведенных в таблице 9, в указанном порядке.

Таблица 9 – Тестовые утверждения для испытания ПБУ идентификации

Номер	Наименование утверждения	Пакет
1a	*BioSPI_BSPLoad_InvalidUUID*	020e90c8-0c19-1085-ab54-0002a5d5fd2e
1b	*BioSPI_BSPLoad_ValidParam*	01f6c6f0-0c19-1085-97fe-0002a5d5fd2e
2a	*BioSPI_BSPUnload_ValidParam*	01661010-0c22-1085-8688-0002a5d5fd2e
2b	*BioSPI_BSPUnload_InvalidUUID*	01c2e5b0-0c3b-1085-b31d-0002a5d5fd2e
2c	*BioSPI_BSPUnload_UnmatchedLoad*	02f6c618-0c23-1085-ba89-0002a5d5fd2e
2d	*BioSPI_BSPUnload_Confirm*	03daf040-0c3b-1085-a9fd-0002a5d5fd2e
3a	*BioSPI_BSPAttach_ValidParam*	00ae6488-0c3d-1085-9912-0002a5d5fd2e
3b	*BioSPI_BSPAttach_InvalidUUID*	049cc170-0c5f-1085-981f-0002a5d5fd2e
3c	*BioSPI_BSPAttach_InvalidVersion*	0052ac10-0c60-1085-9883-0002a5d5fd2e
3d	*BioSPI_BSPAttach_InvalidBSPHandle*	03826830-0c57-1085-bfb0-0002a5d5fd2e
4a	*BioSPI_BSPDetach_ValidParam*	00e0d2b0-0c7a-1085-b8ac-0002a5d5fd2e
4b	*BioSPI_BSPDetach_InvalidBSPHandle*	0434c458-0c79-1085-9f2c-0002a5d5fd2e
4c	*BioSPI_BSPDetach_Confirm*	002e7e58-0c78-1085-9e1d-0002a5d5fd2e
5a	*BioSPI_FreeBIRHandle_ValidParam*	0280a7d0-0c80-1085-a9a0-0002a5d5fd2e
5b	*BioSPI_FreeBIRHandle_InvalidBSPHandle*	047aed48-0c80-1085-898b-0002a5d5fd2e
5c	*BioSPI_FreeBIRHandle_InvalidBIRHandle*	018e6c18-0c9c-1085-afdf-0002a5d5fd2e
6a	*BioSPI_GetBIRFromHandle_ValidParam*	0460b658-0cb4-1085-a304-0002a5d5fd2e
6b	*BioSPI_GetBIRFromHandle_InvalidBSPHandle*	02445668-0cc5-1085-a3ac-0002a5d5fd2e
6c	*BioSPI_GetBIRFromHandle_InvalidBIRHandle*	0194a9c0-0cc7-1085-8780-0002a5d5fd2e
7a	*BioSPI_GetHeaderFromHandle_ValidParam*	027a7db0-0cc7-1085-9391-0002a5d5fd2e
7b	*BioSPI_GetHeaderFromHandle_InvalidBSPHandle*	057e0d38-0ccd-1085-83b8-0002a5d5fd2e
7c	*BioSPI_GetHeaderFromHandle_InvalidBIRHandle*	02195e68-0cce-1085-a46f-0002a5d5fd2e
7d	*BioSPI_GetHeaderFromHandle_BIRHandleNotFreed*	01cc0988-0ccf-1085-a367-0002a5d5fd2e
8a	*BioSPI_EnableEvents_ValidParam*	0333f628-0ccf-1085-aceb-0002a5d5fd2e

Окончание таблицы 9

Номер	Наименование утверждения	Пакет
8b	*BioSPI_EnableEvents_InvalidBSPHandle*	04ed0838-0ccf-1085-b64e-0002a5d5fd2e
13a	*BioSPI_Enroll_ValidParam*	0b5ebb60-eefb-11d9-990c-0002a5d5c51b
13b	*BioSPI_Enroll_Payload*	e8969d40-ef05-11d9-9098-0002a5d5c51b
13c	*BioSPI_Enroll_AuditData*	b40a5260-ef14-11d9-a4fe-0002a5d5c51b
13d	*BioSPI_Enroll_BIRHeaderQuality*	6f727320-ef1a-11d9-9143-0002a5d5c51b
14a	*BioSPI_Verify_ValidParam*	b78e5be0-efcb-11d9-b2c7-0002a5d5c51b
14b	*BioSPI_Verify_Payload*	32969ec0-eff8-11d9-9831-0002a5d5c51b
14c	*BioSPI_Verify_AuditData*	89719700-f218-11d9-b028-0002a5d5c51b

7.3.3 Если реализация поддерживает функцию *BioSPI_Capture*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 10, в указанном порядке.

Таблица 10 – Утверждения для испытания ПБУ идентификации, поддерживающего функцию *BioSPI_Capture*

Номер	Наименование утверждения	Пакет
9a	*BioSPI_Capture_AuditData*	02704c50-0cd8-1085-96cb-0002a5d5fd2e
9b	*BioSPI_Capture_ReturnQuality*	03f601f0-0cd8-1085-bd59-0002a5d5fd2e
9c	*BioSPI_Capture_IntermediateProcessedBIR*	055ddb08-0cd6-1085-a6d3-0002a5d5fd2e
9d	*BioSPI_Capture_InvalidBSPHandle*	0244f2a8-0cf2-1085-8443-0002a5d5fd2e

7.3.4 Если реализация поддерживает функцию *BioSPI_CreateTemplate*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 11, в указанном порядке.

Таблица 11 – Утверждения для испытания ПБУ идентификации, поддерживающего функцию *BioSPI_CreateTemplate*

Номер	Наименование утверждения	Пакет
10a	*BioSPI_CreateTemplate_PayloadSupported*	04a01118-0cf9-1085-96d4-0002a5d5fd2e
10b	*BioSPI_CreateTemplate_BIRHeaderQuality*	00b5c728-0cfb-1085-8969-0002a5d5fd2e
10c	*BioSPI_CreateTemplate_OutputBIRDataType*	0193c730-0cf9-1085-b0a3-0002a5d5fd2e
10d	*BioSPI_CreateTemplate_OutputBIRPurpose*	03dbdaa0-0cf2-1085-99ed-0002a5d5fd2e
10e	*BioSPI_CreateTemplate_InputBIRDataType*	6d543ea0-2ce9-11d9-9669-0800200c9a66
10f	*BioSPI_CreateTemplate_Inconsistent_Purpose*	28ec1620-e995-11d9-b1d1-0002a5d5c51b

7.3.5 Если реализация поддерживает функцию *BioSPI_Process*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 12, в указанном порядке.

Таблица 12 – Утверждения для испытания ПБУ идентификации, поддерживающего функцию *BioSPI_Process*

Номер	Наименование утверждения	Пакет
11a	*BioSPI_Process_ValidParam*	4ec34700-e9a0-11d9-8fc8-0002a5d5c51b
11b	*BioSPI_Process_BIRHeaderQuality*	211668e0-e9a6-11d9-bcc8-0002a5d5c51b
11c	*BioSPI_Process_OutputBIRPurpose*	e1bb4f20-ed61-11d9-9344-0002a5d5c51b
11d	*BioSPI_Process_OutputBIRPurpose*	f2ce6540-ed66-11d9-9618-0002a5d5c51b
11e	*BioSPI_Process_InputBIRDataType*	3cf96080-ed6b-11d9-9acf-0002a5d5c51b

7.3.6 Если реализация поддерживает функцию *BioSPI_VerifyMatch*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 13, в указанном порядке.

Таблица 13 – Утверждения для испытания ПБУ идентификации, поддерживающего функцию *BioSPI_VerifyMatch*

Номер	Наименование утверждения	Пакет
12a	*BioSPI_VerifyMatch_ValidParam*	688aad60-ee30-11d9-a62c-0002a5d5c51b
12b	*BioSPI_VerifyMatch_Payload*	692ebe20-ee47-11d9-bd34-0002a5d5c51b
12c	*BioSPI_VerifyMatch_Inconsistent_Purpose*	9108ec70-2e9b-11d9-9669-0800200c9a66

7.3.7 Если реализация поддерживает базу данных контролируемой ПБУ (см. 19784-1, приложение А, пункт А.4.6.1.2), то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 14, в указанном порядке.

Таблица 14 – Утверждения для испытания ПБУ идентфкации, поддерживающей функции работы с базой данных

Номер	Наименование утверждения	Пакет
15a	*BioSPI_DbOpen_ValidParam*	e68ff9a0-e506-11d9-a6a1-0002a5d5c51b
15b	*BioSPI_DbOpen_InvalidBSPHandle*	bfd44400-e5de-11d9-bdb9-0002a5d5c51b
16a	*BioSPI_DbOpen_InvalidBSPHandle*	39aa9560-e5f1-11d9-89f3-0002a5d5c51b
16b	*BioSPI_DbClose_InvalidBSPHandle*	6e3f5c00-e5f3-11d9-b663-0002a5d5c51b
17a	*BioSPI_DbCreate_DbProtected*	7b6c2f40-e650-11d9-812f-0002a5d5c51b
17b	*BioSPI_DbCreate_ValidParam*	1421ec38-1db6-49d4-873d-03e2de17598b
17c	*BioSPI_DbCreate_InvalidBSPHandle*	ef4bb862-79f6-4f01-8f5d-af5c3abf23c0
18a	*BioSPI_DbDelete_InvalidBSPHandle*	678e5d12-3d51-41ec-a672-13f34ea24545
18b	*BioSPI_DbDelete_OpenDbProtected*	9e4d0c6d-4d59-479c-9f58-160ecde99aad
18c	*BioSPI_DbDelete_ValidParam*	499d9cc3-4269-4671-9d69-29a31bc1a08f
19a	*BioSPI_DbSetMarker_ValidParam*	94271080-e723-11d9-898c-0002a5d5c51b
19b	*BioSPI_DbSetMarker_InvalidBSPHandle*	69f7ce20-e72e-11d9-b4e0-0002a5d5c51b
19c	*BioSPI_DbSetMarker_RecordNotFound*	b9c90f40-e7ec-11d9-a435-0002a5d5c51b

Окончание таблицы 14

Номер	Наименование утверждения	Пакет
20a	*BioSPI_DbFreeMarker_ValidParam*	2e1c9520-e7f1-11d9-a011-0002a5d5c51b
20b	*BioSPI_DbFreeMarker_InvalidBSPHandle*	0f735140-e800-11d9-8a8a-0002a5d5c51b
20c	*BioSPI_DbFreeMarker_InvalidMarker*	a9007b60-e802-11d9-8a5d-0002a5d5c51b
21a	*BioSPI_DbStoreBIR_ValidParam*	da953680-e806-11d9-90d3-0002a5d5c51b
21b	*BioSPI_DbStoreBIR_InvalidBSPHandle*	e39027e0-e80b-11d9-85eb-0002a5d5c51b
22a	*BioSPI_DbGetBIR_ValidParam*	67512700-e811-11d9-a5e0-0002a5d5c51b
22b	*BioSPI_DbGetBIR_InvalidBSPHandle*	f62947e0-e8b7-11d9-9dad-0002a5d5c51b
22c	*BioSPI_DbGetBIR_RecordNotFound*	37457440-e8ba-11d9-87da-0002a5d5c51b
23a	*BioSPI_DbGetNextBIR_ValidParam*	e3396400-e8c9-11d9-990f-0002a5d5c51b
24b	*BioSPI_DbGetNextBIR_InvalidBSPHandle*	f0ef5320-e8d3-11d9-bbc1-0002a5d5c51b
23a	*BioSPI_DbDeleteBIR_ValidParam*	ed4afbe0-e8d6-11d9-9ed0-0002a5d5c51b
24b	*BioSPI_DbDeleteBIR_InvalidBSPHandle*	72eca940-e8d9-11d9-aa0d-0002a5d5c51b

7.4 Испытание ПБУ подкласса ПБУ получения данных

7.4.1 В настоящем подразделе приведены тестовые утверждения, которые следует использовать для испытания ПБУ получения данных на соответствие требованиям ИСО/МЭК 19784-1, приложение А, раздел А.4.

7.4.2 Все ПБУ данного подкласса должны быть подвергнуты испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 15, в указанном порядке.

Таблица 15 – Утверждения для испытания ПБУ получения данных

Номер	Наименование утверждения	Пакет
1a	*BioSPI_BSPLoad_InvalidUUID*	020e90c8-0c19-1085-ab54-0002a5d5fd2e
1b	*BioSPI_BSPLoad_ValidParam*	01f6c6f0-0c19-1085-97fe-0002a5d5fd2e

Продолжение таблицы 15

Номер	Наименование утверждения	Пакет
2a	*BioSPI_BSPUnload_ValidParam*	01661010-0c22-1085-8688-0002a5d5fd2e
2b	*BioSPI_BSPUnload_InvalidUUID*	01c2e5b0-0c3b-1085-b31d-0002a5d5fd2e
2c	*BioSPI_BSPUnload_UnmatchedLoad*	02f6c618-0c23-1085-ba89-0002a5d5fd2e
2d	*BioSPI_BSPUnload_Confirm*	03daf040-0c3b-1085-a9fd-0002a5d5fd2e
3a	*BioSPI_BSPAttach_ValidParam*	00ae6488-0c3d-1085-9912-0002a5d5fd2e
3b	*BioSPI_BSPAttach_InvalidUUID*	049cc170-0c5f-1085-981f-0002a5d5fd2e
3c	*BioSPI_BSPAttach_InvalidVersion*	0052ac10-0c60-1085-9883-0002a5d5fd2e
3d	*BioSPI_BSPAttach_InvalidBSPHandle*	03826830-0c57-1085-bfb0-0002a5d5fd2e
4a	*BioSPI_BSPDetach_ValidParam*	00e0d2b0-0c7a-1085-b8ac-0002a5d5fd2e
4b	*BioSPI_BSPDetach_InvalidBSPHandle*	0434c458-0c79-1085-9f2c-0002a5d5fd2e
4c	*BioSPI_BSPDetach_Confirm*	002e7e58-0c78-1085-9e1d-0002a5d5fd2e
5a	*BioSPI_FreeBIRHandle_ValidParam*	0280a7d0-0c80-1085-a9a0-0002a5d5fd2e
5b	*BioSPI_FreeBIRHandle_InvalidBSPHandle*	047aed48-0c80-1085-898b-0002a5d5fd2e
5c	*BioSPI_FreeBIRHandle_InvalidBIRHandle*	018e6c18-0c9c-1085-afdf-0002a5d5fd2e
6a	*BioSPI_GetBIRFromHandle_ValidParam*	0460b658-0cb4-1085-a304-0002a5d5fd2e
6b	*BioSPI_GetBIRFromHandle_InvalidBSPHandle*	02445668-0cc5-1085-a3ac-0002a5d5fd2e
6c	*BioSPI_GetBIRFromHandle_InvalidBIRHandle*	0194a9c0-0cc7-1085-8780-0002a5d5fd2e
7a	*BioSPI_GetHeaderFromHandle_ValidParam*	027a7db0-0cc7-1085-9391-0002a5d5fd2e
7b	*BioSPI_GetHeaderFromHandle_InvalidBSPHandle*	057e0d38-0ccd-1085-83b8-0002a5d5fd2e
7c	*BioSPI_GetHeaderFromHandle_InvalidBIRHandle*	02195e68-0cce-1085-a46f-0002a5d5fd2e
7d	*BioSPI_GetHeaderFromHandle_BIRHandleNotFreed*	01cc0988-0ccf-1085-a367-0002a5d5fd2e
8a	*BioSPI_EnableEvents_ValidParam*	0333f628-0ccf-1085-aceb-0002a5d5fd2e
8b	*BioSPI_EnableEvents_InvalidBSPHandle*	04ed0838-0ccf-1085-b64e-0002a5d5fd2e
9a	*BioSPI_Capture_AuditData*	02704c50-0cd8-1085-96cb-0002a5d5fd2e

Окончание таблицы 15

Номер	Наименование утверждения	Пакет
9b	*BioSPI_Capture_ReturnQuality*	03f601f0-0cd8-1085-bd59-0002a5d5fd2e
9c	*BioSPI_Capture_IntermediateProcessedBIR*	055ddb08-0cd6-1085-a6d3-0002a5d5fd2e
9d	*BioSPI_Capture_InvalidBSPHandle*	0244f2a8-0cf2-1085-8443-0002a5d5fd2e

7.4.3 Если реализация поддерживает функцию *BioSPI_CreateTemplate*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 16, в указанном порядке.

Таблица 16 – Утверждения для испытания ПБУ получения данных, поддерживающего функцию *BioSPI_CreateTemplate*

Номер	Наименование утверждения	Пакет
10a	*BioSPI_CreateTemplate_PayloadSupported*	04a01118-0cf9-1085-96d4-0002a5d5fd2e
10b	*BioSPI_CreateTemplate_BIRHeaderQuality*	00b5c728-0cfb-1085-8969-0002a5d5fd2e
10c	*BioSPI_CreateTemplate_OutputBIRDataType*	0193c730-0cf9-1085-b0a3-0002a5d5fd2e
10d	*BioSPI_CreateTemplate_OutputBIRPurpose*	03dbdaa0-0cf2-1085-99ed-0002a5d5fd2e
10e	*BioSPI_CreateTemplate_InputBIRDataType*	6d543ea0-2ce9-11d9-9669-0800200c9a66
10f	*BioSPI_CreateTemplate_Inconsistent_Purpose*	28ec1620-e995-11d9-b1d1-0002a5d5c51b

7.4.4 Если реализация поддерживает функцию *BioSPI_Process*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 17, в указанном порядке.

Таблица 17 – Утверждения для испытания ПБУ получения данных, поддерживающего функцию *BioSPI_Process*

Номер	Наименование утверждения	Пакет
11a	*BioSPI_Process_ValidParam*	4ec34700-e9a0-11d9-8fc8-0002a5d5c51b
11b	*BioSPI_Process_BIRHeaderQuality*	211668e0-e9a6-11d9-bcc8-0002a5d5c51b

Окончание таблицы 17

Номер	Наименование утверждения	Пакет
11c	*BioSPI_Process_OutputBIRPurpose*	e1bb4f20-ed61-11d9-9344-0002a5d5c51b
11d	*BioSPI_Process_BuildsProcessedBIR*	f2ce6540-ed66-11d9-9618-0002a5d5c51b
11e	*BioSPI_Process_InputBIRDataType*	3cf96080-ed6b-11d9-9acf-0002a5d5c51b

7.4.5 Если реализация поддерживает функцию *BioSPI_VerifyMatch*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 18, в указанном порядке.

Таблица 18 – Утверждения для испытания ПБУ получения данных, поддерживающего функцию *BioSPI_VerifyMatch*

Номер	Наименование утверждения	Пакет
12a	*BioSPI_VerifyMatch_ValidParam*	688aad60-ee30-11d9-a62c-0002a5d5c51b
12b	*BioSPI_VerifyMatch_Payload*	692ebe20-ee47-11d9-bd34-0002a5d5c51b
12c	*BioSPI_VerifyMatch_Inconsistent_Purpose*	9108ec70-2e9b-11d9-9669-0800200c9a66

7.4.6 Если реализация поддерживает функцию *BioSPI_Enroll*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 19, в указанном порядке.

Таблица 19 – Утверждения для испытания ПБУ получения данных, поддерживающего функцию *BioSPI_Enroll*

Номер	Наименование утверждения	Пакет
13a	*BioSPI_Enroll_ValidParam*	0b5ebb60-eefb-11d9-990c-0002a5d5c51b
13b	*BioSPI_Enroll_Payload*	e8969d40-ef05-11d9-9098-0002a5d5c51b
13c	*BioSPI_Enroll_AuditData*	b40a5260-ef14-11d9-a4fe-0002a5d5c51b
13d	*BioSPI_Enroll_BIRHeaderQuality*	6f727320-ef1a-11d9-9143-0002a5d5c51b

7.4.7 Если реализация поддерживает функцию *BioSPI_Verify*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 20, в указанном порядке.

Таблица 20 – Утверждения для испытания ПБУ получения данных, поддерживающего функцию *BioSPI_Verify*

Номер	Наименование утверждения	Пакет
14a	*BioSPI_Verify_ValidParam*	b78e5be0-efcb-11d9-b2c7-0002a5d5c51b
14b	*BioSPI_Verify_Payload*	32969ec0-eff8-11d9-9831-0002a5d5c51b
14c	*BioSPI_Verify_AuditData*	89719700-f218-11d9-b028-0002a5d5c51b

7.4.8 Если реализация поддерживает базу данных контролируемой ПБУ (см. 19784-1, приложение А, пункт А.4.6.1.2), то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 21, в указанном порядке.

Таблица 21 – Утверждения для ПБУ получения данных, поддерживающего функции работы с базой данных

Номер	Наименование утверждения	Пакет
15a	*BioSPI_DbOpen_ValidParam*	e68ff9a0-e506-11d9-a6a1-0002a5d5c51b
15b	*BioSPI_DbOpen_InvalidBSPHandle*	bfd44400-e5de-11d9-bdb9-0002a5d5c51b
16a	*BioSPI_DbClose_ValidParam*	39aa9560-e5f1-11d9-89f3-0002a5d5c51b
16b	*BioSPI_DbClose_InvalidBSPHandle*	6e3f5c00-e5f3-11d9-b663-0002a5d5c51b
17a	*BioSPI_DbCreate_DbProtected*	7b6c2f40-e650-11d9-812f-0002a5d5c51b
17b	*BioSPI_DbCreate_ValidParam*	1421ec38-1db6-49d4-873d-03e2de17598b
17c	*BioSPI_DbCreate_InvalidBSPHandle*	ef4bb862-79f6-4f01-8f5d-af5c3abf23c0
18a	*BioSPI_DbDelete_InvalidBSPHandle*	678e5d12-3d51-41ec-a672-13f34ea24545
18b	*BioSPI_DbDelete_OpenDbProtected*	9e4d0c6d-4d59-479c-9f58-160ecde99aad

Окончание таблицы 21

Номер	Наименование утверждения	Пакет
18c	*BioSPI_DbDelete_ValidParam*	499d9cc3-4269-4671-9d69-29a31bc1a08f
19a	*BioSPI_DbSetMarker_ValidParam*	94271080-e723-11d9-898c-0002a5d5c51b
19b	*BioSPI_DbSetMarker_InvalidBSPHandle*	69f7ce20-e72e-11d9-b4e0-0002a5d5c51b
19c	*BioSPI_DbSetMarker_RecordNotFound*	b9c90f40-e7ec-11d9-a435-0002a5d5c51b
20a	*BioSPI_DbFreeMarker_ValidParam*	2e1c9520-e7f1-11d9-a011-0002a5d5c51b
20b	*BioSPI_DbFreeMarker_InvalidBSPHandle*	0f735140-e800-11d9-8a8a-0002a5d5c51b
20c	*BioSPI_DbFreeMarker_InvalidMarker*	a9007b60-e802-11d9-8a5d-0002a5d5c51b
21a	*BioSPI_DbStoreBIR_ValidParam*	da953680-e806-11d9-90d3-0002a5d5c51b
21b	*BioSPI_DbStoreBIR_InvalidBSPHandle*	e39027e0-e80b-11d9-85eb-0002a5d5c51b
22a	*BioSPI_DbGetBIR_ValidParam*	67512700-e811-11d9-a5e0-0002a5d5c51b
22b	*BioSPI_DbGetBIR_InvalidBSPHandle*	f62947e0-e8b7-11d9-9dad-0002a5d5c51b
22c	*BioSPI_DbGetBIR_RecordNotFound*	37457440-e8ba-11d9-87da-0002a5d5c51b
23a	*BioSPI_DbGetNextBIR_ValidParam*	e3396400-e8c9-11d9-990f-0002a5d5c51b
23b	*BioSPI_DbGetNextBIR_InvalidBSPHandle*	f0ef5320-e8d3-11d9-bbc1-0002a5d5c51b
24a	*BioSPI_DbDeleteBIR_ValidParam*	ed4afbe0-e8d6-11d9-9ed0-0002a5d5c51b
24b	*BioSPI_DbDeleteBIR_InvalidBSPHandle*	72eca940-e8d9-11d9-aa0d-0002a5d5c51b

7.5 Испытание ПБУ подкласса механизм верификации

7.5.1 В настоящем подразделе приведены тестовые утверждения, которые следует использовать для испытания ПБУ подкласса механизм верификации на соответствие требованиям ИСО/МЭК 19784-1, приложение А, раздел А.4.

7.5.2 Все ПБУ данного подкласса должны быть подвергнуты испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 22, в указанном порядке.

Таблица 22 – Утверждения для испытания ПБУ подкласса механизм верификации

Номер	Наименование утверждения	Пакет
1a	*BioSPI_BSPLoad_InvalidUUID*	020e90c8-0c19-1085-ab54-0002a5d5fd2e
1b	*BioSPI_BSPLoad_ValidParam*	01f6c6f0-0c19-1085-97fe-0002a5d5fd2e
2a	*BioSPI_BSPUnload_ValidParam*	01661010-0c22-1085-8688-0002a5d5fd2e
2b	*BioSPI_BSPUnload_InvalidUUID*	01c2e5b0-0c3b-1085-b31d-0002a5d5fd2e
2c	*BioSPI_BSPUnload_UnmatchedLoad*	02f6c618-0c23-1085-ba89-0002a5d5fd2e
2d	*BioSPI_BSPUnload_Confirm*	03daf040-0c3b-1085-a9fd-0002a5d5fd2e
3a	*BioSPI_BSPAttach_ValidParam*	00ae6488-0c3d-1085-9912-0002a5d5fd2e
3b	*BioSPI_BSPAttach_InvalidUUID*	049cc170-0c5f-1085-981f-0002a5d5fd2e
3c	*BioSPI_BSPAttach_InvalidVersion*	0052ac10-0c60-1085-9883-0002a5d5fd2e
3d	*BioSPI_BSPAttach_InvalidBSPHandle*	03826830-0c57-1085-bfb0-0002a5d5fd2e
4a	*BioSPI_BSPDetach_ValidParam*	00e0d2b0-0c7a-1085-b8ac-0002a5d5fd2e
4b	*BioSPI_BSPDetach_InvalidBSPHandle*	0434c458-0c79-1085-9f2c-0002a5d5fd2e
4c	*BioSPI_BSPDetach_Confirm*	002e7e58-0c78-1085-9e1d-0002a5d5fd2e
5a	*BioSPI_FreeBIRHandle_ValidParam*	0280a7d0-0c80-1085-a9a0-0002a5d5fd2e
5b	*BioSPI_FreeBIRHandle_InvalidBSPHandle*	047aed48-0c80-1085-898b-0002a5d5fd2e
5c	*BioSPI_FreeBIRHandle_InvalidBIRHandle*	018e6c18-0c9c-1085-afdf-0002a5d5fd2e
6a	*BioSPI_GetBIRFromHandle_ValidParam*	0460b658-0cb4-1085-a304-0002a5d5fd2e
6b	*BioSPI_GetBIRFromHandle_InvalidBSPHandle*	02445668-0cc5-1085-a3ac-0002a5d5fd2e
6c	*BioSPI_GetBIRFromHandle_InvalidBIRHandle*	0194a9c0-0cc7-1085-8780-0002a5d5fd2e
7a	*BioSPI_GetHeaderFromHandle_ValidParam*	027a7db0-0cc7-1085-9391-0002a5d5fd2e
7b	*BioSPI_GetHeaderFromHandle_InvalidBSPHandle*	057e0d38-0ccd-1085-83b8-0002a5d5fd2e

Окончание таблицы 22

Номер	Наименование утверждения	Пакет
7c	*BioSPI_GetHeaderFromHandle_InvalidBIRHandle*	02195e68-0cce-1085-a46f-0002a5d5fd2e
7d	*BioSPI_GetHeaderFromHandle_BIRHandleNotFreed*	01cc0988-0ccf-1085-a367-0002a5d5fd2e
8a	*BioSPI_EnableEvents_ValidParam*	0333f628-0ccf-1085-aceb-0002a5d5fd2e
8b	*BioSPI_EnableEvents_InvalidBSPHandle*	04ed0838-0ccf-1085-b64e-0002a5d5fd2e
10a	*BioSPI_CreateTemplate_PayloadSupported*	04a01118-0cf9-1085-96d4-0002a5d5fd2e
10b	*BioSPI_CreateTemplate_BIRHeaderQuality*	00b5c728-0cfb-1085-8969-0002a5d5fd2e
10c	*BioSPI_CreateTemplate_OutputBIRDataType*	0193c730-0cf9-1085-b0a3-0002a5d5fd2e
10d	*BioSPI_CreateTemplate_OutputBIRPurpose*	03dbdaa0-0cf2-1085-99ed-0002a5d5fd2e
10e	*BioSPI_CreateTemplate_InputBIRDataType*	6d543ea0-2ce9-11d9-9669-0800200c9a66
10f	*BioSPI_CreateTemplate_Inconsistent_Purpose*	28ec1620-e995-11d9-b1d1-0002a5d5c51b
11a	*BioSPI_Process_ValidParam*	4ec34700-e9a0-11d9-8fc8-0002a5d5c51b
11b	*BioSPI_Process_BIRHeaderQuality*	211668e0-e9a6-11d9-bcc8-0002a5d5c51b
11c	*BioSPI_Process_OutputBIRPurpose*	e1bb4f20-ed61-11d9-9344-0002a5d5c51b
11d	*BioSPI_Process_BuildsProcessedBIR*	f2ce6540-ed66-11d9-9618-0002a5d5c51b
11e	*BioSPI_Process_InputBIRDataType*	3cf96080-ed6b-11d9-9acf-0002a5d5c51b
12a	*BioSPI_VerifyMatch_ValidParam*	688aad60-ee30-11d9-a62c-0002a5d5c51b
12b	*BioSPI_VerifyMatch_Payload*	692ebe20-ee47-11d9-bd34-0002a5d5c51b
12c	*BioSPI_VerifyMatch_Inconsistent_Purpose*	9108ec70-2e9b-11d9-9669-0800200c9a66

7.5.3 Если реализация поддерживает функцию *BioSPI_Capture*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 23, в указанном порядке.

Таблица 23 – Утверждения для испытания ПБУ подкласса механизм верификации, поддерживающего функцию *BioSPI_Capture*

Номер	Наименование утверждения	Пакет
9a	*BioSPI_Capture_AuditData*	02704c50-0cd8-1085-96cb-0002a5d5fd2e
9b	*BioSPI_Capture_ReturnQuality*	03f601f0-0cd8-1085-bd59-0002a5d5fd2e
9c	*BioSPI_Capture_IntermediateProcessedBIR*	055ddb08-0cd6-1085-a6d3-0002a5d5fd2e
9d	*BioSPI_Capture_InvalidBSPHandle*	0244f2a8-0cf2-1085-8443-0002a5d5fd2e

7.5.4 Если реализация поддерживает функцию *BioSPI_Enroll*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 24, в указанном порядке.

Таблица 24 – Утверждения для испытания ПБУ подкласса механизм верификации, поддерживающего функцию BioSPI_Enroll

Номер	Наименование утверждения	Пакет
13a	*BioSPI_Enroll_ValidParam*	0b5ebb60-eefb-11d9-990c-0002a5d5c51b
13b	*BioSPI_Enroll_Payload*	e8969d40-ef05-11d9-9098-0002a5d5c51b
13c	*BioSPI_Enroll_AuditData*	b40a5260-ef14-11d9-a4fe-0002a5d5c51b
13d	*BioSPI_Enroll_BIRHeaderQuality*	6f727320-ef1a-11d9-9143-0002a5d5c51b

7.5.5 Если реализация поддерживает функцию *BioSPI_Verify*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 25, в указанном порядке.

Таблица 25 – Утверждения для испытания ПБУ подкласса механизм верификации, поддерживающего функцию *BioSPI_Verify*

Номер	Наименование утверждения	Пакет
14a	*BioSPI_Verify_ValidParam*	b78e5be0-efcb-11d9-b2c7-0002a5d5c51b
14b	*BioSPI_Verify_Payload*	32969ec0-eff8-11d9-9831-0002a5d5c51b
14c	*BioSPI_Verify_AuditData*	89719700-f218-11d9-b028-0002a5d5c51b

7.5.6 Если реализация поддерживает базу данных контролируемой ПБУ (см. 19784-1, приложение А, пункт А.4.6.1.2), то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 26, в указанном порядке.

Таблица 26 – Утверждения для испытания ПБУ подкласса механизм верификации, поддерживающего функции работы с базой данных

Номер	Наименование утверждения	Пакет
15a	*BioSPI_DbOpen_ValidParam*	e68ff9a0-e506-11d9-a6a1-0002a5d5c51b
15b	*BioSPI_DbOpen_InvalidBSPHandle*	bfd44400-e5de-11d9-bdb9-0002a5d5c51b
16a	*BioSPI_DbClose_ValidParam*	39aa9560-e5f1-11d9-89f3-0002a5d5c51b
16b	*BioSPI_DbClose_InvalidBSPHandle*	6e3f5c00-e5f3-11d9-b663-0002a5d5c51b
17a	*BioSPI_DbCreate_DbProtected*	7b6c2f40-e650-11d9-812f-0002a5d5c51b
17b	*BioSPI_DbCreate_ValidParam*	1421ec38-1db6-49d4-873d-03e2de17598b
17c	*BioSPI_DbCreate_InvalidBSPHandle*	ef4bb862-79f6-4f01-8f5d-af5c3abf23c0
18a	*BioSPI_DbDelete_InvalidBSPHandle*	678e5d12-3d51-41ec-a672-13f34ea24545
18b	*BioSPI_DbDelete_OpenDbProtected*	9e4d0c6d-4d59-479c-9f58-160ecde99aad
18c	*BioSPI_DbDelete_ValidParam*	499d9cc3-4269-4671-9d69-29a31bc1a08f

Окончание таблицы 26

Номер	Наименование утверждения	Пакет
19a	*BioSPI_DbSetMarker_ValidParam*	94271080-e723-11d9-898c-0002a5d5c51b
19b	*BioSPI_DbSetMarker_InvalidBSPHandle*	69f7ce20-e72e-11d9-b4e0-0002a5d5c51b
19c	*BioSPI_DbSetMarker_RecordNotFound*	b9c90f40-e7ec-11d9-a435-0002a5d5c51b
20a	*BioSPI_DbFreeMarker_ValidParam*	2e1c9520-e7f1-11d9-a011-0002a5d5c51b
20b	*BioSPI_DbFreeMarker_InvalidBSPHandle*	0f735140-e800-11d9-8a8a-0002a5d5c51b
20c	*BioSPI_DbFreeMarker_InvalidMarker*	a9007b60-e802-11d9-8a5d-0002a5d5c51b
21a	*BioSPI_DbStoreBIR_ValidParam*	da953680-e806-11d9-90d3-0002a5d5c51b
21b	*BioSPI_DbStoreBIR_InvalidBSPHandle*	e39027e0-e80b-11d9-85eb-0002a5d5c51b
22a	*BioSPI_DbGetBIR_ValidParam*	67512700-e811-11d9-a5e0-0002a5d5c51b
22b	*BioSPI_DbGetBIR_InvalidBSPHandle*	f62947e0-e8b7-11d9-9dad-0002a5d5c51b
22c	*BioSPI_DbGetBIR_RecordNotFound*	37457440-e8ba-11d9-87da-0002a5d5c51b
23a	*BioSPI_DbGetNextBIR_ValidParam*	e3396400-e8c9-11d9-990f-0002a5d5c51b
23b	*BioSPI_DbGetNextBIR_InvalidBSPHandle*	f0ef5320-e8d3-11d9-bbc1-0002a5d5c51b
24a	*BioSPI_DbDeleteBIR_ValidParam*	ed4afbe0-e8d6-11d9-9ed0-0002a5d5c51b
24b	*BioSPI_DbDeleteBIR_InvalidBSPHandle*	72eca940-e8d9-11d9-aa0d-0002a5d5c51b

7.6 Испытание ПБУ подкласса механизм идентификации

7.6.1 В настоящем подразделе приведены тестовые утверждения, которые следует использовать для испытания ПБУ подкласса механизм идентификации на соответствие требованиям ИСО/МЭК 19784-1, приложение А, раздел А.4.

7.6.2 Все ПБУ данного подкласса должны быть подвергнуты испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 27, в указанном порядке.

Таблица 27 – Утверждения для испытания ПБУ подкласса механизм идентификации

Номер	Наименование утверждения	Пакет
1a	*BioSPI_BSPLoad_InvalidUUID*	020e90c8-0c19-1085-ab54-0002a5d5fd2e
1b	*BioSPI_BSPLoad_ValidParam*	01f6c6f0-0c19-1085-97fe-0002a5d5fd2e
2a	*BioSPI_BSPUnload_ValidParam*	01661010-0c22-1085-8688-0002a5d5fd2e
2b	*BioSPI_BSPUnload_InvalidUUID*	01c2e5b0-0c3b-1085-b31d-0002a5d5fd2e
2c	*BioSPI_BSPUnload_UnmatchedLoad*	02f6c618-0c23-1085-ba89-0002a5d5fd2e
2d	*BioSPI_BSPUnload_Confirm*	03daf040-0c3b-1085-a9fd-0002a5d5fd2e
3a	*BioSPI_BSPAttach_ValidParam*	00ae6488-0c3d-1085-9912-0002a5d5fd2e
3b	*BioSPI_BSPAttach_InvalidUUID*	049cc170-0c5f-1085-981f-0002a5d5fd2e
3c	*BioSPI_BSPAttach_InvalidVersion*	0052ac10-0c60-1085-9883-0002a5d5fd2e
3d	*BioSPI_BSPAttach_InvalidBSPHandle*	03826830-0c57-1085-bfb0-0002a5d5fd2e
4a	*BioSPI_BSPDetach_ValidParam*	00e0d2b0-0c7a-1085-b8ac-0002a5d5fd2e
4b	*BioSPI_BSPDetach_InvalidBSPHandle*	0434c458-0c79-1085-9f2c-0002a5d5fd2e
4c	*BioSPI_BSPDetach_Confirm*	002e7e58-0c78-1085-9e1d-0002a5d5fd2e
5a	*BioSPI_FreeBIRHandle_ValidParam*	0280a7d0-0c80-1085-a9a0-0002a5d5fd2e
5b	*BioSPI_FreeBIRHandle_InvalidBSPHandle*	047aed48-0c80-1085-898b-0002a5d5fd2e
5c	*BioSPI_FreeBIRHandle_InvalidBIRHandle*	018e6c18-0c9c-1085-afdf-0002a5d5fd2e
6a	*BioSPI_GetBIRFromHandle_ValidParam*	0460b658-0cb4-1085-a304-0002a5d5fd2e
6b	*BioSPI_GetBIRFromHandle_InvalidBSPHandle*	02445668-0cc5-1085-a3ac-0002a5d5fd2e
6c	*BioSPI_GetBIRFromHandle_InvalidBIRHandle*	0194a9c0-0cc7-1085-8780-0002a5d5fd2e
7a	*BioSPI_GetHeaderFromHandle_ValidParam*	027a7db0-0cc7-1085-9391-0002a5d5fd2e

Окончание таблицы 27

Номер	Наименование утверждения	Пакет
7b	*BioSPI_GetHeaderFromHandle_InvalidBSPHandle*	057e0d38-0ccd-1085-83b8-0002a5d5fd2e
7c	*BioSPI_GetHeaderFromHandle_InvalidBIRHandle*	02195e68-0cce-1085-a46f-0002a5d5fd2e
7d	*BioSPI_GetHeaderFromHandle_BIRHandleNotFreed*	01cc0988-0ccf-1085-a367-0002a5d5fd2e
8a	*BioSPI_EnableEvents_ValidParam*	0333f628-0ccf-1085-aceb-0002a5d5fd2e
8b	*BioSPI_EnableEvents_InvalidBSPHandle*	04ed0838-0ccf-1085-b64e-0002a5d5fd2e
10a	*BioSPI_CreateTemplate_PayloadSupported*	04a01118-0cf9-1085-96d4-0002a5d5fd2e
10b	*BioSPI_CreateTemplate_BIRHeaderQuality*	00b5c728-0cfb-1085-8969-0002a5d5fd2e
10c	*BioSPI_CreateTemplate_OutputBIRDataType*	0193c730-0cf9-1085-b0a3-0002a5d5fd2e
10d	*BioSPI_CreateTemplate_OutputBIRPurpose*	03dbdaa0-0cf2-1085-99ed-0002a5d5fd2e
10e	*BioSPI_CreateTemplate_InputBIRDataType*	6d543ea0-2ce9-11d9-9669-0800200c9a66
10f	*BioSPI_CreateTemplate_Inconsistent_Purpose*	28ec1620-e995-11d9-b1d1-0002a5d5c51b
11a	*BioSPI_Process_ValidParam*	4ec34700-e9a0-11d9-8fc8-0002a5d5c51b
11b	*BioSPI_Process_BIRHeaderQuality*	211668e0-e9a6-11d9-bcc8-0002a5d5c51b
11c	*BioSPI_Process_OutputBIRPurpose*	e1bb4f20-ed61-11d9-9344-0002a5d5c51b
11d	*BioSPI_Process_BuildsProcessedBIR*	f2ce6540-ed66-11d9-9618-0002a5d5c51b
11e	*BioSPI_Process_InputBIRDataType*	3cf96080-ed6b-11d9-9acf-0002a5d5c51b
12a	*BioSPI_VerifyMatch_ValidParam*	688aad60-ee30-11d9-a62c-0002a5d5c51b
12b	*BioSPI_VerifyMatch_Payload*	692ebe20-ee47-11d9-bd34-0002a5d5c51b
12c	*BioSPI_VerifyMatch_Inconsistent_Purpose*	9108ec70-2e9b-11d9-9669-0800200c9a66

7.6.3 Если реализация поддерживает функцию *BioSPI_Capture*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 28, в указанном порядке.

Таблица 28 – Утверждения для испытания ПБУ подкласса механизм идентификации, поддерживающего функцию *BioSPI_Capture*

Номер	Наименование утверждения	Пакет
9a	*BioSPI_Capture_AuditData*	02704c50-0cd8-1085-96cb-0002a5d5fd2e
9b	*BioSPI_Capture_ReturnQuality*	03f601f0-0cd8-1085-bd59-0002a5d5fd2e
9c	*BioSPI_Capture_IntermediateProcessedBIR*	055ddb08-0cd6-1085-a6d3-0002a5d5fd2e
9d	*BioSPI_Capture_InvalidBSPHandle*	0244f2a8-0cf2-1085-8443-0002a5d5fd2e

7.6.4 Если реализация поддерживает функцию *BioSPI_Enroll*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенныхв таблице 29, в указанном порядке.

Таблица 29 – Утверждения для ПБУ подкласса механизм идентификации, поддерживающего функцию *BioSPI_Enroll*

Номер	Наименование утверждения	Пакет
13a	*BioSPI_Enroll_ValidParam*	0b5ebb60-eefb-11d9-990c-0002a5d5c51b
13b	*BioSPI_Enroll_Payload*	e8969d40-ef05-11d9-9098-0002a5d5c51b
13c	*BioSPI_Enroll_AuditData*	b40a5260-ef14-11d9-a4fe-0002a5d5c51b
13d	*BioSPI_Enroll_BIRHeaderQuality*	6f727320-ef1a-11d9-9143-0002a5d5c51b

7.6.5 Если реализация поддерживает функцию *BioSPI_Verify*, то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 30, в указанном порядке.

Таблица 30 – Утверждения для испытания ПБУ подкласса механизм идентификации, поддерживающего функцию *BioSPI_Verify*

Номер	Наименование утверждения	Пакет
14a	*BioSPI_Verify_ValidParam*	b78e5be0-efcb-11d9-b2c7-0002a5d5c51b
14b	*BioSPI_Verify_Payload*	32969ec0-eff8-11d9-9831-0002a5d5c51b
14c	*BioSPI_Verify_AuditData*	89719700-f218-11d9-b028-0002a5d5c51b

7.6.6 Если реализация поддерживает базу данных контролируемой ПБУ (см. 19784-1, приложение А, пункт А.4.6.1.2), то данная реализация должна быть подвергнута испытанию путем выполнения всех тестовых утверждений, приведенных в таблице 31, в указанном порядке.

Таблица 31 – Утверждения для испытания ПБУ подкласса механизм идентификации, поддерживающего функции работы с базой данных

Номер	Наименование утверждения	Пакет
15a	*BioSPI_DbOpen_ValidParam*	e68ff9a0-e506-11d9-a6a1-0002a5d5c51b
15b	*BioSPI_DbOpen_InvalidBSPHandle*	bfd44400-e5de-11d9-bdb9-0002a5d5c51b
16a	*BioSPI_DbClose_ValidParam*	39aa9560-e5f1-11d9-89f3-0002a5d5c51b
16b	*BioSPI_DbClose_InvalidBSPHandle*	6e3f5c00-e5f3-11d9-b663-0002a5d5c51b
17a	*BioSPI_DbCreate_DbProtected*	7b6c2f40-e650-11d9-812f-0002a5d5c51b
17b	*BioSPI_DbCreate_ValidParam*	1421ec38-1db6-49d4-873d-03e2de17598b
17c	*BioSPI_DbCreate_InvalidBSPHandle*	ef4bb862-79f6-4f01-8f5d-af5c3abf23c0
18a	*BioSPI_DbDelete_InvalidBSPHandle*	678e5d12-3d51-41ec-a672-13f34ea24545
18b	*BioSPI_DbDelete_OpenDbProtected*	9e4d0c6d-4d59-479c-9f58-160ecde99aad
18c	*BioSPI_DbDelete_ValidParam*	499d9cc3-4269-4671-9d69-29a31bc1a08f

Окончание таблицы 31

Номер	Наименование утверждения	Пакет
19a	*BioSPI_DbSetMarker_ValidParam*	94271080-e723-11d9-898c-0002a5d5c51b
19b	*BioSPI_DbSetMarker_InvalidBSPHandle*	69f7ce20-e72e-11d9-b4e0-0002a5d5c51b
19c	*BioSPI_DbSetMarker_RecordNotFound*	b9c90f40-e7ec-11d9-a435-0002a5d5c51b
20a	*BioSPI_DbFreeMarker_ValidParam*	2e1c9520-e7f1-11d9-a011-0002a5d5c51b
20b	*BioSPI_DbFreeMarker_InvalidBSPHandle*	0f735140-e800-11d9-8a8a-0002a5d5c51b
20c	*BioSPI_DbFreeMarker_InvalidMarker*	a9007b60-e802-11d9-8a5d-0002a5d5c51b
21a	*BioSPI_DbStoreBIR_ValidParam*	da953680-e806-11d9-90d3-0002a5d5c51b
21b	*BioSPI_DbStoreBIR_InvalidBSPHandle*	e39027e0-e80b-11d9-85eb-0002a5d5c51b
22a	*BioSPI_DbGetBIR_ValidParam*	67512700-e811-11d9-a5e0-0002a5d5c51b
22b	*BioSPI_DbGetBIR_InvalidBSPHandle*	f62947e0-e8b7-11d9-9dad-0002a5d5c51b
22c	*BioSPI_DbGetBIR_RecordNotFound*	37457440-e8ba-11d9-87da-0002a5d5c51b
23a	*BioSPI_DbGetNextBIR_ValidParam*	e3396400-e8c9-11d9-990f-0002a5d5c51b
23b	*BioSPI_DbGetNextBIR_InvalidBSPHandle*	f0ef5320-e8d3-11d9-bbc1-0002a5d5c51b
24a	*BioSPI_DbDeleteBIR_ValidParam*	ed4afbe0-e8d6-11d9-9ed0-0002a5d5c51b
24b	*BioSPI_DbDeleteBIR_InvalidBSPHandle*	72eca940-e8d9-11d9-aa0d-0002a5d5c51b

8 Тестовые утверждения

8.1 Основные положения

8.1.1 В настоящем разделе приведены все тестовые утверждения, связанные с ПБУ, а также тестовые утверждения для ПБУ всех подклассов соответствия.

8.1.2 В 8.2 установлены требования к пакету, содержащему процессы, общие для всех тестовых утверждений.

8.2 Общие процессы

В настоящем подразделе установлен пакет, включающий в себя процессы, на которые ссылаются многие тестовые утверждения.

```
<package name="02c59458-0c46-1085-95d7-0002a5d5fd2e">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит несколько основных процессов,
        вызываемых процессами, входящими в другие пакеты.
    </description>

    <activity name="LoadAndAttach">
        <input name="bspUuid"/>
        <input name="bspVersion"/>
        <input name="unitIDOrNull"/>
        <input name="bspHandle"/>
        <input name="eventtimeouttime"/>

        <!-- Инициализация глобальной переменной "_unitIDOrNull" для
        обеспечения доступа к ней из процесса "EventHandler". -->
        <set name="_unitIDOrNull" var="unitIDOrNull"/>

        <!-- Инициализация глобальной переменной "_insert"
        значением "false". Процесс "EventHandler" присваивает
        данной переменной значение "true" после получения
        уведомления о событии BioAPI_NOTIFY_INSERT и значение
        "false" после получения уведомления о событии
        BioAPI_NOTIFY_REMOVE.-->
        <set name="_insert" value="false"/>
```

```
<!-- Инициализация глобальной переменной "_sourcePresent"
значением "false". Процесс "EventHandler" присваивает
данной переменной значение "true" после получения
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT и
значение "false" после получения уведомления о событии
BioAPI_NOTIFY_SOURCE_REMOVED. -->
<set name="_sourcePresent" value="false"/>

<set name="eventtimeoutflag" value="false"/>

<!-- Вызов функции BioSPI_BSPLoad. -->
<invoke function="BioSPI_BSPLoad">
     <input name="BSPUuid" var="bspUuid"/>
     <input name="BioAPINotifyCallback" value="*"/>
     <input name="BFPEnumerationHandler" value="*"/>
     <input name="MemoryFreeHandler" value="*" />
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
описанное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и
выполнение процесса прерывается, в противном случае
выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
     break_if_false="true">
<description>
     Функция BioSPI_BSPLoad возвращает значение BioAPI_OK.
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Уведомление о событии должно быть получено в
BioAPI_NOTIFY_INSERT в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="eventtimeouttime"
     setvar="eventtimeoutflag" var="_insert"/>
<!-- Выдача заключения о соответствии.
```

```
        Если условие, описанное далее в элементе <description>,
        ложное, выдается заключение о соответствии "UNDECIDED" и
        выполнение процесса прерывается, в противном случае
        выдается заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Уведомление о событии BioAPI_NOTIFY_INSERT получено
            в пределах указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
    </assert_condition>

    <!-- Вызов функции BioSPI_BSPAttach. -->
    <invoke function="BioSPI_BSPAttach">
        <input name="BSPUuid" var="bspUuid"/>
        <input name="Version" var="bspVersion"/>
        <input name="Unit_1_UnitCategory" var="_unitCategory"/>
        <input name="Unit_1_UnitID" var="_unitID"/>
        <input name="NumUnits" value="1" />
        <input name="BSPHandle" var="bspHandle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, описанное
    далее в элементе <description>, ложное, выдается заключение о
    соответствии "UNDECIDED" и выполнение процесса прерывается, в
    противном случае выдается заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_BSPAttach возвращает значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>
```

```
</activity>

<!-- Данный процесс выполняется при входящем вызове функции
BioSPI_ModuleEventHandler тестируемым компонентом ПБУ. В данном
процессе значения глобальным переменным "_unitID", "_insert",
"_sourcePresent" и "_eventtype" присваиваются в зависимости от
значений входных параметров. -->
<activity name="EventHandler" atomic="true">
     <input name="BSPUuid"/>
     <input name="UnitID"/>
     <input name="UnitSchema_BspUuid" />
     <input name="UnitSchema_UnitManagerUuid" />
     <input name="UnitSchema_UnitId" />
     <input name="UnitSchema_UnitCategory" />
     <input name="UnitSchema_UnitProperties" />
     <input name="UnitSchema_VendorInformation" />
     <input name="UnitSchema_EventNotifyInsert" />
     <input name="UnitSchema_EventNotifyRemove" />
     <input name="UnitSchema_EventNotifyFault" />
     <input name="UnitSchema_EventNotifySourcePresent" />
     <input name="UnitSchema_EventNotifySourceRemoved" />
     <input name="UnitSchema_UnitPropertyID" />
     <input name="UnitSchema_UnitProperty" />
     <input name="UnitSchema_HardwareVersion" />
     <input name="UnitSchema_FirmwareVersion" />
     <input name="UnitSchema_SoftwareVersion" />
     <input name="UnitSchema_HardwareSerialNumber" />
     <input name="UnitSchema_AuthenticatedHardware" />
     <input name="UnitSchema_MaxBspDbSize" />
     <input name="UnitSchema_MaxIdentify" />
     <input name="EventType" />
     <output name="return"/>
     <!-- Присвоение начального значения переменной "_unitID"
     выполняется если:
          - оно не было присвоено;
```

```
        - получено уведомление о событии BioAPI_NOTIFY_INSERT или
        BioAPI_NOTIFY_SOURCE_PRESENT;
        - событие связано с ожидаемым модулем в соответствии со
        значением параметра "unitIDOrNULL" вызова процесса
        "LoadAndAttach" -->
    <set name="_unitID" var="UnitID">
        <only_if>
            <not>
                <existing var="_unitID"/>
            </not>
            <or>
                <equal_to var1="EventType"
                    var2="__BioAPI_NOTIFY_INSERT"/>
                <equal_to var1="EventType"
                    var2="__BioAPI_NOTIFY_SOURCE_PRESENT"/>
            </or>
            <or>
                <equal_to var1="_unitIDOrNull" value2="0"/>
                <equal_to var1="_unitIDOrNull" var2="UnitID"/>
            </or>
        </only_if>
    </set>

    <!-- Установка категории модуля. -->
    <set name="_unitCategory" var="UnitSchema_UnitCategory">
        <only_if>
            <not>
                <existing var="_unitCategory"/>
            </not>
            <equal_to var1="EventType"
                var2="__BioAPI_NOTIFY_INSERT"/>
            <or>
                <equal_to var1="_unitIDOrNull" value2="0"/>
                <equal_to var1="_unitIDOrNull" var2="UnitID"/>
            </or>
```

```
            </only_if>
        </set>

        <invoke activity="EventHandlerSetGlobalData">
            <only_if>
                <existing var="_unitID"/>
            </only_if>
            <input name="UnitID" var="UnitID"/>
            <input name="EventType" var="EventType"/>
        </invoke>

        <set name="return" var="__BioAPI_OK"/>
</activity>


<activity name="EventHandlerSetGlobalData" atomic="true">
        <input name="UnitID"/>
        <input name="EventType"/>

        <!-- Присвоить глобальной переменной "_insert" начальное
        значение "true" если:
            - поступило уведомление о событии BioAPI_NOTIFY_INSERT;
            - событие связано с ожидаемым модулем, определенным
            параметром "unitIDOrNULL" или вызовом процесса
            "LoadAndAttach". -->
        <set name="_insert" value="true">
            <only_if>
                <equal_to var1="EventType"
                var2="__BioAPI_NOTIFY_INSERT"/>
                <equal_to var1="_unitID" var2="UnitID"/>
            </only_if>
        </set>

        <!-- Присвоить глобальной переменной "_insert" начальное
        значение "false", если:
            - поступило уведомление о событии BioAPI_NOTIFY_REMOVE;
```

```
        - событие связано с ожидаемым модулем, определенным
        параметром "unitIDOrNULL" или вызовом процесса
        "LoadAndAttach". -->
    <set name="_insert" value="false">
        <only_if>
            <equal_to var1="EventType"
            var2="__BioAPI_NOTIFY_REMOVE"/>
            <equal_to var1="_unitID" var2="UnitID"/>
        </only_if>
    </set>

    <!-- Присвоить глобальной переменной "_sourcePresent"
    начальное значение "true", если:
        - поступило уведомление о событии
        BioAPI_NOTIFY_SOURCE_PRESENT;
        - событие связано с ожидаемым модулем, определенным
        параметром "unitIDOrNULL" или вызовом процесса
        "LoadAndAttach".-->
    <set name="_sourcePresent" value="true">
        <only_if>
            <equal_to var1="EventType"
                var2="__BioAPI_NOTIFY_SOURCE_PRESENT"/>
            <equal_to var1="_unitID" var2="UnitID"/>
        </only_if>
    </set>

    <!-- Присвоить глобальной переменной "_sourcePresent"
    начальное значение "false", если:
        - поступило уведомление о событии
        BioAPI_NOTIFY_SOURCE_REMOVED;
        - событие связано с ожидаемым модулем, определенным
        параметром "unitIDOrNULL" или вызовом процесса
        "LoadAndAttach". -->
    <set name="_sourcePresent" value="false">
        <only_if>
```

```
                <equal_to var1="EventType"
                    var2="__BioAPI_NOTIFY_SOURCE_REMOVED"/>
                <equal_to var1="_unitID" var2="UnitID"/>
            </only_if>
        </set>

        <!-- Присвоение начального значения глобальной переменной
        "_eventtype". -->
        <set name="_eventtype" var="EventType">
            <only_if>
                <equal_to var1="_unitID" var2="UnitID"/>
            </only_if>
        </set>
    </activity>


    <!-- Данный процесс вызывает функции BioSPI_BSPDetach и
    BioSPI_BSPUnload. -->
    <activity name="DetachAndUnload" >
        <input name="bspUuid"/>
        <input name="BSPHandle"/>

        <!-- Вызов функции BioSPI_BSPDetach. -->
        <invoke function="BioSPI_BSPDetach" >
            <input name="BSPHandle" var="BSPHandle"/>
            <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие, описанное
        далее в элементе <description>, ложное, выдается заключение о
        соответствии "UNDECIDED" и выполнение процесса прерывается, в
        противном случае выдается заключение о соответствии "PASS" . -->
        <assert_condition response_if_false="undecided"
                break_if_false="true">
            <description>
                Функция BioSPI_BSPDetach возвращает значение BioAPI_OK.
```

```
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Вызов функции BioSPI_BSPUnload. -->
        <invoke function="BioSPI_BSPUnload" >
            <input name="BSPUuid" var="bspUuid" />
            <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное, выдается
        заключение о соответствии "UNDECIDED" и выполнение процесса
        прерывается, в противном случае выдается заключение о
        соответствии "PASS".-->
        <assert_condition response_if_false="undecided"
                break_if_false="true">
            <description>
                Функция BioSPI_BSPUnload возвращает значение BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
    </activity>


    <activity name="check_capturedBIR_datatype">
        <input name="BSPHandle" />
        <input name="BIRHandle" />

        <!-- Вызов функции BioSPI_GetHeaderFromHandle для полученной
        ЗБИ. -->
        <invoke function="BioSPI_GetHeaderFromHandle">
            <input name="BSPHandle" var="BSPHandle"/>
            <input name="Handle" var="BIRHandle"/>
            <output name="ProcessedLevel" setvar="processedLevel"/>
```

```
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное, выдается
    заключение о соответствии "UNDECIDED" и выполнение процесса
    прерывается, в противном случае выдается заключение о
    соответствии "PASS".-->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_GetHeaderFromHandle возвращает
            значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное, выдается
    заключение о соответствии "UNDECIDED" и выполнение процесса
    прерывается, в противном случае выдается заключение о
    соответствии "PASS".-->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Степень обработки полученной ЗБИ не соответствует
            степени "PROCESSED".
        </description>
        <not_equal_to var1="processedLevel"
            var2="__BioAPI_BIR_DATA_TYPE_PROCESSED" />
    </assert_condition>
</activity>


<activity name="process_bir">
    <input name="BSPHandle" />
```

```
    <input name="CapturedBIR_BIRHandle" />

    <!-- Вызов функции BioSPI_Process. -->
    <invoke function="BioSPI_Process">
        <input name="BSPHandle" var="BSPHandle"/>
        <input name="CapturedBIR_Form"
            var="__BioAPI_BIR_HANDLE_INPUT" />
        <input name="CapturedBIR_BIRHandle"
            var="CapturedBIR_BIRHandle"/>
        <output name="ProcessedBIR" set-
        var="_processedbir_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное, выдается
    заключение о соответствии "UNDECIDED" и выполнение процесса
    прерывается, в противном случае выдается заключение о
    соответствии "PASS".-->

    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_Process возвращает значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>
</activity>


<activity name="payloadSupport_checkPayload" >
    <input name="inputPayload"/>
    <input name="outputPayload"/>
    <input name="result"/>
    <input name="payloadPolicy"/>
```

```
    <input name="fmrAchieved"/>
    <invoke activity="resultFalse_checkPayload"
            break_on_break="true">
        <only_if>
            <same_as var1="result" value2="false"/>
        </only_if>
        <input name="outputPayload" var="outputPayload" />
    </invoke>

    <invoke activity="resultTrue_checkPayload"
            break_on_break="true">
        <only_if>
            <same_as var1="result" value2="true" />
        </only_if>
        <input name="inputPayload" var="inputPayload" />
        <input name="outputPayload" var="outputPayload" />
        <input name="payloadPolicy" var="payloadPolicy" />
        <input name="fmrAchieved" var="fmrAchieved"/>
    </invoke>
</activity>

<activity name="payloadNotSupport_checkPayload" >
    <input name="outputPayload"/>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное, выдается
    заключение о соответствии "FAIL" и выполнение процесса
    прерывается, в противном случае выдается заключение о
    соответствии "PASS".-->
    <assert_condition break_if_false="true">
        <description>
            Данные не были возвращены.
        </description>
        <same_as var1="outputPayload" value2=""/>
    </assert_condition>
```

```
</activity>

<activity name="resultFalse_checkPayload" >
     <input name="outputPayload"/>

     <!-- Выдача заключения о соответствии. Если условие,
     определенное далее в элементе <description>, ложное, выдается
     заключение о соответствии "FAIL" и выполнение процесса
     прерывается, в противном случае выдается заключение о
     соответствии "PASS".-->
     <assert_condition break_if_false="true">
          <description>
               ПБУ поддерживает выдачу данных, при этом результат
               является ложным, и данные не были возвращены.
          </description>
          <same_as var1="outputPayload" value2=""/>
     </assert_condition>
</activity>


<activity name="resultTrue_checkPayload">
     <input name="inputPayload"/>
     <input name="outputPayload"/>
     <input name="payloadPolicy"/>
     <input name="fmrAchieved"/>
     <!-- Выдача заключения о соответствии. Если условие,
     определенное далее в элементе <description>, ложное, выдается
     заключение о соответствии "UNDECIDED" и выполнение процесса
     прерывается, в противном случае выдается заключение о
     соответствии "PASS".-->

     <assert_condition response_if_false="undecided"
          break_if_false="true">
          <description>
```

```
            Возвращаемые данные совпадают с представленными или
            отсутствуют, поскольку вероятность ошибки ложного
            совпадения слишком велика.
        </description>
        <or>
            <and>
                <less_than var1="fmrAchieved"
                var2="payloadPolicy"/>
                <same_as var1="outputPayload"
                var2="inputPayload"/>
            </and>
            <and>
                <greater_than_or_equal_to var1="fmrAchieved"
                var2="payloadPolicy"/>
                <same_as var1="outputPayload" value2=""/>
            </and>
        </or>
    </assert_condition>
</activity>

<!-- Данный процесс обеспечивает проверку значения возвращенной
после адаптации ЗБИ и используется в случае, если ПБУ поддерживает
адаптацию шаблона. -->
<activityname="check_adaptation_supported">
    <input name="adaptedbir_handle" />

    <assert_condition response_if_false="undecided"
            break_if_false="true" >
        <description>
            Дескриптор адаптированной ЗБИ имеет действительное
            значение.
        </description>
        <not_equal_to var1="adaptedbir_handle" value2="-1" />
    </assert_condition>
```

```
    <assert_condition break_if_false="true">
        <description>
            Дескриптор адаптированной ЗБИ имеет неотрицательное
            значение.
        </description>
    <greater_than_or_equal_to var1="adaptedbir_handle"
            value2="0"/>
    </assert_condition>
</activity>

<!-- Данный процесс обеспечивает проверку значения возвращенной
после адаптации ЗБИ и используется в случае,если ПБУ
не поддерживает адаптацию шаблона. -->
<activityname="check_adaptation_not_supported">
    <input name="adaptedbir_handle"/>

    <assert_condition break_if_false="true">
        <description>
            Адаптация не поддерживается ПБУ и ПБУ возвращает
            значение BioAPI_UNSUPPORTED_BIR_HANDLE ("-2").
        </description>
        <equal_to var1="adaptedbir_handle" value2="-2" />
    </assert_condition>
</activity>


<!-- Данный процесс обеспечивает проверку возвращаемого значения
качества, если возвращение обработанного значения качества
поддерживается ПБУ. -->
<activity name="check_quality_supported" >
    <input name="quality"/>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное, выдается
```

```
заключение о соответствии "UNDECIDED" и выполнение процесса
прерывается, в противном случае выдается заключение
о соответствии "PASS".-->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Значение качества является допустимым.
     </description>
     <not_equal_to var1="quality" value2="-1"/>
</assert_condition>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное, выдается
заключение о соответствии "FAIL" и выполнение процесса
прерывается, в противном случае выдается заключение
о соответствии "PASS".-->
<assert_condition break_if_false="true">
     <description>
          Качество поддерживается ПБУ и имеет значение
          в пределах от 0 до 100.
     </description>
     <greater_than_or_equal_to var1="quality" value2="0"/>
     <less_than_or_equal_to var1="quality" value2="100"/>
</assert_condition>
</activity>

<!-- Данный процесс обеспечивает проверку возвращаемого значения
качества, если возвращение обработанного значения качества
не поддерживается ПБУ.-->
<activity name="check_quality_not_supported" >
     <input name="quality"/>
     <assert_condition break_if_false="true">
          <description>
               Возвращение значения качества не поддерживается ПБУ,
               и возвращаемым значением является "-2".
```

```
            </description>
            <equal_to var1="quality" value2="-2"/>
        </assert_condition>
</activity>

<!-- Данный процесс обеспечивает создание базы данных. -->
<activity name="PrepareDBTesting" >
        <input name="bspHandle" />
        <input name="dbUuid"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <output name="biruuid"/>

        <invoke function="BioSPI_DbDelete" >
            <input name="BSPHandle" var="bspHandle" />
            <input name="DbUuid" var="dbUuid" />
            <return setvar="return"/>
        </invoke>

        <invoke function="BioSPI_DbCreate">
            <input name="BSPHandle" var="bspHandle"/>
            <input name="DbUuid" var="dbUuid"/>
            <input name="NumberOfRecords" value="1"/>
            <input name="ReadAccessRequest" value="true"/>
            <input name="WriteAccessRequest" value="true"/>
            <output name="DbHandle" setvar="dbHandle"/>
            <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное, выдается
        заключение о соответствии "UNDECIDED", в противном случае
        выдается заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
```

```
          break_if_false="true">
     <description>
          Функция BioSPI_DbCreate возвращает значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported" var2="_sourcePresent" />
</wait_until>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное, выдается
заключение о соответствии "UNDECIDED" и выполнение процесса
прерывается, в противном случае выдается заключение
о соответствии "PASS".-->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Или испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT, или
          уведомление о событии получено в пределах указанной
          максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению данных. Вызвать функцию
BioSPI_Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
```

```
        <input name="BSPHandle" var="bspHandle"/>
        <input name="Purpose"
            var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
        <input name="Timeout" value="15000"/>
        <output name="NewTemplate" setvar="newtemplate_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное, выдается
    заключение о соответствии "UNDECIDED" и выполнение процесса
    прерывается, в противном случае выдается заключение
    о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_Enroll возвращает значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Вызвать функцию BioSPI_DbStoreBIR для сохранения
    зарегистрированной ЗБИ в базе данных. -->
    <invoke function="BioSPI_DbStoreBIR">
        <input name="BSPHandle" var="bspHandle"/>
        <input name="BIRToStore_Form"
        var="__BioAPI_BIR_HANDLE_INPUT"/>
        <input name="BIRToStore_BIRHandle"
        var="newtemplate_handle"/>
        <input name="DbHandle" var="dbHandle"/>
        <input name="no_BirUuid" value="false"/>
        <output name="BirUuid" setvar="biruuid"/>
        <return setvar="return"/>
    </invoke>
```

```
    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное, выдается
    заключение о соответствии "FAIL", в противном случае выдается
    заключение о соответствии "PASS".-->
    <assert_condition response_if_false="undecided"
              break_if_false="true">
         <description>
              Функция BioSPI_DbStoreBIR возвращает значение
              BioAPI_OK и выходное значение УУИД не равно NULL.
         </description>
         <equal_to var1="return" var2="__BioAPI_OK"/>
         <not_same_as var1="biruuid" value2=""/>
    </assert_condition>

    <!-- Вызов функции BioSPI_DbClose с достоверными значениями
    входных параметров. -->
    <invoke function="BioSPI_DbClose">
         <input name="BSPHandle" var="bspHandle"/>
         <input name="DbHandle" var="dbHandle"/>
         <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное, выдается
    заключение о соответствии "UNDECIDED", в противном случае
    выдается заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
              break_if_false="true">
         <description>
              Функция BioSPI_DbClose возвращает значение BioAPI_OK.
         </description>
         <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>
</activity>
```

```
<!-- Данный процесс обеспечивает удаление базы данных. -->
<activity name="CleanUpDBTesting" >
     <input name="BSPHandle" />
     <input name="dbUuid"/>

     <!-- Вызов функции BioSPI_DbDelete с допустимыми значениями
     входных параметров. -->
     <invoke function="BioSPI_DbDelete">
          <input name="BSPHandle" var="_bsphandle"/>
          <input name="DbUuid" var="dbUuid"/>
          <return setvar="return"/>
     </invoke>

     <!-- Выдача заключения о соответствии. Если условие,
     определенное далее в элементе <description>, ложное, выдается
     заключение о соответствии "UNDECIDED",в противном случае
     выдается заключение о соответствии "PASS".-->
     <assert_conditionresponse_if_false="undecided">
          <description>
               Функция BioSPI_DbDelete возвращает значение BioAPI_OK.
          </description>
          <equal_to var1="return" var2="__BioAPI_OK"/>
     </assert_condition>

</activity>
</package>
```

8.3 Утверждение 1a BioSPI_BSPLoad_InvalidUUID

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPIERR_H_FRAMEWORK_INVALID_UUID при вызове функции *BioSPI_BSPLoad* с недостоверным значением входного параметра УУИД.

Выдержки

Подпункт 9.3.1.1

*BioAPI_RETURN BioAPI BioSPI_BSPLoad(constBioAPI_UUID *BSPUuid,*

BioSPI_EventHandler BioAPINotifyCallback,

BioSPI_BFP_ENUMERATION_HANDLER BFPEnumerationHandler,

BioSPI_MEMORY_FREE_HANDLER MemoryFreeHandler).

Данная функция завершает процесс инициализации между БиоАПИ и ПБУ.

Возвращаемое значение

Значение BioAPI_RETURN указывает на успешное выполнение функции или определяет тип ошибки. Значение BioAPI_OK соответствует успешному выполнению. Остальные значения определяют тип ошибки.

Значение BSPUuid определяет вызванный модуль.

Подпункт 11.2.3

Ошибки: BioAPIERR_INVALID_UUID.

Ссылки: 9.3.1.1 и 11.2.3.

Порядок действий:

1) Вызвать функцию *BioSPI_BSPLoad* с недостоверным значением входного параметра УУИД.

2) Проверить возвращаемое значение. Если это значение BioAPIERR_INVALID_UUID, выдать заключение о соответствии "PASS", в противном случае выдать заключение о соответствии "FAIL".

Если какие-либо промежуточные операции прошли неуспешно, выдать заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_BSPLoad* возвращает значение BioAPIERR_INVALID_UUID.

Пакет языка утверждения

```
<package name="020e90c8-0c19-1085-ab54-0002a5d5fd2e">
    <author>
        ISO/IEC JTC SC37
    </author>

    <description>
```

```
        Данный пакет содержит утверждение
        "BioSPI_BSPLoad_InvalidUUID" (см. далее элемент
        утверждения <description>).
    </description>

    <assertion name="BioSPI_BSPLoad_InvalidUUID" model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью
            определения возвращения значения BioAPIERR_H_
            FRAMEWORK_INVALID_UUID при вызове функции
            BioSPI_BSPLoad с недостоверным значением входного
            параметра УУИД.
            Следующий текст соответствует приведенному в 9.3.1.1
            и 11.2.3 спецификации БиоАПИ 2.0
            ----------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_BSPLoad(const
            BioAPI_UUID *BSPUuid,
                BioSPI_EventHandler BioAPINotifyCallback,
                BioSPI_BFP_ENUMERATION_HANDLER
                BFPEnumerationHandler,
                BioSPI_MEMORY_FREE_HANDLER MemoryFreeHandler).

            Данная функция завершает процесс инициализации между
            БиоАПИ и модулем биометрической услуги.

            Возвращаемое значение
            Значение BioAPI_RETURN указывает на успешное
            выполнение функции или определяет тип ошибки.
            Значение BioAPI_OK соответствует успешному
            выполнению функции. Остальные значения определяют
            тип ошибки.
            ----------------------------------------------------
            Подпункт 9.3.1.1:
            BSPUuid определяет вызванный модуль.
            Пункт 11.2.3:
```

```
            Ошибки: BioAPIERR_INVALID_UUID.

            Порядок действий при испытании на соответствие:

                1) Вызвать функцию BioSPI_BSPLoad
                с недостоверным значением входного параметра
                УУИД.
                2) Проверить возвращаемое значение. Если это
                значение BioAPIERR_INVALID_UUID, то выдать
                заключение о соответствии "PASS", в противном
                случае выдать заключение о соответствии "FAIL".

            Если какие-либо промежуточные операции прошли
            неуспешно, выдать заключение о соответствии
            "UNDECIDED".
        </description>

        <!-- Вызов основного процесса данного утверждения
        со значениями входных параметров, определенными
        значениями параметров утверждения. -->
        <invoke activity="BioSPI_BSPLoad">
            <input name="bspUuid"
            value="00000000-0000-0000-0000-000000000000"/>
        </invoke>
    </assertion>

    <activity name="BioSPI_BSPLoad">
        <input name="bspUuid"/>

        <!-- Вызов функции BioSPI_BSPLoad. -->
        <invoke function="BioSPI_BSPLoad">
            <input name="BSPUuid" var="bspUuid"/>
            <input name="BioAPINotifyCallback" value="*"/>
            <input name="BFPEnumerationHandler" value="*"/>
            <input name="MemoryFreeHandler" value="*" />
            <return setvar="return"/>
```

```
        </invoke>
        <!-- Выдача заключения о соответствии. Если условие, оп-
        ределенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS".-->
        <assert_condition>
            <description>
                Функция BioSPI_BSPLoad возвращает значение
                BioAPIERR_INVALID_UUID.
            </description>
            <equal_to var1="return"
                var2="__BioAPIERR_BSP_INVALID_UUID"/>
        </assert_condition>
    </activity>
</package>
```

8.4 Утверждение 1b BioSPI_BSPLoad_ValidParam

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_BSPLoad* с достоверными значениями входных параметров.

Выдержки

Подпункт 9.3.1.1

*BioAPI_RETURN BioAPI BioSPI_BSPLoad (const BioAPI_UUID *BSPUuid,*
BioSPI_EventHandler BioAPINotifyCallback,
BioSPI_BFP_ENUMERATION_HANDLER BFPEnumerationHandler,
BioSPI_MEMORY_FREE_HANDLER MemoryFreeHandler).

Данная функция завершает процесс инициализации между БиоАПИ и ПБУ.

Возвращаемое значение

Значение BioAPI_RETURN указывает на успешное выполнение функции или определяет тип ошибки. Значение BioAPI_OK соответствует успешному выполнению. Остальные значения определяют тип ошибки.

Значение BSPUuid определяет вызванный ПБУ.

Значение BioAPINotifyCallback определяет обратный вызов, используемый для оповещения инфраструктуры БиоАПИ о событиях с типом BioAPI_EVENT в любой текущей присоединенной сессии. ПБУ должен сохранить эту информацию для использования в дальнейшем.

Значение BFPEnumerationHandler представляет собой адрес обратного вызова обработчика перечислений ПБФ, представленного инфраструктурой ПБУ. ПБУ может использовать обратный вызов для получения информации о ПБФ, имеющихся в инфраструктуре.

Значение MemoryFreeHandler представляет собой адрес обратного вызова обработчика освобождения памяти, представленного инфраструктурой ПБУ. ПБУ должен сохранить эту информацию для использования в дальнейшем. ПБУ должен использовать обратный вызов для освобождения блока памяти, который был выделен инфраструктурой во время предыдущего обратного вызова обработчика перечислений ПБФ.

Раздел А.4

Данная функция должна поддерживаться всеми типами ПБУ.

Ссылки: 9.3.1.1 и А.4.

Порядок действий:

1) Вызвать функцию *BioSPI_BSPLoad* с достоверными значениями входных параметров.

2) Проверить возвращаемое значение. Если это значение BioAPI_OK, выдать заключение о соответствии "PASS", в противном случае выдать заключение о соответствии "FAIL".

Если какие-либо промежуточные операции прошли неуспешно, выдать заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_BSPLoad* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="01f6c6f0-0c19-1085-97fe-0002a5d5fd2e">
    <author>
```

```
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение
        "BioSPI_BSPLoad_ValidParam" (см. далее элемент
        утверждения <description>).
    </description>

    <assertion name="BioSPI_BSPLoad_ValidParam"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью
            определения возвращения значения функции BioAPI_OK
            при вызове функции BioSPI_BSPLoad с достоверными
            значениями входных параметров.
            Далее приведены выдержки из 9.3.1.1 и A.4
            спецификации БиоАПИ 2.0
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_BSPLoad(const
            BioAPI_UUID *BSPUuid,
                BioSPI_EventHandler BioAPINotifyCallback,
                BioSPI_BFP_ENUMERATION_HANDLER
                BFPEnumerationHandler,
                BioSPI_MEMORY_FREE_HANDLER MemoryFreeHandler).

            Данная функция завершает процесс инициализации между
            БиоАПИ и модулем биометрической услуги.

            Возвращаемое значение
            Значение BioAPI_Return указывает на успешное
            выполнение функции или определяет тип ошибки.
            Значение BioAPI_OK соответствует успешному
            выполнению. Остальные значения определяют тип
            ошибки.
```

Подпункт 9.3.1.1:

Данная функция завершает процесс инициализации между БиоАПИ и ПБУ.

Значение BSPUuid определяет вызванный ПБУ.

Значение BioAPINotifyCallback определяет обратный вызов, используемый для оповещения инфраструктуры БиоАПИ о событиях с типом BioAPI_EVENT в любой текущей присоединенной сессии. ПБУ должен сохранить эту информацию для использования в дальнейшем.

Значение BFPEnumerationHandler представляет собой адрес обратного вызова обработчика перечислений ПБФ, представленного инфраструктурой ПБУ. ПБУ может использовать обратный вызов для получения информации о ПБФ, имеющихся в инфраструктуре.

Значение MemoryFreeHandler представляет собой адрес обратного вызова обработчика освобождения памяти, представленного инфраструктурой ПБУ. ПБУ должен сохранить эту информацию для использования в дальнейшем. ПБУ должен использовать обратный вызов для освобождения блока памяти, который был выделен инфраструктурой во время предыдущего обратного вызова обработчика перечислений ПБФ.

Раздел А.4:

Данная функция должна поддерживаться всеми типами ПБУ.

Порядок действий при испытании на соответствие:

1) Вызвать функцию BioSPI_BSPLoad с достоверными значениями входных параметров.

2) Проверить возвращаемое значение. Если это значение BioAPI_OK, выдать заключение о соответствии "PASS", в противном случае выдать заключение о соответствии "FAIL".

```
            Если какие-либо промежуточные операции прошли
            неуспешно, то выдать заключение о соответствии
            "UNDECIDED".
        </description>

        <!-- УУИД испытуемого ПБУ. -->
        <input name="_BspUuid"/>

        <!-- Указывает, будет ли предоставлен адрес обратного вызова
        инфраструктуры для уведомления о события ПБУ; значение "*" оз-
        начает предоставление адреса, значение " " - отсутствие. -->
        <input name="_BioAPINotifyCallback" />

        <!-- Адрес обработчика перечислений ПБФ. -->
        <input name="_BFPEnumerationHandler" />

        <!-- Адрес обработчика освобождения памяти. -->
        <input name="_MemoryFreeHandler" />

        <!-- Вызов основного процесса данного утверждения
        со значениями входных параметров, определенными
        значениями параметров утверждения. -->
        <invoke activity="BioSPI_BspLoad">
            <input name="BspUuid" var="_BspUuid"/>
            <input name="BioAPINotifyCallback"
                var="_BioAPINotifyCallback" />
            <input name="BFPEnumerationHandler"
                var="_BFPEnumerationHandler" />
            <input name="MemoryFreeHandler"
                var="_MemoryFreeHandler" />
        </invoke>
    </assertion>

    <activity name="BioSPI_BspLoad">
        <input name="BspUuid"/>
```

```
<input name="BioAPIdNotifyCallback"/>
<input name="BFPEnumerationHandler" />
<input name="MemoryFreeHandler" />

<!-- Вызов функции BioSPI_BspLoad. -->
<invoke function="BioSPI_BSPLoad">
     <input name="BSPUuid" var="BspUuid"/>
     <input name="BioAPINotifyCallback"
          var="BioAPIdNotifyCallback"/>
     <input name="BFPEnumerationHandler"
          var="BFPEnumerationHandler"/>
     <input name="MemoryFreeHandler"
          var="MemoryFreeHandler" />
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдать заключение о соответствии "FAIL", в противном
случае выдать заключение о соответствии "PASS".-->
<assert_condition>
     <description>
          Функция BioSPI_BSPLoad возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_BSPUnload. -->
<invoke function="BioSPI_BSPUnload">
     <input name="BSPUuid" var="BspUuid"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
```

```
            выдается заключение о соответствии "UNDECIDED",
            в противном случае выдается заключение о соответствии
            "PASS".-->
            <assert_condition response_if_false="undecided">
                <description>
                    Функция BioSPI_BSPUnload возвращает значение
                    BioAPI_OK.
                </description>
                <equal_to var1="return" var2="__BioAPI_OK"/>
            </assert_condition>
        </activity>
</package>
```

8.5 Утверждение 2a BioSPI_BSPUnload_ValidParam

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_BSPUnload* с достоверными значениями входных параметров.

Выдержки

Подпункт 9.3.1.2

BioAPI_RETURN BioAPI BioSPI_BSPUnload

*(const BioAPI_UUID *BSPUuid).*

Данная функция деактивирует события и аннулирует регистрацию функции БиоАПИ уведомления о событиях. Модуль биометрической услуги может производить операции очистки, отменяющие инициализацию, выполненную функцией *BioSPI_BSPLoad*.

Возвращаемое значение

Значение BioAPI_RETURN указывает на успешное выполнение функции или определяет тип ошибки. Значение BioAPI_OK соответствует успешному выполнению. Остальные значения определяют тип ошибки.

Раздел А.4

Данная функция должна поддерживаться всеми типами ПБУ.

Ссылки: 9.3.1.2 и А.4.

Порядок действий:

1) Вызвать функцию *BioSPI_BSPLoad* с достоверными значениями входных параметров. Предполагается, что вызов будет успешным.

2) Вызвать функцию *BioSPI_BSPUnload* с достоверными значениями входных параметров.

3) Проверить возвращаемое значение. Предполагается возвращение значения BioAPI_OK.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_BSPUnload* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="01661010-0c22-1085-8688-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_BSPUnload_ValidParam" (см. далее элемент
          <description> утверждения).
     </description>

     <assertion name="BioSPI_BSPUnload_ValidParam"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               определения возвращения значения BioAPI_OK
               при вызове функции BioSPI_BSPUnload с достоверными
               значениями входных параметров.
               Следующий текст соответствует приведенным в 9.3.1.2
               и А.4 спецификации БиоАПИ 2.0.
               ------------------------------------------------
```

BioAPI_RETURN BioAPI BioSPI_BSPUnload
(const BioAPI_UUID *BSPUuid).

Данная функция деактивирует события и аннулирует регистрацию функции БиоАПИ уведомления о событиях. Модуль биометрической услуги выполняет операции очистки, отменяющие инициализацию, выполненную функцией BioSPI_BSPLoad.
Возвращаемое значение
Значение BioAPI_RETURN указывает на успешное выполнение функции или определяет тип ошибки. Значение BioAPI_OK соответствует успешному выполнению. Остальные значения определяют тип ошибки.

Подпункт 9.3.1.2:
Данная функция деактивирует события и аннулирует регистрацию функции БиоАПИ уведомления о событиях.
Раздел А.4:
Данная функция должна поддерживаться всеми типами ПБУ.

Порядок действий при испытании на соответствие:

1) Вызвать функцию BioSPI_BSPLoad с достоверными значениями входных параметров. Предполагается, что вызов будет успешным.
2) Вызвать функцию BioSPI_BSPUnload с достоверными значениями входных параметров.
3) Проверить возвращаемое значение. Предполагается возвращение значения BioAPI_OK.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

</description>

```
        <!-- УУИД испытуемого ПБУ. -->
        <input name="_bspUuid"/>

        <!-- Вызов основного процесса данного утверждения
        со значениями входных параметров, определенными
        значениями параметров утверждения. -->
        <invoke activity="BioSPI_BSPUnload">
            <input name="bspUuid" var="_bspUuid"/>
        </invoke>
</assertion>

<activity name="BioSPI_BSPUnload">
        <input name="bspUuid"/>

        <!--Вызов функции BioSPI_BSPLoad. -->
        <invoke function="BioSPI_BSPLoad">
            <input name="BSPUuid" var="bspUuid"/>
            <input name="BioAPINotifyCallback" value="*"/>
            <input name="BFPEnumerationHandler" value="*"/>
            <input name="MemoryFreeHandler" value="*" />
            <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие, оп-
        ределенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED"
        и выполнение процесса прерывается, в противном случае
        выдается заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
                break_if_false="true">
            <description>
                Функция BioSPI_BSPLoad возвращает значение
                BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
```

```
        </assert_condition>

        <!-- Вызов функции BioSPI_BSPUnload. -->
        <invoke function="BioSPI_BSPUnload">
            <input name="BSPUuid" var="bspUuid"/>
            <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие, оп-
        ределенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Функция BioSPI_BSPUnload возвращает значение
                BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
    </activity>
</package>
```

8.6 Утверждение 2b BioSPI_BSPUnload_InvalidUUID

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPIERR_INVALID_UUID при вызове функции *BioSPI_BSPUnload* с недостоверным значением входного параметра УУИД.

Выдержки

Подпункт 9.3.1.2

BioAPI_RETURN BioAPI BioSPI_BSPUnload

*(const BioAPI_UUID *BSPUuid).*

Данная функция деактивирует события и аннулирует регистрацию функции БиоАПИ уведомления о событиях. Модуль биометрической услуги может

производить операции очистки, отменяющие инициализацию, выполненную функцией *BioSPI_BSPLoad*.

Возвращаемое значение

Значение BioAPI_RETURN указывает на успешное выполнение функции или определяет тип ошибки. Значение BioAPI_OK соответствует успешному выполнению. Остальные значения определяют тип ошибки.

Параметры: BSPUuid (входной параметр) – УУИД вызванного ПБУ.

Подпункт 11.2.3

Ошибки: BioAPIERR_INVALID_UUID.

Ссылки: 9.3.1.2 и 11.2.3.

Порядок действий:

1) Вызвать функцию *BioSPI_BSPLoad* с достоверными значениями входных параметров. Предполагается, что вызов будет успешным.

2) Вызвать функцию *BioSPI_BSPUnload* с недостоверным значением входного параметра УУИД.

3) Проверить возвращаемое значение. Если это значение отличается от значения BioAPIERR_INVALID_UUID, то выдается заключение о соответствии "FAIL".

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_BSPUnload* возвращает значение BioAPIERR_INVALID_UUID.

Пакет языка утверждения

```
<package name="01c2e5b0-0c3b-1085-b31d-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_BSPUnload_InvalidUUID" (см. далее элемент
          <description> утверждения).
     </description>
```

```
<assertion name="BioSPI_BSPUnload_InvalidUUID"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью
          определения возвращения значения
          BioAPIERR_INVALID_UUID при вызове функции
          BioSPI_BSPUnload с недостоверным входным параметром
          УУИД.
          Следующий текст соответствует приведенному в 9.3.1.2
          и 11.2.3 спецификации БиоАПИ 2.0.
          ---------------------------------------------------
          BioAPI_RETURN BioAPI BioSPI_BSPUnload (const
          BioAPI_UUID *BSPUuid).
          Данная функция деактивирует события и аннулирует
          регистрацию функции БиоАПИ уведомления о событиях.
          Модуль биометрической услуги может выполнять
          операции очистки, отменяющие инициализацию,
          выполненную функцией BioSPI_BSPLoad.
          Возвращаемое значение
          Значение BioAPI_RETURN указывает на успешное
          выполнение функции или определяет тип ошибки.
          Значение BioAPI_OK соответствует успешному
          выполнению. Остальные значения определяют тип ошибки.
          ---------------------------------------------------
          Подпункт 9.3.1.2:
          Параметры: ПБУ Uuid (входной параметр) – УУИД
          вызванного ПБУ.
          Пункт 11.2.3:
          Ошибки: BioAPIERR_INVALID_UUID.
          ---------------------------------------------------

          Порядок действий при испытании на соответствие:
```

```
            1) Вызвать функцию BioSPI_BSPLoad с достоверными
            значениями входных параметров. Предполагается, что
            вызов будет успешным.
            2) Вызвать функцию BioSPI_BSPUnload с недостоверным
            УУИД.
            3) Проверить возвращаемое значение. Если это значение
            отличается от значения BioAPIERR_INVALID_UUID,
            выдается заключение о соответствии "FAIL".

            Если какие-либо промежуточные операции прошли
            неуспешно, то выдается заключение о соответствии
            "UNDECIDED".
        </description>

        <!-- УУИД испытуемого ПБУ. -->
        <input name="_bspUuid"/>

        <!-- Вызов основного процесса данного утверждения
        со значениями входных параметров, определенными
        значениями параметров утверждения. -->
        <invoke activity="BioSPI_BSPUnload">
            <input name="bspUuid" var="_bspUuid"/>
        </invoke>
    </assertion>

    <activity name="BioSPI_BSPUnload">
        <input name="bspUuid"/>

        <!-- Вызвать функцию BioSPI_BSPLoad. -->
        <invoke function="BioSPI_BSPLoad">
            <input name="BSPUuid" var="bspUuid"/>
            <input name="BioAPINotifyCallback" value="*"/>
            <input name="BFPEnumerationHandler" value="*"/>
            <input name="MemoryFreeHandler" value="*" />
            <return setvar="return"/>
```

```
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и
выполнение процесса прерывается, в противном случае
выдается заключение о соответствии "PASS".-->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_BSPLoad возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_BSPUnload. -->
<invoke function="BioSPI_BSPUnload">
     <input name="BSPUuid"
          value="00000000-0000-0000-0000-000000000000"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
     <description>
          Функция BioSPI_BSPUnload возвращает значение
          BioAPIERR_INVALID_UUID.
     </description>
     <equal_to var1="return"
          var2="__BioAPIERR_BSP_INVALID_UUID"/>
</assert_condition>
```

```
        <!-- Вызов функции BioSPI_BSPUnload для выгрузки BSP.-->
        <invoke function="BioSPI_BSPUnload">
             <input name="BSPUuid" var="bspUuid"/>
             <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие, оп-
        ределенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS".-->
        <assert_condition>
             <description>
                  Функция BioSPI_BSPUnload возвращает значение
                  BioAPI_OK.
             </description>
             <equal_to var1="return"
                  var2="__BioAPI_OK"/>
        </assert_condition>
    </activity>
</package>
```

8.7 Утверждение 2c *BioSPI_BSPUnload_UnmatchedLoad*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPIERR_BSP_NOT_LOADED при вызове функции *BioSPI_BSPUnload* без соответствующего вызова функции *BioSPI_BSPLoad*.

Выдержки

Подпункт 9.3.1.2

BioAPI_RETURN BioAPI BioSPI_BSPUnload

*(const BioAPI_UUID *BSPUuid).*

Данная функция деактивирует события и аннулирует регистрацию функции БиоАПИ уведомления о событиях. Модуль биометрической услуги может выполнять операции по очистке, отменяющие инициализацию, выполненную функцией *BioSPI_BSPLoad*.

Подпункт 8.1.6.1

Данную функцию следует вызывать для представленного УУИД ПБУ только в том случае, если существует по крайней мере один вызов функции *BioAPI_BSPLoad* для УУИД ПБУ, для которого не был произведен соответствующий вызов функции *BioAPI_BSPUnload*.

Ссылки: 9.3.1.2 и 8.1.6.1.

Порядок действий:

1) Вызвать функцию *BioSPI_BSPLoad* с достоверными значениями входных параметров.

2) Вызвать функцию *BioSPI_BSPUnload* с достоверными значениями входных параметров.

3) Повторно вызвать функцию *BioSPI_BSPUnload* с достоверными значениями входных параметров.

4) Проверить возвращаемое значение. Если этим значением является BioAPIERR_BSP_NOT_LOADED, то испытание прошло успешно, в противном случае испытание прошло неуспешно.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Повторный вызов функции *BioSPI_BSPUnload* возвращает значение BioAPIERR_BSP_NOT_LOADED.

Пакет языка утверждения

```
<package name="02f6c618-0c23-1085-ba89-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_BSPUnload_UnmatchedLoad" (см. далее описание
          элемента <description> утверждения).
     </description>
```

```
<assertion name="BioSPI_BSPUnload_UnmatchedLoad"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью
          определения возвращения значения BioAPIERR_BSP_
          NOT_LOADED при вызове функции BioSPI_BSPUnload без
          соответствующего вызова функции BioSPI_BSPLoad.
          Следующий текст соответствует приведенному в 9.3.1.2
          и 8.1.6.1 спецификации БиоАПИ 2.0.
          ---------------------------------------------------
          BioAPI_RETURN BioAPI BioSPI_BSPUnload
               (const BioAPI_UUID *BSPUuid).
          Данная функция деактивирует события и аннулирует
          регистрацию функции БиоАПИ уведомления о событиях.
          Модуль биометрической услуги выполняет операции
          очистки, отменяющие инициализацию, выполненную
          функцией BioSPI_BSPLoad.
          ---------------------------------------------------
          Подпункт 8.1.6.1
          Данная функция должна вызываться для представленного
          УУИД ПБУ только в том случае, если существует по
          крайней мере один вызов функции BioAPI_BSPLoad для
          этого УУИД ПБУ, для которого не был произведен
          соответствующий вызов функции BioAPI_BSPUnload.
          ---------------------------------------------------
          Порядок действий при испытании на соответствие:

          1) Вызвать функцию BioSPI_BSPLoad с достоверными
          значениями входных параметров.
          2) Вызвать функцию BioSPI_BSPUnload с достоверными
          значениями входных параметров.
          3) Повторно вызвать функцию BioSPI_BSPUnload
          с достоверными значениями входных параметров.
```

```
            4) Проверить возвращаемое значение. Если этим
            значением является BioAPIERR_BSP_NOT_LOADED, то
            испытание прошло успешно, в противном случае
            испытание прошло неуспешно.

            Если какие-либо промежуточные операции прошли
            неуспешно, то выдается заключение о соответствии
            "UNDECIDED".
        </description>

        <!-- УУИД испытуемого ПБУ. -->
        <input name="_bspUuid"/>

        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_BSPUnload_Unmatch">
            <input name="bspUuid" var="_bspUuid"/>
        </invoke>
    </assertion>

    <activity name="BioSPI_BSPUnload_Unmatch">
        <input name="bspUuid"/>

        <!-- Вызов функции BioSPI_BSPLoad. -->
        <invoke function="BioSPI_BSPLoad">
            <input name="BSPUuid" var="bspUuid"/>
            <input name="BioAPINotifyCallback" value="*"/>
            <input name="BFPEnumerationHandler" value="*"/>
            <input name="MemoryFreeHandler" value="*" />
            <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
```

```
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS".-->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_BSPLoad возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_BSPUnload. -->
<invoke function="BioSPI_BSPUnload">
     <input name="BSPUuid" var="bspUuid"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и
выполнение процесса прерывается, в противном случае
выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_BSPUnload возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Повторный вызов функции BioSPI_BSPUnload. -->
<invoke function="BioSPI_BSPUnload">
     <input name="BSPUuid" var="bspUuid"/>
     <return setvar="return"/>
```

```
        </invoke>
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Функция BioSPI_BSPUnload возвращает значение
                BioAPIERR_BSP_NOT_LOADED.
            </description>
            <equal_to var1="return"
                var2="__BioAPIERR_BSP_BSP_NOT_LOADED"/>
        </assert_condition>
    </activity>
</package>
```

8.8 Утверждение 2d *BioSPI_BSPUnload_Confirm*

Описание: Проверку данного утверждения проводят с целью определения факта выгрузки ПБУ при вызове функции *BioSPI_BSPUnload*.

Выдержки

Подпункт 9.3.1.2

BioAPI_RETURN BioAPI BioSPI_BSPUnload

*(const BioAPI_UUID *BSPUuid).*

Данная функция деактивирует события и аннулирует регистрацию функции БиоАПИ уведомления о событиях. Модуль биометрической услуги выполняет операции очистки, отменяющие инициализацию, выполненную функцией *BioSPI_BSPLoad*.

Возвращаемое значение

Значение BioAPI_RETURN указывает на успешное выполнение функции или определяет тип ошибки. Значение BioAPI_OK соответствует успешному выполнению. Остальные значения определяют тип ошибки.

Ссылки: 9.3.1.2.

Порядок действий:

1) Вызвать функцию *BioSPI_BSPLoad* с достоверными значениями входных параметров. Предполагается, что вызов будет успешным.

2) Вызвать функцию *BioSPI_BSPUnload* с достоверными значениями входных параметров.

3) Вызвать функцию *BioSPI_BSPAttach* с достоверными значениями входных параметров. Предполагается возврат значения BioAPIERR_BSP_NOT_LOADED.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_BSPAttach* возвращает значение BioAPIERR_BSP_NOT_LOADED.

Пакет языка утверждения

```
<package name="03daf040-0c3b-1085-a9fd-0002a5d5fd2e">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение "BioSPI_BSPUnload_Confirm"
        (см. далее элемент <description> утверждения).
    </description>

    <assertion name="BioSPI_BSPUnload_Confirm" model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью
            определения факта выгрузки ПБУ при вызове функции
            BioSPI_BSPUnload.
            Следующий текст соответствует приведенному в 9.3.1.2
            спецификации БиоАПИ 2.0
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_BSPUnload
                (constBioAPI_UUID *BSPUuid).
```

```
Данная функция деактивирует события и аннулирует
регистрацию функции БиоАПИ уведомления о событиях.
Модуль биометрической услуги может производить
операции очистки, отменяющие инициализацию,
выполненную функцией BioSPI_BSPLoad.
Возвращаемое значение
Значение BioAPI_RETURN указывает на успешное
выполнение функции или определяет тип ошибки.
Значение BioAPI_OK соответствует успешному
выполнению. Остальные значения определяют тип
ошибки.
-----------------------------------------------------
Порядок действий при испытании на соответствие:

1) Вызвать функцию BioSPI_BSPLoad с достоверными
значениями входных параметров. Предполагается, что
вызов будет успешным.
2) Вызвать функцию BioSPI_BSPUnload с достоверными
значениями входных параметров.
3) Вызвать функцию BioSPI_BSPAttach с достоверными
значениями входных параметров. Предполагается
возврат значения BioAPIERR_BSP_NOT_LOADED.

Если какие-либо промежуточные операции прошли
неуспешно, то выдается заключение о соответствии
"UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_BSPUnload">
```

```
            <input name="bspUuid" var="_bspUuid"/>
        </invoke>
</assertion>

<activity name="BioSPI_BSPUnload">
        <input name="bspUuid"/>

        <!-- Вызов функции BioSPI_BSPLoad. -->
        <invoke function="BioSPI_BSPLoad">
            <input name="BSPUuid" var="bspUuid"/>
            <input name="BioAPINotifyCallback" value="*"/>
            <input name="BFPEnumerationHandler" value="*"/>
            <input name="MemoryFreeHandler" value="*" />
            <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED" и
        выполнение процесса прерывается, в противном случае
        выдается заключение о соответствии "PASS".-->
        <assert_condition response_if_false="undecided"
                break_if_false="true">
            <description>
                Функция BioSPI_BSPLoad возвращает значение
                BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Вызвать функцию BioSPI_BSPUnload.-->
        <invoke function="BioSPI_BSPUnload">
            <input name="BSPUuid" var="bspUuid"/>
            <return setvar="return"/>
        </invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и
выполнение процесса прерывается, в противном случае
выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_BSPUnload возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_BSPAttach для проверки
выгрузки ПБУ. -->
<invoke function="BioSPI_BSPAttach">
     <input name="BSPUuid" var="bspUuid"/>
     <input name="Version" value="32"/>
     <input name="NumUnits" value="0" />
     <input name="BSPHandle" value="1"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
     <description>
          Функция BioSPI_BSPAttach возвращает значение
          BioAPIERR_BSP_NOT_LOADED.
     </description>
     <equal_to var1="return"
          var2="__BioAPIERR_BSP_BSP_NOT_LOADED"/>
```

```
        </assert_condition>
    </activity>
</package>
```

8.9 Утверждение 3а *BioSPI_BSPAttach_ValidParam*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_BSPAttach* с достоверными значениями входных параметров.

Выдержки

Подпункт 9.3.1.3

BioAPI_RETURN BioAPI BioSPI_BSPAttach

*(const BioAPI_UUID *BSPUuid,*

BioAPI_VERSION Version,

*const BioAPI_UNIT_LIST_ELEMENT *UnitList,*

uint32_t NumUnits,

BioAPI_HANDLE BSPHandle).

Данная функция вызывается инфраструктурой при каждом вызове функции *BioAPI_BSPAttach*. Вызов функции *BioAPI_BSPAttach* определяет ПБУ с помощью идентификатора BSPUuid.

ПБУ должен выполнить необходимую инициализацию для вызова нового ПБУ.

Раздел А.4

Данная функция должна поддерживаться всеми типами ПБУ.

Ссылки: 9.3.1.3 и А.4.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Проверить возвращаемое значение вызова функции.

4) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "“UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_BSPAttach* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="00ae6488-0c3d-1085-9912-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_BSPAttach_ValidParam" (см. далее элемент
          <description> утверждения).
     </description>

     <assertion name="BioSPI_BSPAttach_ValidParam"
     model="BSPTesting">
          <description>
               Проверка данного утверждения проводится с целью
               определения возвращения значения BioAPI_OK
               при вызове функции BioSPI_BSPAttach с достоверными
               значениями входных параметров.
               Следующий текст соответствует приведенному в 9.3.1.3
               и А.4 спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioSPI_BSPAttach
                    (const BioAPI_UUID *BSPUuid,
                    BioAPI_VERSION Version,
                    const BioAPI_UNIT_LIST_ELEMENT *UnitList,
                    uint32_t NumUnits,
                    BioAPI_HANDLE BSPHandle).
```

```
      Данная функция вызывается инфраструктурой при каждом
      вызове функции BioAPI_BSPAttach. Вызов функции
      BioAPI_BSPAttach определяет ПБУ с помощью
      идентификатора BSPUuid.
      ПБУ должен выполнить необходимую инициализацию для
      вызова нового ПБУ.
      ------------------------------------------------
      Раздел А.4:
      Данная функция должна поддерживаться всеми типами
      ПБУ.
      ------------------------------------------------
      Порядок действий при испытании на соответствие:

      1) Загрузить испытуемый ПБУ.
      2) Присоединить испытуемый ПБУ.
      3) Проверить возвращаемое значение вызова функции.
      4) Отсоединить и выгрузить ПБУ.

      Если какие-либо промежуточные операции прошли
      неуспешно, то выдается заключение о соответствии
      "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Номер версии испытуемого ПБУ. -->
<input name="_bspVersion"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>
<!-- Дескриптор модуля присоединного ПБУ. -->
<input name="_bsphandle"/>

<!-- Вид модуля БиоАПИ. -->
```

```
        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="LoadAndAttach" >
            <input name="bspUuid" var="_bspUuid"/>
            <input name="bspVersion" var="_bspVersion"/>
            <input name="unitIDOrNull" value="0"/>
            <input name="bspHandle" var="_bsphandle"/>
            <input name="eventtimeouttime"
            var="_inserttimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного
        вызова инфраструктуры, предоставленной испытуемым
        компонентом ПБУ. При каждом входящем вызове функции
        связанный с ней процесс будет вызываться автоматически. -->
        <bindactivity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="LoadAndAttach">
        <input name="bspUuid"/>
        <input name="bspVersion"/>
        <input name="unitIDOrNull"/>
        <input name="bspHandle"/>
        <input name="eventtimeouttime"/>

        <!-- Инициализация глобальной переменной "_unitIDOrNull" с
        целью сделать ее доступной для события "EventHandler". -->
        <set name="_unitIDOrNull" var="unitIDOrNull"/>

        <!-- Инициализация глобальной переменной "_insert" значе-
        нием "false". Событие "EventHandler" присваивает данной
```

```
переменной значение "true", если получено уведомление о
событии BioAPI_NOTIFY_INSERT, и значение "false", если
получено уведомление о событии BioAPI_NOTIFY_REMOVE. -->
<set name="_insert" value="false"/>

<set name="eventtimeoutflag" value="false"/>

<!-- Вызов функции BioSPI_BSPLoad. -->
<invoke function="BioSPI_BSPLoad">
     <input name="BSPUuid" var="bspUuid"/>
     <input name="BioAPINotifyCallback" value="*"/>
     <input name="BFPEnumerationHandler" value="*"/>
     <input name="MemoryFreeHandler" value="*" />
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и
выполнение процесса прерывается, в противном случае
выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_BSPLoad возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Уведомление о событии BioAPI_NOTIFY_INSERT должно
быть получено в течение указанной максимальной
продолжительности. -->
<wait_untiltimeout_var="eventtimeouttime"
     setvar="eventtimeoutflag" var="_insert"/>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED", и
выполнение процесса прерывается, в противном случае
выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Уведомление о событии BioAPI_NOTIFY_INSERT
          получено в течение указанной максимальной
          продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- Вызов функции BioSPI_BSPAttach. -->
<invoke function="BioSPI_BSPAttach">
     <input name="BSPUuid" var="bspUuid"/>
     <input name="Version" var="bspVersion"/>
     <input name="Unit_1_UnitCategory"
     var="_unitCategory" />
     <input name="Unit_1_UnitID" var="_unitID"/>
     <input name="NumUnits" value="1" />
     <input name="BSPHandle" var="bspHandle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition break_if_false="true">
     <description>
```

```
                Функция BioSPI_BSPAttach возвращает значение
                BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Вызов функции BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid"/>
            <input name="BSPHandle" var="bspHandle"/>
        </invoke>
    </activity>
</package>
```

8.10 Утверждение 3b *BioSPI_BSPAttach_InvalidUUID*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPIERR_INVALID_UUID при вызове функции *BioSPI_BSPAttach* с недостоверным значением входного параметра УУИД.

Выдержки

Подпункт 9.3.1.3

BioAPI_RETURN BioAPI BioSPI_BSPAttach

*(const BioAPI_UUID *BSPUuid,*

BioAPI_VERSION Version,

*const BioAPI_UNIT_LIST_ELEMENT *UnitList,*

uint32_t NumUnits,

BioAPI_HANDLE BSPHandle).

Данная функция вызывается инфраструктурой при каждом вызове функции *BioAPI_BSPAttach*. Вызов функции *BioAPI_BSPAttach* определяет ПБУ с помощью идентификатора BSPUuid.

Параметры: BSPUuid (входной параметр) – BioAPI_UUID вызванного модуля ПБУ.

Подпункт 11.2.3

Ошибки: BioAPIERR_INVALID_UUID

Ссылки: 9.3.1.3 и 11.2.3.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ с предоставлением недостоверного УУИД.

3) Проверить возвращаемое значение вызова функции.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_BSPAttach* возвращает значение BioAPIERR_INVALID_UUID.

Пакет языка утверждения

```
<package name="049cc170-0c5f-1085-981f-0002a5d5fd2e">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение
        "BioSPI_BSPAttach_InvalidUUID" (см. далее элемент
        <description> утверждения).
    </description>

    <assertion name="BioSPI_BSPAttach_InvalidUUID"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью определения
            возвращения значения BioAPIERR_INVALID_UUID при вызове
            функции BioSPI_BSPAttach с недостоверным значением
            входного параметра УУИД.
```

```
Следующий текст соответствует приведенному в 9.3.1.3
и 11.2.3 спецификации БиоАПИ 2.0.
-------------------------------------------------
BioAPI_RETURN BioAPI BioSPI_BSPAttach
     (const BioAPI_UUID *BSPUuid,
     BioAPI_VERSION Version,
     const BioAPI_UNIT_LIST_ELEMENT *UnitList,
     uint32_t NumUnits,
     BioAPI_HANDLE BSPHandle).

     Данная функция вызывается инфраструктурой при
     каждом вызове функции BioAPI_BSPAttach. Вызов
     функции BioAPI_BSPAttach определяет ПБУ
     с помощью идентификатора BSPUuid.
     ---------------------------------------------
     Подпункт 9.3.1.3:
     Параметры: BSPUuid (входной параметр) -
     BioAPI_UUID вызванного модуля ПБУ.
     Пункт 11.2.3:
     Ошибки: BioAPIERR_INVALID_UUID
     ---------------------------------------------
     Порядок действий при испытании на соответствие:

     1) Загрузить испытуемый ПБУ.
     2) Присоединить испытуемый ПБУ
     с предоставлением недостоверного УУИД.
     3) Проверить возвращаемое значение вызова
     функции.

     Если какие-либо промежуточные операции прошли
     неуспешно, то выдается заключение
     о соответствии "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>
```

```
        <!-- Номер версии испытуемого ПБУ. -->
        <input name="_bspVersion"/>

        <!-- Время ожидания для события Timeout для
        BioAPI_NOTIFY_INSERT. -->
        <input name="_inserttimeout"/>

        <!-- Дескриптор модуля присоединенного ПБУ. -->
        <input name="_bsphandle"/>

        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными
        значениями параметров утверждения. -->
        <invoke activity="LoadAndAttach" >
             <input name="bspUuid" var="_bspUuid"/>
             <input name="bspVersion" var="_bspVersion"/>
             <input name="unitIDOrNull" value="0"/>
             <input name="bspHandle" var="_bsphandle"/>
             <input name="eventtimeouttime"
             var="_inserttimeout"/>
        </invoke>
        <!-- Процесс связан с функцией интерфейса обратного
        вызова инфраструктуры, предоставленной испытуемым
        компонентом ПБУ. При каждом входящем вызове функции
        связанный с ней процесс будет вызываться
        автоматически. -->

        <bindactivity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
</assertion>

<activity name="LoadAndAttach">
     <input name="bspUuid"/>
     <input name="bspVersion"/>
```

```
     <input name="unitIDOrNull"/>
     <input name="bspHandle"/>
     <input name="eventtimeouttime"/>

<!-- Инициализировать глобальную переменную
"_unitIDOrNull" с целью сделать ее доступной для события
"EventHandler". -->
<set name="_unitIDOrNull" var="unitIDOrNull"/>

<!-- Инициализировать глобальную переменную "_insert" с
помощью значениея "false". Событие "EventHandler"
присваивает данной переменной значение "true" после
получения уведомления о событии BioAPI_NOTIFY_INSERT и
значение "false" после получения уведомления о событии
BioAPI_NOTIFY_REMOVE. -->

<set name="_insert" value="false"/>
<set name="eventtimeoutflag" value="false"/>

<!-- Вызов функции BioSPI_BSPLoad. -->
<invoke function="BioSPI_BSPLoad">
     <input name="BSPUuid" var="bspUuid"/>
     <input name="BioAPINotifyCallback" value="*"/>
     <input name="BFPEnumerationHandler" value="*"/>
     <input name="MemoryFreeHandler" value="*" />
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и
выполнение процесса прерывается, в противном случае
выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
```

```
            Функция BioSPI_BSPLoad возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Уведомление о событии BioAPI_NOTIFY_INSERT не должно
    быть получено в течение указанной максимальной
    продолжительности.-->
    <wait_until timeout_var="eventtimeouttime"
        setvar="eventtimeoutflag" var="_insert"/>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и
    выполнение процесса прерывается, в противном случае
    выдается заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Уведомление о событии BioAPI_NOTIFY_INSERT
            получено в пределах указанной максимальной
            продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
    </assert_condition>

    <!-- Вызов функции BioSPI_BSPAttach с недостоверным УУИД. -->
    <invoke function="BioSPI_BSPAttach">
        <input name="BSPUuid"
            value="00000000-0000-0000-0000-000000000000"/>
        <input name="Version" var="bspVersion"/>
        <input name="Unit_1_UnitCategory"
        var="_unitCategory" />
        <input name="Unit_1_UnitID" var="_unitID"/>
```

```
    <input name="NumUnits" value="1" />
    <input name="BSPHandle" var="bspHandle"/>
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
    <description>
        Функция BioSPI_BSPAttach возвращает значение
        BioAPIERR_INVALID_UUID.
    </description>
    <equal_to var1="return"
        var2="__BioAPIERR_BSP_INVALID_UUID"/>
</assert_condition>

<!-- Вызов функции BioSPI_BSPUnload. -->
<invoke function="BioSPI_BSPUnload" >
    <input name="BSPUuid" var="bspUuid" />
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED", в
противном случае выдается заключение о соответствии
"PASS". -->
<assert_condition response_if_false="undecided"
        break_if_false="true">
    <description>
        Функция BioSPI_BSPUnload возвращает значение
        BioAPI_OK.
    </description>
    <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
```

```
    </activity>
</package>
```

8.11 Утверждение 3c *BioSPI_BSPAttach_InvalidVersion*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPIERR_INCOMPATIBLE_VERSION при вызове функции *BioSPI_BSPAttach* с несовместимым значением версии.

Выдержки

Подпункт 9.3.1.3

BioAPI_RETURN BioAPI BioSPI_BSPAttach

*(const BioAPI_UUID *BSPUuid,*

BioAPI_VERSION Version,

*const BioAPI_UNIT_LIST_ELEMENT *UnitList,*

uint32_tNumUnits,

BioAPI_HANDLEBSPHandle).

ПБУ должен подтвердить соответствие номеру версии, определенному значением параметра Version. Если версии несовместимы, то выполнение функции завершится неудачно.

Параметры: Version (входной параметр) – основной и дополнительный номера версии спецификации БиоАПИ, который по требованию приложения должен поддерживаться ПБУ. ПБУ должен определить, совместим ли он с требуемой версией.

Подпункт 11.2.3

Ошибки: BioAPIERR_INCOMPATIBLE_VERSION

Ссылки: 9.3.1.3 и 11.2.3.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ с недействительным номером версии.

3) Проверить возвращаемое значение вызова функции.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_BSPAttach* возвращает значение BioAPIERR_INCOMPATIBLE_VERSION.

Пакет языка утверждения

```
<package name="0052ac10-0c60-1085-9883-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_BSPAttach_InvalidVersion" (см. далее элемент
          <description> утверждения).
     </description>

     <assertion name="BioSPI_BSPAttach_InvalidVersion"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               определения возвращения значения BioAPIERR_INCOMPATIBLE_
               VERSION при вызове функции BioSPI_BSPAttach
               с несовместимым значением версии.
               Следующий текст соттветствует приведенному в 9.3.1.3
               и 11.2.3 спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioSPI_BSPAttach
                    (const BioAPI_UUID *BSPUuid,
                    BioAPI_VERSION Version,
                    const BioAPI_UNIT_LIST_ELEMENT *UnitList,
                    uint32_tNumUnits,
                    BioAPI_HANDLEBSPHandle).
               ПБУ должен подтвердить соответствие номеру версии,
               определенному значением параметра Version. Если
```

```
        версии несовместимы, то выполнение функции завершится
        неуспешно.
        ------------------------------------------------------
        Подпункт 9.3.1.3:
        Параметры: Version (входной параметр) - основной и
        дополнительный номера версии спецификации БиоАПИ,
        который по требованию приложения должен
        поддерживаться ПБУ. ПБУ должен определить, совместим
        ли он с требуемой версией.
        Пункт 11.2.3
        Ошибки: BioAPIERR_INCOMPATIBLE_VERSION
        ------------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ с недействительным
        номером версии.
        3) Проверить возвращаемое значение вызова функции.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Дескриптор модуля присоединенного ПБУ. -->
<input name="_bsphandle"/>
```

```
        <!-- Вызов основного процесса данного утверждения
        со значениями входных параметров, определенными
        значениями параметров утверждения. -->
        <invoke activity="LoadAndAttach" >
            <input name="bspUuid" var="_bspUuid"/>
            <input name="unitIDOrNull" value="0"/>
            <input name="bspHandle" var="_bsphandle"/>
            <input name="eventtimeouttime"
            var="_inserttimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного
        вызова инфраструктуры, предоставленной испытуемым
        компонентом ПБУ. При каждом входящем вызове функции
        связанный с ней процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
</assertion>

<activity name="LoadAndAttach">
        <input name="bspUuid"/>
        <input name="unitIDOrNull"/>
        <input name="bspHandle"/>
        <input name="eventtimeouttime"/>

        <!-- Инициализация глобальной переменной "_unitIDOrNull" с
        целью сделать ее доступной для процесса "EventHandler". -->
        <set name="_unitIDOrNull" var="unitIDOrNull"/>

        <!-- Инициализация глобальной переменной "_insert"
        значением "false". Событие "EventHandler" присваивает
        данной переменной значение "true" после получения
        уведомления о событии BioAPI_NOTIFY_INSERT и значение
        "false" после получения уведомления о событии BioAPI_
        NOTIFY_REMOVE. -->
        <set name="_insert" value="false"/>
```

```
<set name="eventtimeoutflag" value="false"/>

<!-- Вызов функции BioSPI_BSPLoad. -->
<invoke function="BioSPI_BSPLoad">
     <input name="BSPUuid" var="bspUuid"/>
     <input name="BioAPINotifyCallback" value="*"/>
     <input name="BFPEnumerationHandler" value="*"/>
     <input name="MemoryFreeHandler" value="*" />
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и
выполнение процесса прерывается, в противном случае
выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_BSPLoad возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Уведомление о событии BioAPI_NOTIFY_INSERT должно
быть получено в течение указанной максимальной
продолжительности.-->
<wait_until timeout_var="eventtimeouttime"
     setvar="eventtimeoutflag" var="_insert"/>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и
```

```
выполнение процесса прерывается, в противном случае
выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Уведомление о событии BioAPI_NOTIFY_INSERT
          получено в пределах указанной максимальной
          продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- Вызов функции BioSPI_BSPAttach с недействительным
номером версии. -->
<invoke function="BioSPI_BSPAttach">
     <input name="BSPUuid" var="bspUuid"/>
     <input name="Version" value="1"/>
     <input name="Unit_1_UnitCategory"
     var="_unitCategory" />
     <input name="Unit_1_UnitID" var="_unitID"/>
     <input name="NumUnits" value="1" />
     <input name="BSPHandle" var="bspHandle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
     <description>
          Функция BioSPI_BSPAttach возвращает значение
          BioAPIERR_INCOMPATIBLE_VERSION.
     </description>
     <equal_to var1="return"
          var2="__BioAPIERR_BSP_INCOMPATIBLE_VERSION"/>
```

```
        </assert_condition>

        <!-- Вызов функции BioSPI_BSPUnload. -->
        <invoke function="BioSPI_BSPUnload" >
             <input name="BSPUuid" var="bspUuid" />
             <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED", в
        противном случае выдается заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
                  break_if_false="true">
             <description>
                  Функция BioSPI_BSPUnload возвращает значение
                  BioAPI_OK.
             </description>
             <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
    </activity>
</package>
```

8.12 Утверждение 3d *BioSPI_BSPAttach_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью определения возвращения сообщения об ошибке при вызове функции *SPI_BSPAttach* с недействительным дескриптором ПБУ.

Выдержки

Подпункт 9.3.1.3

BioAPI_RETURN BioAPI BioSPI_BSPAttach

*(const BioAPI_UUID *BSPUuid,*

BioAPI_VERSION Version,

*const BioAPI_UNIT_LIST_ELEMENT *UnitList,*

uint32_t NumUnits,

BioAPI_HANDLE BSPHandle).

Данная функция вызывается инфраструктурой при каждом вызове функции *BioAPI_BSPAttach*. Вызов функции *BioAPI_BSPAttach* определяет ПБУ с помощью идентификатора BSPUuid.

Параметр BSPHandle (входной параметр) – значение BioAPI_HANDLE, присвоенное инфраструктурой и связанное с сессией присоединения, создаваемой данной функцией.

Ссылки: 9.3.1.3.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ с помощью недействительного дескриптора ПБУ.

3) Проверить возвращаемое значение вызова функции.

4) Выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_BSPAttach* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="03826830-0c57-1085-bfb0-0002a5d5fd2e">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение
        "BioSPI_BSPAttach_InvalidModuleHandle" (см. далее элемент
        <description> утверждения).
    </description>
```

```
<assertion name="BioSPI_BSPAttach_InvalidBSPHandle"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью
          определения возвращения сообщения об ошибке при
          вызове функции SPI_BSPAttach с недействительным
          дескриптором ПБУ.
          Следующий текст соответствует приведенному в 9.3.1.3
          спецификации БиоАПИ 2.0.
          ------------------------------------------------
          BioAPI_RETURN BioAPI BioSPI_BSPAttach
               (const BioAPI_UUID *BSPUuid,
               BioAPI_VERSION Version,
               const BioAPI_UNIT_LIST_ELEMENT *UnitList,
               uint32_t NumUnits,
               BioAPI_HANDLE BSPHandle).

          Данная функция вызывается инфраструктурой при каждом
          вызове функции BioAPI_BSPAttach. Вызов функции
          BioAPI_BSPAttach определяет ПБУ с помощью
          идентификатора BSPUuid.
          ------------------------------------------------
          Подпункт 9.3.1.3.2:
          Параметры: BSPHandle (входной параметр) - значение
          BioAPI_HANDLE, присвоенное инфраструктурой и
          связанное с сессией присоединения, создаваемой
          данной функцией.
          ------------------------------------------------
          Примечание - В настоящее время в описании функции
          BioAPI_BSPAttach содержится только указание
          на необходимость наличия специального значения,
          соответствующего недействительному значению
          дескриптора ПБУ (см. 8.1.7.2):
```

```
      NewBSPHandle (выходной параметр) - новый дескриптор,
      который может использоваться для взаимодействия с
      запрашиваемым ПБУ. Если выполнение функции
      заканчивается ошибкой, будет установлено значение
      BioAPIERR_FRAMEWORK_INVALID_BSP_HANDLE.
      ---------------------------------------------------
      Название BioAPIERR_FRAMEWORK_INVALID_BSP_HANDLE
      означает ошибку БиоАПИ и не может быть использовано
      в качестве дескриптора.
      Кроме того, значение BioAPIERR_FRAMEWORK_INVALID_
      BSP_HANDLE не определено.
      Настоящее утверждение использует 0 в качестве
      недействительного дескриптора ПБУ.
      Предполагается, что в поправках к настоящему
      стандарту данная проблема будет решена.
      ---------------------------------------------------
      Порядок действий при испытании на соответствие:

      1) Загрузить испытуемый ПБУ.
      2) Присоединить испытуемый ПБУ с помощью
      недостоверного дескриптора ПБУ.
      3) Проверить возвращаемое значение вызова функции.
      4) Выгрузить испытуемый ПБУ.

      Если какие-либо промежуточные операции прошли
      неуспешно, то выдается заключение о соответствии
      "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Номер версии испытуемого ПБУ. -->
<input name="_bspVersion"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
```

```
        <input name="_inserttimeout"/>

        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="LoadAndAttach" >
             <input name="bspUuid" var="_bspUuid"/>
             <input name="bspVersion" var="_bspVersion"/>
             <input name="unitIDOrNull" value="0"/>
             <input name="eventtimeouttime"
             var="_inserttimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->

        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="LoadAndAttach">
        <input name="bspUuid"/>
        <input name="bspVersion"/>
        <input name="unitIDOrNull"/>
        <input name="eventtimeouttime"/>

        <!-- Инициализация глобальной переменной "_unitIDOrNull" с
        целью сделать ее доступной для события "EventHandler". -->
        <set name="_unitIDOrNull" var="unitIDOrNull"/>

        <!-- Присвоение глобальной переменной "_insert" значения
        "false". Событие "EventHandler" присваивает данной
```

```
переменной значение "true", после получения уведомления о
событии BioAPI_NOTIFY_INSERT и значение "false" после
получения уведомления о событии BioAPI_NOTIFY_REMOVE. -->
<set name="_insert" value="false"/>

<set name="eventtimeoutflag" value="false"/>

<!-- Вызов функции BioSPI_BSPLoad. -->
<invoke function="BioSPI_BSPLoad">
     <input name="BSPUuid" var="bspUuid"/>
     <input name="BioAPINotifyCallback" value="*"/>
     <input name="BFPEnumerationHandler" value="*"/>
     <input name="MemoryFreeHandler" value="*" />
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и
выполнение процесса прерывается, в противном случае
выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_BSPLoad возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Уведомление о событии BioAPI_NOTIFY_INSERT должно
быть получено в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="eventtimeouttime"
     setvar="eventtimeoutflag" var="_insert"/>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и
выполнение процесса прерывается, в противном случае
выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Уведомление о событии BioAPI_NOTIFY_INSERT
          получено в течение указанной максимальной
          продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- Вызов функции BioSPI_BSPAttach с недействительным
значением входного парметра "BSPHandle". -->
<invoke function="BioSPI_BSPAttach">
     <input name="BSPUuid" var="bspUuid"/>
     <input name="Version" var="bspVersion"/>
     <input name="Unit_1_UnitCategory"
     var="_unitCategory" />
     <input name="Unit_1_UnitID" var="_unitID"/>
     <input name="NumUnits" value="1" />
     <input name="BSPHandle" value="0"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
     <description>
```

```
                Функция BioSPI_BSPAttach возвращает значение
                BioAPIERR_INVALID_BSP_HANDLE.
            </description>
            <equal_to var1="return"
                var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
        </assert_condition>

        <!-- Вызов функции BioSPI_BSPUnload. -->
        <invoke function="BioSPI_BSPUnload" >
            <input name="BSPUuid" var="bspUuid" />
            <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
                break_if_false="true">
            <description>
                Функция BioSPI_BSPUnload возвращает значение
                BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
    </activity>
</package>
```

8.13 Утверждение 4a *BioSPI_BSPDetach_ValidParam*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_BSPDetach* с достоверным дескриптором модуля.

Выдержки

Подпункт 9.3.1.4

BioAPI_RETURN BioAPI BioSPI_BSPDetach

(BioAPI_HANDLE BSPHandle).

Данная функция вызывается БиоАПИ при каждом вызове функции *BioAPI_BSPDetach*, устанавливающей сессию присоединения, определенную значением параметра BSPHandle. ПБУ должен выполнить все операции по очистке памяти, связанные с указанным дескриптором присоединения.

Раздел А.4

Данная функция должна поддерживаться всеми типами ПБУ.

Ссылки: 9.3.1.4 и А.4.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Отсоединить испытуемый ПБУ с помощью дескриптора модуля.

4) Проверить возвращаемое значение.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_BSPDetach* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="00e0d2b0-0c7a-1085-b8ac-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_BSPDetach_ValidParam" (см. далее элемент
          <description> утверждения).
     </description>
```

```
<assertion name="BioSPI_BSPDetach_ValidParam"
model="BSPTesting">
     <description>
          Проверку данного утверждения провводят с целью
          определения возвращения значения BioAPI_OK при
          вызове функции BioSPI_BSPDetach с достоверным
          дескриптором модуля.
          Следующий текст соответствует приведенному в 9.3.1.4
          спецификации БиоАПИ 2.0.
          ---------------------------------------------------
          BioAPI_RETURN BioAPI BioSPI_BSPDetach
               (BioAPI_HANDLEBSPHandle).
          Данная функция вызывается БиоАПИ при каждом вызове
          функции BioAPI_BSPDetach, устанавливающей сессию
          присоединения, определенную значением параметра
          BSPHandle. ПБУ должен выполнить все операции
          по очистке памяти, связанные с указанным
          дескриптором присоединения.
          ---------------------------------------------------
          Раздел А.4
          Данная функция должна поддерживаться всеми типами
          ПБУ.
          ---------------------------------------------------
          Порядок действий при испытании на соответствие:

          1) Загрузить испытуемый ПБУ.
          2) Присоединить испытуемый ПБУ.
          3) Отсоединить испытуемый ПБУ с помощью дескриптора
          модуля.
          4) Проверить возвращаемое значение.

          Если какие-либо промежуточные операции прошли
          неуспешно, то выдается заключение о соответствии
          "UNDECIDED".
     </description>
```

```
        <!-- УУИД испытуемого ПБУ.-->
        <input name="_bspUuid"/>

        <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
        <input name="_inserttimeout"/>

        <!-- Модуль дескриптора для ПБУ. -->
        <input name="_bsphandle"/>

        <!-- Вызов основного процесса данного утверждения
        со значениями входных параметров, определенными
        значениями параметров утверждения. -->
        <invoke activity="LoadAndAttachAndDetach" >
             <input name="bspUuid" var="_bspUuid"/>
             <input name="unitIDOrNull" value="0"/>
             <input name="bspHandle" var="_bsphandle"/>
             <input name="eventtimeouttime"
             var="_inserttimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="LoadAndAttachAndDetach">
        <input name="bspUuid"/>
        <input name="unitIDOrNull"/>
        <input name="bspHandle"/>
        <input name="eventtimeouttime"/>
```

```
<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" var="unitIDOrNull"/>
     <input name="bspHandle" var="bspHandle"/>
     <input name="eventtimeouttime"
     var="eventtimeouttime"/>
</invoke>

<!-- Вызов функции BioSPI_BSPDetach. -->
<invoke function="BioSPI_BSPDetach">
     <input name="BSPHandle" var="bspHandle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
     <description>
          Функция BioSPI_BSPDetach возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_BSPUnload. -->
<invoke function="BioSPI_BSPUnload" >
     <input name="BSPUuid" var="bspUuid"/>
     <return setvar="return"/>
```

```
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED" и
        выполнение процесса прерывается, в противном случае
        выдается заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
                break_if_false="true">
            <description>
                Функция BioSPI_BSPUnload возвращает значение
                BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
    </activity>
</package>
```

8.14 Утверждение 4b *BioSPI_BSPDetach_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью определения возвращения сообщения об ошибке при вызове функции *BioSPI_BSPDetach* с недостоверным дескриптором модуля.

Выдержки

Подпункт 9.3.1.4

BioAPI_RETURN BioAPI BioSPI_BSPDetach

(BioAPI_HANDLE BSPHandle).

Данная функция вызывается БиоАПИ при каждом вызове функции *BioAPI_BSPDetach*, устанавливающей сессию присоединения, определенную значением параметра BSPHandle.

BSPHandle (входной параметр) – дескриптор, относящийся к сессии присоединения, завершаемой данной функцией.

Подпункт 8.1.8.1

Данную функцию следует вызывать только после вызова функции *BioAPI_BSPAttach* не более одного раза для одного и того же дескриптора ПБУ, созданного путем вызова функции *BioAPI_BSPAttach*.

Ссылки: 9.3.1.4 и 8.1.8.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Отсоединить испытуемый ПБУ с помощью недействительного дескриптора модуля.

4) Проверить возвращаемое значение вызова функции.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_BSPDetach* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="0434c458-0c79-1085-9f2c-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_BSPDetach_
          InvalidModuleHandle" (см. далее элемент <description>
          утверждения).
     </description>

     <assertion name="BioSPI_BSPDetach_InvalidBSPHandle"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               определения возвращения сообщения об ошибке
```

при вызове функции BioSPI_BSPDetach с недостоверным дескриптором модуля.
Следующий текст соответствует приведенному в 9.3.1.4 и 8.1.8.1 спецификации БиоАПИ 2.0.

BioAPI_RETURN BioAPI BioSPI_BSPDetach
(BioAPI_HANDLE BSPHandle).

Данная функция вызывается БиоАПИ при каждом вызове функции BioAPI_BSPDetach, устанавливающей сессию присоединения, определенную значением параметра BSPHandle.

Подпункт 8.1.8.1:
Данную функцию следует вызывать только после вызова функции BioAPI_BSPAttach не более одного раза для одного и того же дескриптора ПБУ, созданного с помощью вызова функции BioAPI_BSPAttach.

Подпункт 9.3.1.4:
BSPHandle (входной параметр) - дескриптор, относящийся к сессии присоединения, завершаемой данной функцией.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Отсоединить испытуемый ПБУ с помощью недостоверного дескриптора модуля.
4) Проверить возвращаемое значение вызова функции.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

```
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>

    <!-- Модуль дескриптора для присоединенного ПБУ. -->
    <input name="_bsphandle"/>

    <!-- Вызов основного процесса данного утверждения
    со значениями входных параметров, определенными
    значениями параметров утверждения. -->
    <invoke activity="LoadAndAttachAndDetach" >
        <input name="bspUuid" var="_bspUuid"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bsphandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="_inserttimeout"/>
    </invoke>

    <!-- Процесс связан с функцией интерфейса обратного вызова
    инфраструктуры, предоставленной испытуемым компонентом
    ПБУ. При каждом входящем вызове функции связанный с ней
    процесс будет вызываться автоматически. -->
    <bind activity="EventHandler"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        function="BioSPI_EventHandler"/>
</assertion>

<activity name="LoadAndAttachAndDetach">
    <input name="bspUuid"/>
    <input name="unitIDOrNull"/>
    <input name="bsphandle"/>
    <input name="eventtimeouttime"/>
```

```
<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" var="unitIDOrNull"/>
     <input name="bspHandle" var="bsphandle"/>
     <input name="eventtimeouttime"
     var="eventtimeouttime"/>
</invoke>

<!-- Вызов функции BioSPI_BSPDetach с недействительным
значением дескриптора модуля. -->
<invoke function="BioSPI_BSPDetach">
     <input name="BSPHandle" value="0"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
     <description>
          Функция BioSPI_BSPDetach возвращает значение
          BioAPIERR_INVALID_BSP_HANDLE.
     </description>
     <equal_to var1="return"
          var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
</assert_condition>

<!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
```

```
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="bsphandle" />
        </invoke>
    </activity>
</package>
```

8.15 Утверждение 4с *BioSPI_BSPDetach_Confirm*

Описание: Проверку данного утверждения проводят с целью определения завершения сессии присоединения при вызове функции *BioSPI_BSPDetach*.

Выдержки

Подпункт 9.3.1.4

BioAPI_RETURN BioAPI BioSPI_BSPDetach

(BioAPI_HANDLE BSPHandle).

Данная функция вызывается БиоАПИ при каждом вызове функции *BioAPI_BSPDetach*, устанавливающей сессию присоединения, определенную значением параметра BSPHandle. ПБУ должен выполнить все операции по очистке памяти, связанные с указанным дескриптором присоединения.

Подпункт 8.1.8.4

Ошибки: BioAPIERR_INVALID_BSP_HANDLE

Ссылки: 9.3.1.4 и 8.1.8.4.

Порядок действий:

1) Загрузить испытуемый ПБУ

2) Присоединить испытуемый ПБУ

3) Отсоединить испытуемый ПБУ с помощью действительного дескриптора модуля.

4) Проверить возвращаемое значение вызова функции.

5) Вызвать функцию *BioSPI_Enroll*. Предполагается завершение функции с ошибкой BioAPIERR_INVALID_BSP_HANDLE.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_Enroll* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="002e7e58-0c78-1085-9e1d-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_BSPDetach_
          Confirm" (см. далее элемент <description> утверждения).
     </description>

     <assertion name="BioSPI_BSPDetach_Confirm" model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               определения завершения сессии присоединения
               при вызове функции BioSPI_BSPDetach.
               Следующий текст соответствует приведенному в 9.3.1.4
               спецификации БиоАПИ 2.0.
               ---------------------------------------------------
               BioAPI_RETURN BioAPI BioSPI_BSPDetach
                    (BioAPI_HANDLE BSPHandle).
               Данная функция вызывается БиоАПИ при каждом вызове
               функции BioAPI_BSPDetach, устанавливающей сессию
               присоединения, определенную значением параметра
               BSPHandle. ПБУ должен выполнить все операции очистки
               памяти, связанные с указанным дескриптором присоеди-
               нения.
               ---------------------------------------------------
```

```
     Подпункт 8.1.8.4
          Ошибки: BioAPIERR_INVALID_BSP_HANDLE.
     ------------------------------------------------
     Порядок действий при испытании на соответствие:

     1) Загрузить испытуемый ПБУ.
     2) Присоединить испытуемый ПБУ.
     3) Отсоединить испытуемый ПБУ с помощью действительного
     дескриптора модуля.
     4) Проверить возвращаемое значение вызова функции.
     5) Вызвать функцию BioSPI_Enroll. Предполагается
     завершение функции с ошибкой BioAPIERR_INVALID_
     BSP_HANDLE.

     Если какие-либо промежуточные операции прошли
     неуспешно, то выдается заключение о соответствии
     "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Дескриптор модуля для присоединенного ПБУ. -->
<input name="_bsphandle"/>

<!-- Вызов основного процесса данного утверждения
со значениями входных параметров, определенными
значениями параметров утверждения. -->
<invoke activity="LoadAndAttachAndDetach" >
     <input name="bspUuid" var="_bspUuid"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
```

```
            <input name="eventtimeouttime"
            var="_inserttimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного
        вызова инфраструктуры, предоставленной испытуемым
        компонентом ПБУ. При каждом входящем вызове функции
        связанный с ней процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="LoadAndAttachAndDetach">
        <input name="bspUuid"/>
        <input name="unitIDOrNull"/>
        <input name="bspHandle"/>
        <input name="eventtimeouttime"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
            <input name="bspVersion" value="32"/>
            <input name="unitIDOrNull" var="unitIDOrNull"/>
            <input name="bspHandle" var="bspHandle"/>
            <input name="eventtimeouttime"
            var="eventtimeouttime"/>
        </invoke>

        <!-- Вызов функции BioSPI_BSPDetach. -->
        <invoke function="BioSPI_BSPDetach">
            <input name="BSPHandle" var="bspHandle"/>
```

```
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition break_if_false="true">
    <description>
        Функция BioSPI_BSPDetach возвращает значение
        BioAPI_OK.
    </description>
    <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_Enroll. Предполагается, что
данная функция завершится с ошибкой, что подтверждает
отсоединение ПБУ.-->
<invoke function="BioSPI_Enroll">
    <input name="BSPHandle" var="bspHandle"/>
    <input name="Purpose"
    var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
    <input name="Subtype" value="0"/>
    <input name="Timeout" value="15000"/>
    <output name="NewTemplate" set-
    var="newtemplate_handle"/>
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
    <description>
```

```
                    Функция BioSPI_Enroll возвращает значение
                    BioAPIERR_INVALID_BSP_HANDLE.
               </description>
               <equal_to var1="return"
               var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
          </assert_condition>

          <!-- Вызов функции BioSPI_BSPUnload. -->
          <invoke function="BioSPI_BSPUnload" >
               <input name="BSPUuid" var="bspUuid"/>
               <return setvar="return"/>
          </invoke>

          <!-- Выдача заключения о соответствии. Если условие,
          определенное далее в элементе <description>, ложное,
          выдается заключение о соответствии "UNDECIDED", в противном
          случае выдается заключение о соответствии "PASS". -->
          <assert_condition response_if_false="undecided">
               <description>
                    Функция BioSPI_BSPUnload возвращает значение
                    BioAPI_OK.
               </description>
               <equal_to var1="return" var2="__BioAPI_OK"/>
          </assert_condition>
     </activity>
</package>
```

8.16 Утверждение 5a *BioSPI_FreeBIRHandle_ValidParam*

Описание: Проверку данного утверждения проводят с целью определения освобождения дескриптора ЗБИ при вызове функции *BioSPI_FreeBIRHandle* с достоверным дескриптором ЗБИ.

Выдержки

Подпункт 9.3.2.1

BioAPI_RETURN BioAPI BioSPI_FreeBIRHandle

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_HANDLE Handle).

Пункт 8.2.1

Данная функция освобождает память и ресурсы, связанные с установленным дескриптором ЗБИ. Ссылка на данную ЗБИ с помощью данного дескриптора становится невозможной.

Раздел А.4

Данная функция должна поддерживаться всеми типами ПБУ.

Ссылки: 9.3.2.1, 8.2.1 и А.4.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Зарегистрировать ЗБИ для получения дескриптора ЗБИ.

4) Вызвать функцию *BioSPI_FreeBIRHandle* для освобождения дескриптора ЗБИ.

5) Вызвать функцию *BioSPI_GetBIRFromHandle*, которая предположительно должна завершиться с ошибкой BioAPIERR_INVALID_BIR_HANDLE.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_FreeBIRHandle* возвращает значение BioAPI_OK, а следующий вызов функции *BioSPI_GetBIRFromHandle* возвращает значение BioAPIERR_INVALID_BIR_HANDLE.

Пакет языка утверждения

```
<package name="0280a7d0-0c80-1085-a9a0-0002a5d5fd2e">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
```

```
        Данный пакет содержит утверждение "BioSPI_FreeBIRHandle_
        ValidParam " (см. далее элемент <description> утверждения).
    </description>

    <assertion name="BioSPI_FreeBIRHandle_ValidParam"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью
            определения освобождения дескриптора ЗБИ при вызове
            функции BioSPI_FreeBIRHandle с достоверным
            дескриптором ЗБИ.
            Следующий текст соответствует приведенному в А.4 и
            9.3.2.1, 8.2.1 спецификации БиоАПИ 2.0.
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_FreeBIRHandle
                 (BioAPI_HANDLE BSPHandle,
                 BioAPI_BIR_HANDLE Handle).
            Примечание - Подробное определение функции
            BioAPI_FreeBIRHandle приведено в 8.2.1.
            ------------------------------------------------
            Пункт 8.2.1:
            Данная функция освобождает память и ресурсы, связанные
            с установленным дескриптором ЗБИ. Ссылка на данную
            ЗБИ с помощью данного дескриптора становится
            невозможной.
            ------------------------------------------------
            Раздел А.4:
            Данная функция должна поддерживаться всеми типами ПБУ.
            ------------------------------------------------
            Порядок действий при испытании на соответствие:

            1) Загрузить испытуемый ПБУ.
            2) Присоединить испытуемый ПБУ.
```

```
     3) Зарегистрировать ЗБИ для получения дескриптора ЗБИ.
     4) Вызвать функцию BioSPI_FreeBIRHandle для
     освобождения дескриптора ЗБИ.
     5) Вызвать функцию BioSPI_GetBIRFromHandle,
     предположительно завершающуюся с ошибкой Bio-
     APIERR_INVALID_BIR_HANDLE.

     Если какие-либо промежуточные операции прошли
     неуспешно, то выдается заключение о соответствии
     "UNDECIDED".

</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на требование испытуемым ПБУ поддержки
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported"/>

<!-- Время ожидания для события
BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для BioSPI_Enroll. -->
<input name="_capturetimeout"/>

<!-- Вызов основного процесса данного утверждения
со значениями входных параметров, определенными
значениями параметров утверждения. -->
<invoke activity="BioSPI_FreeBIRHandle">
     <input name="bspUuid" var="_bspUuid"/>
```

```
            <input name="inserttimeouttime"
            var="_inserttimeout"/>
            <input name="nosourcepresentsupported"
            var="_noSourcePresentSupported"/>
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
        </invoke>
        <!-- Процесс связан с функцией интерфейса обратного
        вызова инфраструктуры, предоставленной испытуемым
        компонентом ПБУ. При каждом входящем вызове функции
        связанный с ней процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_FreeBIRHandle">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported"/>
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>

        <!-- Данное утверждение использует функцию BSPHandle со
        значением "1" для всех вызовов BioSPI, которые этого
        требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                break_on_break="true">
```

```
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о
событии BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать
получения уведомления о событиив пределах указанной
максимальной продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
              setvar="eventtimeoutflag">
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent"/>
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
              break_if_false="true">
        <description>
              Испытуемый ПБУ не поддерживает уведомление о
              событии BioAPI_NOTIFY_SOURCE_PRESENT или
              уведомление о событии получено в пределах
              указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
</assert_condition>
```

```
<!-- ПБУ готов к регистрации ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации ЗБИ. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
          var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_
          ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <output name="NewTemplate" set-
     var="newtemplate_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и
выполнение процесса прерывается, в противном случае
выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Enroll возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_FreeBIRHandle с передачей
дескриптора зарегистрированной ЗБИ. -->
<invoke function="BioSPI_FreeBIRHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var=" newtemplate_handle"/>
     <return setvar="return"/>
</invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition break_if_false="true">
     <description>
          Функция BioSPI_FreeBIRHandle возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetBIRFromHandle с передачей
дескриптора зарегистрированной ЗБИ с целью проверки
освобождения дескриптора. -->
<invoke function="BioSPI_GetBIRFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="newtemplate_handle"/>
     <output name="HeaderVersion"
     setvar="headerversion"/>
     <output name="ProcessedLevel"
     setvar="processedlevel"/>
     <output name="FormatOwner" setvar="formatowner"/>
     <output name="FormatType" setvar="formattype"/>
     <output name="Quality" setvar="quality"/>
     <output name="Purpose" setvar="purpose"/>
     <output name="ProductOwner" setvar="productowner" />
     <output name="ProductType" setvar="producttype" />
     <output name="Creation_Year" setvar="creationyear" />
     <output name="Creation_Month" setvar="creationmonth" />
     <output name="Creation_Day" setvar="creationday" />
     <output name="Creation_Hour" setvar="creationhour" />
     <output name="Creation_Minute"
     setvar="creationminute" />
```

```
        <output name="Creation_Second"
        setvar="creationsecond" />
        <output name="Expiration_Year"
        setvar="expirationyear" />
        <output name="Expiration_Month"
        setvar="expirationmonth" />
        <output name="Expiration_Day" setvar="expirationday" />
        <output name="SBFormatOwner"
        setvar="securityformatowner" />
        <output name="SBFormatType"
        setvar="securityformattype" />
        <output name="Index" setvar="index" />
        <output name="BiometricData" setvar="biometircdata" />
        <output name="SecurityBlock"
        setvar="securityblock"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED", в противном
    случае выдается заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided">
        <description>
            Функция BioSPI_GetHeaderFromHandle возвращает
            значение BioAPIERR_INVALID_BIR_HANDLE.
        </description>
        <equal_to var1="return"
        var2="__BioAPIERR_BSP_INVALID_BIR_HANDLE"/>
    </assert_condition>

    <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
    <invoke activity="DetachAndUnload"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
        <input name="bspUuid" var="bspUuid"/>
```

```
            <input name="BSPHandle" var="_bsphandle"/>
        </invoke>
    </activity>
</package>
```

8.17 Утверждение 5b *BioSPI_FreeBIRHandle_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью определения возвращения ошибки при вызове функции *BioSPI_FreeBIRHandle* с недействительным дескриптором ПБУ.

Выдержки

Подпункт 9.3.2.1

BioAPI_RETURN BioAPI BioSPI_FreeBIRHandle

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_HANDLE Handle).

Параметры: BSPHandle (входной параметр) – дескриптор присоединенного ПБУ.

Пункт 8.2.1

Данная функция освобождает память и ресурсы, относящиеся к определенному дескриптору ЗБИ.

Ссылки: 9.3.2.1 и 8.2.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Зарегистрировать ЗБИ для получения дескриптора ЗБИ.

4) Вызвать функцию *BioSPI_FreeBIRHandle* с недостоверным дескриптором ПБУ.

5) Проверить возвращаемое значение.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_FreeBIRHandle* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="047aed48-0c80-1085-898b-0002a5d5fd2e">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение "BioSPI_FreeBIRHandle_
        InvalidBSPHandle" (см. далее элемент <description>
        утверждения).
    </description>

    <assertion name="BioSPI_FreeBIRHandle_InvalidBSPHandle"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью опреде-
            ления возвращения ошибки при вызове функции BioSPI_
            FreeBIRHandle с недействительным дескриптором ПБУ.
            Следующий текст соответствует приведенному в 9.3.2.1
            и 8.2.1 спецификации БиоАПИ 2.0.
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_FreeBIRHandle
                (BioAPI_HANDLE BSPHandle,
                BioAPI_BIR_HANDLE Handle).
            Примечание - Подробное определение функции
            BioAPI_FreeBIRHandle приведено в 8.2.1.
            ------------------------------------------------
            Пункт 8.2.1:
            Данная функция освобождает память и ресурсы,
            относящиеся к определенному дескриптору ЗБИ.
            ------------------------------------------------
            Подпункт 9.3.2.1:
            Параметры: BSPHandle (входной параметр) - дескриптор
            присоединенного ПБУ.
```

```
     ------------------------------------------------
     Порядок действий при испытании на соответствие:

     1) Загрузить испытуемый ПБУ.
     2) Присоединить испытуемый ПБУ.
     3) Зарегистрировать ЗБИ для получения дескриптора ЗБИ.
     4) Вызвать функцию BioSPI_FreeBIRHandle с недостоверным
     дескриптором ПБУ.
     5) Проверить возвращаемый код.
     Если какие-либо промежуточные операции прошли
     неуспешно, то выдается заключение о соответствии
     "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на поддержку испытуемым ПБУ уведомления о
событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Enroll. -->
<input name="_capturetimeout"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_FreeBIRHandle">
     <input name="bspUuid" var="_bspUuid"/>
```

```
            <input name="inserttimeouttime"
            var="_inserttimeout"/>
            <input name="nosourcepresentsupported"
            var="_noSourcePresentSupported" />
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_FreeBIRHandle">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>

        <!-- Данное утверждение использует функцию BSPHandle со
        значением "1" для всех вызовов BioSPI, которые этого
        требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
```

```
            break_on_break="true">
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>
<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
            setvar="eventtimeoutflag">
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается заключение
о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Испытуемый ПБУ не поддерживает уведомление о
            событии BioAPI_NOTIFY_SOURCE_PRESENT или
            уведомление о событии получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
</assert_condition>
```

```
<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <output name="NewTemplate" set-
     var="newtemplate_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Enroll возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_FreeBIRHandle с недействительным
дескриптором модуля. -->
<invoke function="BioSPI_FreeBIRHandle">
     <input name="BSPHandle" value="0"/>
     <input name="Handle" var="newtemplate_handle"/>
     <return setvar="return"/>
</invoke>
```

```
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Функция BioSPI_FreeBIRHandle возвращает значение
                BioAPIERR_INVALID_BSP_HANDLE.
            </description>
            <equal_to var1="return"
                var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.18 Утверждение 5c *BioSPI_FreeBIRHandle_InvalidBIRHandle*

Описание: Проверку данного утверждения проводят с целью определения возвращения ошибки при вызове функции *BioSPI_FreeBIRHandle* с недействительным дескриптором ЗБИ.

Выдержки

Подпункт 9.3.2.1

BioAPI_RETURN BioAPI BioSPI_FreeBIRHandle

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_HANDLE Handle).

Параметры: Дескриптор (входной параметр) – дескриптор ЗБИ, который должен быть освобожден.

Пункт 8.2.1

Данная функция освобождает память и ресурсы, относящиеся к определенному дескриптору ЗБИ.

Ошибки: BioAPIERR_INVALID_BIR_HANDLE

Ссылки: 9.3.2.1 и 8.2.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Зарегистрировать ЗБИ для получения дескриптора ЗБИ.

4) Вызвать функцию *BioSPI_FreeBIRHandle* для освобождения недействительного дескриптора ЗБИ.

5) Проверить возвращаемый код.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_FreeBIRHandle* возвращает значение BioAPIERR_INVALID_BIR_HANDLE.

Пакет языка утверждения

```
<package name="018e6c18-0c9c-1085-afdf-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_FreeBIRHandle_InvalidBIRHandle" (см. далее
          элемент <description> утверждения).
     </description>

     <assertion name="BioSPI_FreeBIRHandle_InvalidBIRHandle"
     model="BSPTesting">
          <description>
```

Проверку данного утверждения проводят с целью определения возвращения ошибки при вызове функции BioSPI_FreeBIRHandle с недействительным дескриптором ЗБИ.

Следующий текст соответствует приведенному в 9.3.2.1 и 8.2.1 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_FreeBIRHandle
     (BioAPI_HANDLE BSPHandle,
     BioAPI_BIR_HANDLE Handle).
```

Примечание – Подробное определение функции BioAPI_FreeBIRHandle приведено в 8.2.1.

Пункт 8.2.1:

Данная функция освобождает память и ресурсы, относящиеся к определенному дескриптору ЗБИ.

Ошибки: BioAPIERR_INVALID_BIR_HANDLE

Подпункт 9.3.2.1:

Параметры: Дескриптор (входной параметр) – дескриптор ЗБИ, который должен быть освобожден.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Зарегистрировать ЗБИ для получения дескриптора ЗБИ.
4) Вызвать функцию BioSPI_FreeBIRHandle для освобождения недействительного дескриптора ЗБИ.
5) Проверить возвращаемый код.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

```
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на поддержку испытуемым ПБУ уведомления о
событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события
BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Enroll. -->
<input name="_capturetimeout"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_FreeBIRHandle">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
</invoke>
<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
```

```
      ПБУ. При каждом входящем вызове функции связанный с ней
      процесс будет вызываться автоматически. -->
      <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_FreeBIRHandle">
      <input name="bspUuid"/>
      <input name="inserttimeouttime"/>
      <input name="nosourcepresentsupported" />
      <input name="sourcepresenttimeouttime"/>
      <input name="capturetimeouttime"/>

      <!-- Данное утверждение использует функцию BSPHandle со
      значением "1" для всех вызовов BioSPI, которые этого
      требуют. -->
      <set name="_bsphandle" value="1"/>

      <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
      предоставленных испытуемым ПБУ. -->
      <invoke activity="LoadAndAttach"
                  package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                  break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
            <input name="bspVersion" value="32"/>
            <input name="unitIDOrNull" value="0"/>
            <input name="bspHandle" var="_bsphandle"/>
            <input name="eventtimeouttime"
            var="inserttimeouttime"/>
      </invoke>

      <set name="eventtimeoutflag" value="false"/>

      <!-- Если испытуемый ПБУ поддерживает уведомление о событии
      BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
```

```
уведомления в пределах указанной максимальной
продолжительности.-->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию BioSPI_
Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <output name="NewTemplate" set-
     var="newtemplate_handle"/>
```

```
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
    <description>
          Функция BioSPI_Enroll возвращает значение
          BioAPI_OK.
    </description>
    <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_FreeBIRHandle с недействительным
дескриптором ЗБИ. -->
<invoke function="BioSPI_FreeBIRHandle">
    <input name="BSPHandle" var="_bsphandle"/>
    <input name="Handle" value="-1"/>
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
    <description>
          Функция BioSPI_FreeBIRHandle возвращает значение
          BioAPIERR_INVALID_BIR_HANDLE.
    </description>
    <equal_to var1="return"
    var2="__BioAPIERR_BSP_INVALID_BIR_HANDLE"/>
```

```
            </assert_condition>

            <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
            <invoke activity="DetachAndUnload"
                        package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
                <input name="bspUuid" var="bspUuid" />
                <input name="BSPHandle" var="_bsphandle" />
            </invoke>
        </activity>
</package>
```

8.19 Утверждение 6a *BioSPI_GetBIRFromHandle_ValidParam*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_GetBIRFromHandle* с достоверными параметрами.

Выдержки

Подпункт 9.3.2.2

BioAPI_RETURN BioAPI BioSPI_GetBIRFromHandle

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_HANDLE Handle,

*BioAPI_BIR *BIR).*

Пункт 8.2.2

Данная функция возвращает ЗБИ, связанную с дескриптором ЗБИ, возвращенным ПБУ.

Возвращаемое значение: Значение BioAPI_RETURN указывает на успешное выполнение функции или определяет тип ошибки. Значение BioAPI_OK соответствует успешному выполнению. Остальные значения определяют тип ошибки.

Дескриптор ЗБИ освобождается ПБУ до возврата из функции.

Раздел А.4

Данная функция должна поддерживаться всеми типами ПБУ.

Ссылки: 9.3.2.2, 8.2.2 и А.4.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Зарегистрировать ЗБИ для получения дескриптора ЗБИ.

4) Вызвать функцию *BioSPI_GetBIRFromHandle*. Предполагается, что функция возвратит значение BioAPI_OK.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_GetBIRFromHandle* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="0460b658-0cb4-1085-a304-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_ GetBIRFromHan-
          dle_ValidParam" (см. далее элемент
          <description> утверждения).
     </description>

     <assertion name="BioSPI_GetBIRFromHandle_ValidParam"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               определения возвращения значения BioAPI_OK при вызове
               функции BioSPI_GetBIRFromHandle с достоверными
               параметрами.
               Следующий текст соответствует приведенному в 9.3.2.2,
               8.2.2 и А.4 спецификации БиоАПИ 2.0.
               ------------------------------------------------
```

```
BioAPI_RETURN BioAPI BioSPI_GetBIRFromHandle
     (BioAPI_HANDLE BSPHandle,
     BioAPI_BIR_HANDLE Handle,
     BioAPI_BIR *BIR).

Примечание - Подробное определение функции
BioAPI_GetBIRFromHandle приведено в 8.2.2.
-------------------------------------------------
Пункт 8.2.2:
Данная функция возвращает ЗБИ, связанную с дескриптором
ЗБИ, возвращенным ПБУ.
Значение BioAPI_RETURN указывает на успешное
выполнение функции или определяет тип ошибки.
Значение BioAPI_OK соответствует успешному выполнению.
Остальные значения определяют тип ошибки.
Дескриптор ЗБИ освобождает ПБУ до возврата из функции.
-------------------------------------------------
Раздел А.4:
Данная функция должна поддерживаться всеми типами ПБУ.
-------------------------------------------------
Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Зарегистрировать ЗБИ для получения дескриптора ЗБИ.
4) Вызвать функцию BioSPI_GetBIRFromHandle. Предпо-
лагается, что функция возвратит значение BioAPI_OK.

Если какие-либо промежуточные операции прошли
неуспешно, то выдается заключение о соответствии
"UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
```

```
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на поддержку испытуемым ПБУ уведомления о
событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Enroll. -->
<input name="_capturetimeout"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_GetBIRFromHandle">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически. -->
<bind activity="EventHandler"
```

```
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_GetBIRFromHandle">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>

        <!-- Данное утверждение использует функцию BSPHandle со
        значением "1" для всех вызовов BioSPI, которые этого
        требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
            <input name="bspVersion" value="32"/>
            <input name="unitIDOrNull" value="0"/>
            <input name="bspHandle" var="_bsphandle"/>
            <input name="eventtimeouttime"
            var="inserttimeouttime"/>
        </invoke>

        <set name="eventtimeoutflag" value="false"/>

        <!-- Если испытуемый ПБУ поддерживает уведомление о событии
        BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
        уведомления в пределах указанной максимальной
        продолжительности. -->
```

```
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ поддерживает уведомление о событии
          BioAPI_NOTIFY_SOURCE_PRESENT или уведомление о
          событии получено в пределах указанной
          максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>
<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации. -->

<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <output name="NewTemplate" set-
     var="newtemplate_handle"/>
     <return setvar="return"/>
</invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается заклю-
чение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Enroll возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetBIRFromHandle с предоставлением
дескриптора зарегистрированной ЗБИ. -->
<invoke function="BioSPI_GetBIRFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="newtemplate_handle"/>
     <output name="HeaderVersion"
     setvar="headerversion"/>
     <output name="ProcessedLevel"
     setvar="processedlevel"/>
     <output name="FormatOwner" setvar="formatowner"/>
     <output name="FormatType" setvar="formattype"/>
     <output name="Quality" setvar="quality"/>
     <output name="Purpose" setvar="purpose"/>
     <output name="ProductOwner" setvar="productowner" />
     <output name="ProductType" setvar="producttype" />
     <output name="Creation_Year" setvar="creationyear" />
     <output name="Creation_Month" setvar="creationmonth" />
     <output name="Creation_Day" setvar="creationday" />
     <output name="Creation_Hour" setvar="creationhour" />
     <output name="Creation_Minute"
     setvar="creationminute" />
```

```
        <output name="Creation_Second"
        setvar="creationsecond" />
        <output name="Expiration_Year"
        setvar="expirationyear" />
        <output name="Expiration_Month"
        setvar="expirationmonth" />
        <output name="Expiration_Day" setvar="expirationday" />
        <output name="SBFormatOwner"
        setvar="securityformatowner" />
        <output name="SBFormatType"
        setvar="securityformattype" />
        <output name="Index" setvar="index" />
        <output name="BiometricData" setvar="biometircdata" />
        <output name="SecurityBlock"
        setvar="securityblock"/>
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
        <description>
                Функция BioSPI_GetBIRFromHandle возвращает
                значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
<invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
        <input name="bspUuid" var="bspUuid" />
        <input name="BSPHandle" var="_bsphandle" />
</invoke>
```

```
    </activity>
</package>
```

8.20 Утверждение 6b *BioSPI_GetBIRFromHandle_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью определения возвращения сообщения об ошибке при вызове функции *BioSPI_GetBIRFromHandle* с недействительным дескриптором ПБУ.

Выдержки

Подпункт 9.3.2.2

BioAPI_RETURN BioAPI BioSPI_GetBIRFromHandle

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_HANDLE Handle,

*BioAPI_BIR *BIR).*

Пункт 8.2.2

Данная функция возвращает ЗБИ, связанную с дескриптором ЗБИ, возвращенным ПБУ.

Возвращаемое значение: Значение BioAPI_RETURN указывает на успешное выполнение функции или определяет тип ошибки. Значение BioAPI_OK соответствует успешному выполнению. Остальные значения определяют тип ошибки.

Параметры: BSPHandle (входной параметр) – дескриптор присоединенного ПБУ.

Ссылки: 9.3.2.2 и 8.2.2.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Зарегистрировать ЗБИ для получения дескриптора ЗБИ.

4) Вызвать функцию *BioSPI_GetBIRFromHandle* с недействительным дескриптором ПБУ. Предполагается, что функция возвратит ошибку.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_GetBIRFromHandle* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="02445668-0cc5-1085-a3ac-0002a5d5fd2e">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение "BioSPI_ GetBIRFromHan-
        dle_InvalidBSPHandle" (см. далее элемент <description>
        утверждения).
    </description>

    <assertion name="BioSPI_GetBIRFromHandle_InvalidBSPHandle"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью
            определения возвращения сообщения об ошибке при
            вызове функции BioSPI_GetBIRFromHandle
            с недействительным дескриптором ПБУ.
            Следующий текст соответствует приведенному в 9.3.2.2
            и 8.2.2 спецификации БиоАПИ 2.0.
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_GetBIRFromHandle
                (BioAPI_HANDLE BSPHandle,
                BioAPI_BIR_HANDLE Handle,
                BioAPI_BIR *BIR).

            Примечание - Подробное определение функции
            BioAPI_GetBIRFromHandle приведено в 8.2.2.
            ------------------------------------------------
            Пункт 8.2.2:
```

```
        Данная функция возвращает ЗБИ, связанную с дескриптором
        ЗБИ, возвращенным ПБУ.
        Возвращаемое значение: Значение BioAPI_RETURN
        указывает на успешное выполнение функции или определяет
        тип ошибки. Значение BioAPI_OK соответствует
        успешному выполнению. Остальные значения определяют
        тип ошибки.
        Параметры: BSPHandle (входной параметр) - дескриптор
        присоединенного ПБУ.
        ------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Зарегистрировать ЗБИ для получения дескриптора ЗБИ.
        4) Вызвать функцию BioSPI_GetBIRFromHandle с
        недействительным дескриптором ПБУ. Предполагается,
        что функция возвратит ошибку.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>

    <!-- Указание на поддержку испытуемым ПБУ уведомления о
    событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
    <input name="_noSourcePresentSupported" />
```

```
        <!-- Время ожидания для события
        BioAPI_NOTIFY_SOURCE_PRESENT. -->
        <input name="_sourcepresenttimeout"/>

        <!-- Время ожидания для функции BioSPI_Enroll. -->
        <input name="_capturetimeout"/>

        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_GetBIRFromHandle">
             <input name="bspUuid" var="_bspUuid"/>
             <input name="inserttimeouttime"
             var="_inserttimeout"/>
             <input name="nosourcepresentsupported"
             var="_noSourcePresentSupported" />
             <input name="sourcepresenttimeouttime"
             var="_sourcepresenttimeout"/>
             <input name="capturetimeouttime"
             var="_capturetimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_GetBIRFromHandle">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
```

```
<input name="capturetimeouttime"/>

<!-- Данное утверждение использует функцию BSPHandle со
значением "1" для всех вызовов BioSPI, которые этого
требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ заявляет о поддержке уведомления
о событии BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать
получения уведомления в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
```

```
процесса прерывается, в противном случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <output name="NewTemplate" set-
     var="newtemplate_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Enroll возвращает значение
          BioAPI_OK.
```

```
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Вызов функции BioSPI_GetBIRFromHandle с
    недействительным дескриптором модуля. -->
    <invoke function="BioSPI_GetBIRFromHandle">
        <input name="BSPHandle" value="0"/>
        <input name="Handle" var="newtemplate_handle"/>
        <output name="HeaderVersion"
        setvar="headerversion"/>
        <output name="ProcessedLevel"
        setvar="processedlevel"/>
        <output name="FormatOwner" setvar="formatowner"/>
        <output name="FormatType" setvar="formattype"/>
        <output name="Quality" setvar="quality"/>
        <output name="Purpose" setvar="purpose"/>
        <output name="BiometricData"
        setvar="biometricdata"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "FAIL", в противном
    случае выдается заключение о соответствии "PASS". -->
    <assert_condition>
        <description>
            Функция BioSPI_GetBIRFromHandle возвращает
            значение BioAPIERR_INVALID_BSP_HANDLE.
        </description>
        <equal_to var1="return"
        var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
    </assert_condition>
```

```
        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload.-->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.21 Утверждение 6c *BioSPI_GetBIRFromHandle_InvalidBIRHandle*

Описание: Проверку данного утверждения проводят с целью определения возвращения сообщения об ошибке при вызове функции *BioSPI_GetBIRFromHandle* с недействительным дескриптором ЗБИ.

Выдержки

Подпункт 9.3.2.2

BioAPI_RETURN BioAPI BioAPI_GetBIRFromHandle

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_HANDLE Handle,

*BioAPI_BIR *BIR).*

Пункт 8.2.2

Данная функция возвращает ЗБИ, связанную с дескриптором ЗБИ, возвращенным ПБУ.

Параметры: Handle (входной параметр) – дескриптор ЗБИ, заголовок которой необходимо восстановить.

Возвращаемое значение: Значение BioAPI_RETURN указывает на успешное выполнение функции или определяет тип ошибки. Значение BioAPI_OK соответствует успешному выполнению. Остальные значения определяют тип ошибки.

Ссылки: 9.3.2.2 и 8.2.2.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Зарегистрировать ЗБИ для получения дескриптора ЗБИ.

4) Вызвать функцию *BioSPI_GetBIRFromHandle* с недействительным дескриптором ЗБИ. Предполагается, что функция возвратит ошибку.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_GetBIRFromHandle* возвращает значение BioAPIERR_INVALID_BIR_HANDLE.

Пакет языка утверждения

```
<package name="0194a9c0-0cc7-1085-8780-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_ GetBIRFromHan-
          dle_InvalidBIRHandle" (см. далее элемент <description>
          утверждения).
     </description>

     <assertion name="BioSPI_GetBIRFromHandle_InvalidBIRHandle"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               определения возвращения сообщения об ошибке при
               вызове функции BioSPI_GetBIRFromHandle с
               недействительным дескриптором ЗБИ.
               Следующий текст соответствует приведенному в 9.3.2.2
               и 8.2.2 спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioSPI_GetBIRFromHandle
                    (BioAPI_HANDLE BSPHandle,
                    BioAPI_BIR_HANDLE Handle,
                    BioAPI_BIR *BIR).
```

```
      Примечание - Подробное определение функции
      BioAPI_GetBIRFromHandle приведено в 8.2.2.
      ---------------------------------------------------
      Пункт 8.2.2:
      Данная функция возвращает ЗБИ, связанную с дескриптором
      ЗБИ, возвращенным ПБУ.
      Параметры: Handle (входной параметр) - дескриптор
      ЗБИ, заголовок которой должен быть восстановлен.
      Возвращаемое значение: Значение BioAPI_RETURN указывает
      на успешное выполнение функции или определяет тип
      ошибки. Значение BioAPI_OK означает успешное выпол-
      нение. Остальные значения определяют тип ошибки.
      ---------------------------------------------------
      Порядок действий при испытании на соответствие:

      1) Загрузить испытуемый ПБУ.
      2) Присоединить испытуемый ПБУ.
      3) Зарегистрировать ЗБИ для получения дескриптора ЗБИ.
      4) Вызвать функцию BioSPI_GetBIRFromHandle с
      недостоверным дескриптором ЗБИ. Предполагается, что
      функция возвратит ошибку.

      Если какие-либо промежуточные операции прошли
      неуспешно, то выдается заключение о соответствии
      "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на поддержку испытуемым ПБУ уведомления о
событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />
```

```
        <!-- Время ожидания для события
        BioAPI_NOTIFY_SOURCE_PRESENT. -->
        <input name="_sourcepresenttimeout"/>

        <!-- Время ожидания для функции BioSPI_Enroll. -->
        <input name="_capturetimeout"/>

        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_GetBIRFromHandle">
             <input name="bspUuid" var="_bspUuid"/>
             <input name="inserttimeouttime"
             var="_inserttimeout"/>
             <input name="nosourcepresentsupported"
             var="_noSourcePresentSupported"/>
             <input name="sourcepresenttimeouttime"
             var="_sourcepresenttimeout"/>
             <input name="capturetimeouttime"
             var="_capturetimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_GetBIRFromHandle">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
```

```
<input name="nosourcepresentsupported"/>
<input name="sourcepresenttimeouttime"/>
<input name="capturetimeouttime"/>

<!-- Данное утверждение использует функцию BSPHandle со
значением "1" для всех вызовов BioSPI, которые этого
требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>
<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать уведомления о
событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent"/>
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
```

```
процесса прерывается, в противном случае выдается заклю-
чение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию BioSPI
для регистрации. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <output name="NewTemplate"
     setvar="newtemplate_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
```

```
            Функция BioSPI_Enroll возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetBIRFromHandle с
недействительным дескриптором ЗБИ. -->
<invoke function="BioSPI_GetBIRFromHandle">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Handle" value="-1"/>
        <output name="HeaderVersion"
        setvar="headerversion"/>
        <output name="ProcessedLevel"
        setvar="processedLevel"/>
        <output name="FormatOwner" setvar="formatowner"/>
        <output name="FormatType" setvar="formattype"/>
        <output name="Quality" setvar="quality"/>
        <output name="Purpose" setvar="purpose"/>
        <output name="BiometricData"
        setvar="biometricdata"/>
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
        <description>
            Функция BioSPI_GetBIRFromHandle возвращает
            значение BioAPIERR_INVALID_BIR_HANDLE.
        </description>
        <equal_to var1="return"
        var2="__BioAPIERR_BSP_INVALID_BIR_HANDLE"/>
</assert_condition>
```

```
        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid"/>
            <input name="BSPHandle" var="_bsphandle"/>
        </invoke>
    </activity>
</package>
```

8.22 Утверждение 7а *BioSPI_GetHeaderFromHandle_ValidParam*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_GetHeaderFromHandle* с достоверными значениями входных параметров.

Выдержки

Подпункт 9.3.2.3

BioAPI_RETURN BioAPI BioSPI_GetHeaderFromHandle

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_HANDLE Handle,

*BioAPI_BIR_HEADER *Header).*

Пункт 8.2.3

Функция извлекает заголовок ЗБИ, определенной дескриптором.

Раздел А.4

Данная функция должна поддерживаться всеми типами ПБУ.

Ссылки: 9.3.2.3, 8.2.3 и А.4.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll*.

4) Вызвать функцию *BioSPI_GetHeaderFromHandle*.

5) Проверить возвращаемый код вызова.

6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_GetHeaderFromHandle* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="027a7db0-0cc7-1085-9391-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_ GetHeaderFrom-
          Handle_ValidParam" (см. далее элемент
          <description> утверждения).
     </description>

     <assertion name="BioSPI_GetHeaderFromHandle_ValidParam"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               определения возвращения значения BioAPI_OK при вызове
               функции BioSPI_GetHeaderFromHandle с достоверными
               значениями входных параметров.
               Следующий текст соответствует приведенному в 9.3.2.3
               и 8.2.3 и А.4 спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioSPI_GetHeaderFromHandle
                    (BioAPI_HANDLE BSPHandle,
                    BioAPI_BIR_HANDLE Handle,
                    BioAPI_BIR_HEADER *Header).

               Примечание - Подробное определение функции
               BioAPI_GetHeaderFromHandle приведено в 8.2.3.
```

```
        ------------------------------------------------
        Пункт 8.2.3
        Данная функция извлекает заголовок ЗБИ, определенной
        дескриптором.

        Раздел А.4
        Данная функция должна поддерживаться всеми типами ПБУ.
        ------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Вызвать функцию BioSPI_Enroll.
        4) Вызвать функцию BioSPI_GetHeaderFromHandle.
        5) Проверить возвращаемый код вызова.
        6) Отсоединить и выгрузить ПБУ.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ.-->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT.-->
<input name="_inserttimeout"/>

<!-- Указание на поддержку испытуемым ПБУ уведомления о
событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_
SOURCE_PRESENT. -->
<input name="_sourcepresenttimeout"/>
```

```
        <!-- Время ожидания для функции BioSPI_Enroll. -->
        <input name="_capturetimeout"/>

        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_GetHeaderFromHandle">
            <input name="bspUuid" var="_bspUuid"/>
            <input name="inserttimeouttime"
            var="_inserttimeout"/>
            <input name="nosourcepresentsupported"
            var="_noSourcePresentSupported" />
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
        </invoke>
        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом.
        При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_GetHeaderFromHandle">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>
        <!-- Данное утверждение использует параметр BSPHandle со
        значением "1" для всех вызовов BioSPI, которые этого
        требуют.-->
        <set name="_bsphandle" value="1"/>
```

```
<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ.
Входное значение параметра "unitIDOrNull" равно "0",
поэтому утверждение будет тестировать модуль датчика,
выбранный ПБУ.-->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
```

```
            break_if_false="true">
        <description>
            Испытуемый ПБУ не поддерживает уведомление о
            событии BioAPI_NOTIFY_SOURCE_PRESENT или
            уведомление о событии получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Purpose"
        var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <output name="NewTemplate"
        setvar="newtemplate_handle"/>
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_Enroll возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
```

```
<!-- Вызов функции BioSPI_GetHeaderFromHandle. -->
<invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="newtemplate_handle"/>
     <output name="HeaderVersion"
     setvar="headerversion"/>
     <output name="ProcessedLevel"
     setvar="processedLevel"/>
     <output name="FormatOwner" setvar="formatowner"/>
     <output name="FormatType" setvar="formattype"/>
     <output name="Quality" setvar="quality"/>
     <output name="Purpose" setvar="purpose"/>
     <output name="ProductOwner" setvar="productowner" />
     <output name="ProductType" setvar="producttype" />
     <output name="Creation_Year" setvar="creationyear" />
     <output name="Creation_Month" setvar="creationmonth" />
     <output name="Creation_Day" setvar="creationday" />
     <output name="Creation_Hour" setvar="creationhour" />
     <output name="Creation_Minute"
     setvar="creationminute" />
     <output name="Creation_Second"
     setvar="creationsecond" />
     <output name="Expiration_Year"
     setvar="expirationyear" />
     <output name="Expiration_Month"
     setvar="expirationmonth" />
     <output name="Expiration_Day" setvar="expirationday" />
     <output name="SBFormatOwner"
     setvar="securityformatowner" />
     <output name="SBFormatType"
     setvar="securityformattype" />
     <output name="Index" setvar="index" />
     <return setvar="return"/>
</invoke>
```

```
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Функция BioSPI_GetHeaderFromHandle возвращает
                значение BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.23 Утверждение 7b *BioSPI_GetHeaderFromHandle_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью определения возвращения сообщения об ошибке при вызове функции *BioSPI_GetHeaderFromHandle* с недействительным дескриптором модуля.

Выдержки

Подпункт 9.3.2.3

BioAPI_RETURN BioAPI BioSPI_GetHeaderFromHandle

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_HANDLE Handle,

*BioAPI_BIR_HEADER *Header).*

Пункт 8.2.3

Функция восстанавливает заголовок ЗБИ, определенной дескриптором.

Параметры: BSPHandle (входной параметр) – дескриптор присоединенного ПБУ.

Ссылки: 9.3.2.3 и 8.2.3.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll*.

4) Вызвать функцию *BioSPI_GetHeaderFromHandle* с недействительным дескриптором ПБУ.

5) Проверить возвращаемый код вызова.

6) Отсоединить и вывести испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_GetHeaderFromHandle* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="057e0d38-0ccd-1085-83b8-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_GetHeaderFromHandle_ InvalidBSPHandle" (см. далее
          элемент <description> утверждения).
     </description>

     <assertion name="BioSPI_GetHeaderFromHandle_InvalidBSPHandle"
     model="BSPTesting">
          <description>
```

Проверку данного утверждения проводят с целью определения возвращения сообщения об ошибке при вызове функции BioSPI_GetHeaderFromHandle с недействительным дескриптором модуля.
Следующий текст соответствует приведенному в 9.3.2.3 и 8.2.3 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_GetHeaderFromHandle
     (BioAPI_HANDLE BSPHandle,
     BioAPI_BIR_HANDLE Handle,
     BioAPI_BIR_HEADER *Header).
```

Примечание – Подробное определение функции BioAPI_GetHeaderFromHandle приведено в 8.2.3.

Пункт 8.2.3:
Функция восстанавливает заголовок ЗБИ, определенной дескриптором.
Параметры: BSPHandle (входной параметр) – дескриптор присоединенного ПБУ.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Enroll.
4) Вызвать функцию BioSPI_GetHeaderFromHandle с недействительным дескриптором ПБУ.
5) Проверить возвращаемый код вызова.
6) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

</description>

```
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на поддержку испытуемым ПБУ уведомления о
событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события
BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Enroll. -->
<input name="_capturetimeout"/>

<!--Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_GetHeaderFromHandle">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported"/>
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом ПБУ.
```

```
     При каждом входящем вызове функции связанный с ней про-
     цесс будет вызываться автоматически. -->
     <bind activity="EventHandler"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_GetHeaderFromHandle">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"/>
     <input name="capturetimeouttime"/>

     <!-- Данное утверждение использует параметр BSPHandle со
     значением "1" для всех вызовов BioSPI, которые этого
     требуют.-->
     <set name="_bsphandle" value="1"/>

     <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
     предоставленных испытуемым ПБУ.
     Входное значение параметра "unitIDOrNull" равно "0",
     поэтому утверждение будет тестировать модуль датчика,
     выбранный ПБУ. -->
     <invoke activity="LoadAndAttach"
               package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
               break_on_break="true">
          <input name="bspUuid" var="bspUuid"/>
          <input name="bspVersion" value="32"/>
          <input name="unitIDOrNull" value="0"/>
          <input name="bspHandle" var="_bsphandle"/>
          <input name="eventtimeouttime"
          var="inserttimeouttime"/>
     </invoke>

     <set name="eventtimeoutflag" value="false"/>
```

```
<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления в пределах указанной максимальной
продолжительности.-->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent"/>
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается заклю-
чение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов кполучению ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
```

```
    <output name="NewTemplate" set-
    var="newtemplate_handle"/>
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается заклю-
чение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
    <description>
          Функция BioSPI_Enroll возвращает значение
          BioAPI_OK.
    </description>
    <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetHeaderFromHandle с
недействительным дескриптором ПБУ. -->
<invoke function="BioSPI_GetHeaderFromHandle">
    <input name="BSPHandle" value="0"/>
    <input name="Handle" var="newtemplate_handle"/>
    <output name="HeaderVersion"
    setvar="headerversion"/>
    <output name="ProcessedLevel"
    setvar="processedLevel"/>
    <output name="FormatOwner" setvar="formatowner"/>
    <output name="FormatType" setvar="formattype"/>
    <output name="Quality" setvar="quality"/>
    <output name="Purpose" setvar="purpose"/>
    <return setvar="return"/>
</invoke>
```

```
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Функция BioSPI_GetHeaderFromHandle возвращает
                значение BioAPIERR_INVALID_BSP_HANDLE.
            </description>
            <equal_to var1="return"
            var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid"/>
            <input name="BSPHandle" var="_bsphandle"/>
        </invoke>
    </activity>
</package>
```

8.24 Утверждение 7c *BioSPI_GetHeaderFromHandle_InvalidBIRHandle*

Описание: Проверку данного утверждения проводят с целью определения возвращения сообщения об ошибке при вызове функции *BioSPI_GetHeaderFromHandle* с недостоверным дескриптором ЗБИ.

Выдержки

Подпункт 9.3.2.3

BioAPI_RETURN BioAPI BioSPI_GetHeaderFromHandle

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_HANDLE Handle,

*BioAPI_BIR_HEADER *Header).*

Пункт 8.2.3

Функция извлекает заголовок ЗБИ, определенной дескриптором.

Параметры: Handle (входной параметр) – дескриптор ЗБИ, заголовок которой необходимо извлечь.

Ошибки: BioAPIERR_INVALID_BIR_HANDLE.

Ссылки: 9.3.2.3 и 8.2.3.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll*.

4) Вызвать функцию *BioSPI_GetHeaderFromHandle* с недостоверным дескриптором ЗБИ.

5) Проверить возвращаемый код вызова.

6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_GetHeaderFromHandle* возвращает значение BioAPIERR_INVALID_BIR_HANDLE.

Пакет языка утверждения

```
<package name="02195e68-0cce-1085-a46f-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_GetHeaderFromHandle_InvalidBIRHandle" (см. далее
          элемент <description> утверждения).
     </description>

     <assertion name="BioSPI_GetHeaderFromHandle_InvalidBIRHandle"
     model="BSPTesting">
          <description>
```

Проверку данного утверждения проводят с целью определения возвращения сообщения об ошибке при вызове функции BioSPI_GetHeaderFromHandle с недействительным дескриптором ЗБИ.
Следующий текст соответствует приведенному в 9.3.2.3 и 8.2.3 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_GetHeaderFromHandle
     (BioAPI_HANDLE BSPHandle,
     BioAPI_BIR_HANDLE Handle,
     BioAPI_BIR_HEADER *Header).
```

Примечание - Подробное определение функции BioAPI_GetHeaderFromHandle приведено в 8.2.3.

Пункт 8.2.3:
Функция извлекает заголовок ЗБИ, определенной дескриптором.
Параметры: Handle (входной параметр) - дескриптор ЗБИ, заголовок которой необходимо извлечь.
Ошибки: BioAPIERR_INVALID_BIR_HANDLE.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Enroll.
4) Вызвать функцию BioSPI_GetHeaderFromHandle с недействительным дескриптором ЗБИ.
5) Проверить возвращаемый код вызова.
6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

```
</description>

<!-- УУИД испытуемого ПБУ.-->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на поддержку испытуемым ПБУ уведомления о
событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события
BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Enroll. -->
<input name="_capturetimeout"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->

<invoke activity="BioSPI_GetHeaderFromHandle">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
</invoke>
```

```
        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_GetHeaderFromHandle">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ.
        Входное значение параметра "unitIDOrNull" равно "0",
        поэтому утверждение будет тестировать модуль датчика,
        выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                 package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                 break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
            <input name="bspVersion" value="32"/>
            <input name="unitIDOrNull" value="0"/>
            <input name="bspHandle" var="_bsphandle"/>
            <input name="eventtimeouttime"
            var="inserttimeouttime"/>
        </invoke>
```

```
<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о
событии BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать
получения уведомления в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается заклю-
чение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
```

```
        <input name="Timeout" var="capturetimeouttime"/>
        <output name="NewTemplate"
        setvar="newtemplate_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_Enroll возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Вызов функции BioSPI_GetHeaderFromHandle. -->
    <invoke function="BioSPI_GetHeaderFromHandle">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Handle" value="-1" />
        <output name="HeaderVersion"
        setvar="headerversion"/>
        <output name="ProcessedLevel"
        setvar="processedLevel"/>
        <output name="FormatOwner" setvar="formatowner"/>
        <output name="FormatType" setvar="formattype"/>
        <output name="Quality" setvar="quality"/>
        <output name="Purpose" setvar="purpose"/>
        <return setvar="return"/>
    </invoke>
```

```
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Функция BioSPI_GetHeaderFromHandle возвращает
                значение BioAPIERR_INVALID_BIR_HANDLE.
            </description>
            <equal_to var1="return"
            var2="__BioAPIERR_BSP_INVALID_BIR_HANDLE"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.25 Утверждение 7d *BioSPI_GetHeaderFromHandle_BIRHandleNotFreed*

Описание: Проверку данного утверждения проводят с целью определения освобождения дескриптора ЗБИ после вызова функции *BioSPI_GetHeaderFrom Handle*.

Выдержки

Подпункт 9.3.2.3

BioAPI_RETURN BioAPI BioSPI_GetHeaderFromHandle

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_HANDLE Handle,

*BioAPI_BIR_HEADER *Header).*

Пункт 8.2.3

Функция извлекает заголовок ЗБИ, определенной дескриптором. Дескриптор ЗБИ не освобождается ПБУ.

Ссылки: 9.3.2.3 и 8.2.3.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll*.

4) Вызвать функцию *BioSPI_GetHeaderFromHandle*.

5) Вызвать функцию *BioSPI_GetBIRFromHandle*. Предполагается, что функция возвратит значение BioAPI_OK.

6) Выгрузить и отсоединить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции BioSPI_*GetBIRFromHandle* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="01cc0988-0ccf-1085-a367-0002a5d5fd2e" >
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_GetHeaderFromHandle_BIRHandleNotFreed" (см. далее
          элемент <description> утверждения).
     </description>

     <assertion name="BioSPI_GetHeaderFromHandle_BIRHandleNotFreed"
     model="BSPTesting">
          <description>
```

```
Проверку данного утверждения проводят с целью
определения высвобождения дескриптора ЗБИ после
вызова функции BioSPI_GetHeaderFromHandle.
Следующий текст соответствует приведенному в 9.3.2.3
и 8.2.3 спецификации БиоАПИ 2.0.
-------------------------------------------------
BioAPI_RETURN BioAPI BioSPI_GetHeaderFromHandle
     (BioAPI_HANDLE BSPHandle,
     BioAPI_BIR_HANDLE Handle,
     BioAPI_BIR_HEADER *Header).

Примечание - Подробное определение функции
BioAPI_GetHeaderFromHandle приведено в 8.2.3.
-------------------------------------------------
Пункт 8.2.3:
Функция извлекает заголовок ЗБИ, определенной
дескриптором. ПБУ не высвобождает дескриптор ЗБИ.
Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Enroll.
4) Вызвать функцию BioSPI_GetHeaderFromHandle.
5) Вызвать функцию BioSPI_GetBIRFromHandle.
Предполагается возвращение значения BioAPI_OK.
6) Выгрузить и отсоединить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли
неуспешно, то выдается заключение о соответствии
"UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
```

```
<input name="_inserttimeout"/>

<!-- Указание на поддержку испытуемым ПБУ уведомления о
событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события
BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Enroll. -->
<input name="_capturetimeout"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_GetHeaderFromHandle">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported"/>
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически. -->
<bind activity="EventHandler"
     package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
     function="BioSPI_EventHandler"/>
```

```
</assertion>

<activity name="BioSPI_GetHeaderFromHandle">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"/>
     <input name="capturetimeouttime"/>

     <!-- Данное утверждение использует BSPHandle со значением
     "1" для всех вызовов BioSPI, которые этого требуют. -->
     <set name="_bsphandle" value="1"/>

     <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
     предоставленных испытуемым ПБУ.
     Входное значение параметра "unitIDOrNull" равно "0",
     поэтому утверждение будет тестировать модуль датчика,
     выбранный ПБУ. -->
     <invoke activity="LoadAndAttach"
               package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
               break_on_break="true">
          <input name="bspUuid" var="bspUuid"/>
          <input name="bspVersion" value="32"/>
          <input name="unitIDOrNull" value="0"/>
          <input name="bspHandle" var="_bsphandle"/>
          <input name="eventtimeouttime"
          var="inserttimeouttime"/>
     </invoke>

     <set name="eventtimeoutflag" value="false"/>

     <!-- Если испытуемый ПБУ поддерживает уведомление о событии
     BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
     уведомления в пределах указанной максимальной
     продолжительности. -->
     <wait_until timeout_var="sourcepresenttimeouttime"
```

```
            setvar="eventtimeoutflag">
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent"/>
    </wait_until>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Испытуемый ПБУ не поддерживает уведомление о
            событии BioAPI_NOTIFY_SOURCE_PRESENT, или
            уведомление о событии было получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
    </assert_condition>

    <!-- ПБУ готов к получению ЗБИ. Вызвать функцию
    BioSPI_Enroll для регистрации. -->
    <invoke function="BioSPI_Enroll">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Purpose"
        var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <output name="NewTemplate"
        setvar="newtemplate_handle"/>
        <return setvar="return"/>
    </invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Enroll возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetHeaderFromHandle. -->
<invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="newtemplate_handle"/>
     <output name="HeaderVersion"
     setvar="headerversion"/>
     <output name="ProcessedLevel"
     setvar="processedLevel"/>
     <output name="FormatOwner" setvar="formatowner"/>
     <output name="FormatType" setvar="formattype"/>
     <output name="Quality" setvar="quality"/>
     <output name="Purpose" setvar="purpose"/>
     <output name="ProductOwner" setvar="productowner" />
     <output name="ProductType" setvar="producttype" />
     <output name="Creation_Year" setvar="creationyear" />
     <output name="Creation_Month" setvar="creationmonth" />
     <output name="Creation_Day" setvar="creationday" />
     <output name="Creation_Hour" setvar="creationhour" />
     <output name="Creation_Minute"
     setvar="creationminute" />
     <output name="Creation_Second"
     setvar="creationsecond" />
```

```
        <output name="Expiration_Year"
        setvar="expirationyear" />
        <output name="Expiration_Month"
        setvar="expirationmonth" />
        <output name="Expiration_Day" setvar="expirationday" />
        <output name="SBFormatOwner"
        setvar="securityformatowner" />
        <output name="SBFormatType"
        setvar="securityformattype" />
        <output name="Index" setvar="index" />
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided">
        <description>
                Функция BioSPI_GetHeaderFromHandle возвращает
                значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetBIRFromHandle с предоставлением
дескриптора зарегистрированной ЗБИ. -->
<invoke function="BioSPI_GetBIRFromHandle">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Handle" var="newtemplate_handle"/>
        <output name="HeaderVersion"
        setvar="headerversion"/>
        <output name="ProcessedLevel"
        setvar="processedlevel"/>
        <output name="FormatOwner" setvar="formatowner"/>
```

```
        <output name="FormatType" setvar="formattype"/>
        <output name="Quality" setvar="quality"/>
        <output name="Purpose" setvar="purpose"/>
        <output name="ProductOwner" setvar="productowner" />
        <output name="ProductType" setvar="producttype" />
        <output name="Creation_Year" setvar="creationyear" />
        <output name="Creation_Month" setvar="creationmonth" />
        <output name="Creation_Day" setvar="creationday" />
        <output name="Creation_Hour" setvar="creationhour" />
        <output name="Creation_Minute"
        setvar="creationminute" />
        <output name="Creation_Second"
        setvar="creationsecond" />
        <output name="Expiration_Year"
        setvar="expirationyear" />
        <output name="Expiration_Month"
        setvar="expirationmonth" />
        <output name="Expiration_Day" setvar="expirationday" />
        <output name="SBFormatOwner"
        setvar="securityformatowner" />
        <output name="SBFormatType"
        setvar="securityformattype" />
        <output name="Index" setvar="index" />
        <output name="BiometricData" setvar="biometircdata" />
        <output name="SecurityBlock"
        setvar="securityblock"/>
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
        <description>
```

```
                Функция BioSPI_GetBIRFromHandle возвращает
                значение BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid"/>
            <input name="BSPHandle" var="_bsphandle"/>
        </invoke>
    </activity>
</package>
```

8.26 Утверждение 8a *BioSPI_EnableEvents_ValidParam*

Описание: Проверку данного утверждения проводят с целью испытания функции *BioSPI_EnableEvents* с достоверными значениями входных параметров.

Выдержки

Подпункт 9.3.3.1

BioAPI_RETURN BioAPI BioSPI_EnableEvents

(BioAPI_HANDLE BSPHandle,

BioAPI_EVENT_MASK Events).

Пункт 8.3.1

Данная функция с помощью маски EventMask разрешает события, поступающие со всех модулей БиоАПИ, выбранных в сессии присоединения ПБУ, определенной с помощью дескриптора ПБУ. События, не удовлетворяющие условиям маски, запрещают. События, поступающие с других модулей БиоАПИ, прямо или косвенно управляемые рассматриваемым ПБУ (возможно, выбранные в других, а не в данной сессии присоединения), не подвергаются воздействию.

Возвращаемое значение: Значение BioAPI_RETURN указывает на успешное выполнение функции или определяет тип ошибки. Значение BioAPI_OK соответствует успешному выполнению. Остальные значения определяют тип ошибки.

Ссылки: 9.3.3.1 и 8.3.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_EnableEvents* с определенной маской событий.

4) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

5) С помощью функции *BioSPI_EventHandler* убедиться в том, что разрешенные события могут быть приняты, а запрещенные не будут приняты.

6) Отсоединить и выгрузитьиспытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Разрешенные события принимаются, а запрещенные не принимаются.

Пакет языка утверждения

```
<package name="0333f628-0ccf-1085-aceb-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_EnableEvents_ValidParam" (см. Далее элемент
          <description> утверждения).
     </description>

     <assertion name="BioSPI_EnableEvents_ValidParam"
     model="BSPTesting">
```

<description>

Проверку данного утверждения проводят с целью испытания функции BioSPI_EnableEventsc достоверными значениями входных параметров.

Следующий текст соответствует приведенному в 9.3.3.1 и 8.3.1 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_EnableEvents
     (BioAPI_HANDLE BSPHandle,
     BioAPI_EVENT_MASKEvents);
```

Примечание - Подробное определение функции BioAPI_EnableEvents приведено в 8.3.1.

Пункт 8.3.1

Данная функция с помощью маски Event Mask разрешает события, поступающие со всех модулей БиоАПИ, выбранных в сессии присоединения ПБУ, определенной с помощью дескриптора ПБУ. События, не удовлетворяющие условиям маски, запрещаются. События, поступающие с других модулей БиоАПИ, прямо и косвенно управляемые рассматриваемым ПБУ (возможно, выбранные в других, а не в данной сессии присоединения), не подвергаются воздействию.

Возвращаемое значение: значение BioAPI_RETURN указывает на успешное выполнение функции или определяет тип ошибки. Значение BioAPI_OK соответствует успешному выполнению. Остальные значения определяют тип ошибки.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.

```
      3) Вызвать функцию BioSPI_EnableEvents с определенной
      маской событий.
      4) Проверить возвращаемый код. Предполагается
      возвращение значения BioAPI_OK.
      5) С помощью функции BioSPI_EventHandler убедиться в
      том, что разрешенные события будут приняты, а
      запрещенные события не будут приняты.
      6) Отсоединить и выгрузить испытуемый ПБУ.

      Если какие-либо промежуточные операции прошли
      неуспешно, то выдается заключение о соответствии
      "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на разрешение уведомления о событии
BioAPI_NOTIFY_INSERT. -->
<input name="_eventNotifyInsert"/>

<!-- Указание на разрешение уведомления о событии
BioAPI_NOTIFY_REMOVE. -->
<input name="_eventNotifyRemove"/>

<!-- Указание на разрешение уведомления о событии
BioAPI_NOTIFY_FAULT. -->
<input name="_eventNotifyFault"/>

<!-- Указание на разрешение уведомления о событии
BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_eventNotifySourcePresent"/>
```

```
        <!-- Указание на разрешение уведомления о событии
        BioAPI_NOTIFY_SOURCE_REMOVED. -->
        <input name="_eventNotifySourceRemoved"/>

        <!-- Время ожидания. -->
        <input name="_timeout"/>

        <!--Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_EnableEvents">
             <input name="bspUuid" var="_bspUuid"/>
             <input name="eventtimeouttime"
             var="_inserttimeout"/>
             <input name="eventnotifyinsert"
             var="_eventNotifyInsert"/>
             <input name="eventnotifyremove"
             var="_eventNotifyRemove"/>
             <input name="eventnotifyfault"
             var="_eventNotifyFault"/>
             <input name="eventnotifysourcepresent"
             var="_eventNotifySourcePresent"/>
             <input name="eventnotifysourceremoved"
             var="_eventNotifySourceRemoved"/>
             <input name="timeout" var="_timeout" />
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного
        вызова инфраструктуры, предоставленной испытуемым
        компонентом ПБУ. При каждом входящем вызове функции связанный
        с ней процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             function="BioSPI_EventHandler"/>
    </assertion>
```

```
<activity name="BioSPI_EnableEvents">
     <input name="bspUuid"/>
     <input name="eventtimeouttime"/>
     <input name="eventnotifyinsert" />
     <input name="eventnotifyremove" />
     <input name="eventnotifyfault" />
     <input name="eventnotifysourcepresent" />
     <input name="eventnotifysourceremoved" />
     <input name="timeout" />

     <set name="_bsphandle" value="1" />

     <set name="_enableEventsCalled" value="false" />

     <!-- Инициализация глобальной переменной "_insert"
     значением "false". Событие "EventHandler" присваивает
     данной переменной значение "true" после получения
     уведомления о событии BioAPI_NOTIFY_INSERT и значение
     "false" после получения уведомления о событии BioAPI_
     NOTIFY_REMOVE. -->
     <set name="_insert" value="false"/>

     <!-- Вызов функции BioSPI_BSPLoad. -->
     <invoke function="BioSPI_BSPLoad">
          <input name="BSPUuid" var="bspUuid"/>
          <input name="BioAPINotifyCallback" value="*"/>
          <input name="BFPEnumerationHandler" value="*"/>
          <input name="MemoryFreeHandler" value="*" />
          <return setvar="return"/>
     </invoke>

     <!-- Выдача заключения о соответствии. Если условие,
     определенное далее в элементе <description>, ложное,
     выдается заключение о соответствии "UNDECIDED" и выполнение
     процесса прерывается, в противном случае выдается
     заключение о соответствии "PASS". -->
     <assert_condition response_if_false="undecided"
```

```
            break_if_false="true">
        <description>
            Функция BioSPI_BSPLoad возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Получение уведомления о событии BioAPI_NOTIFY_INSERT
следует ждать в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="eventtimeouttime"
        setvar="eventtimeoutflag" var="_insert"/>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Уведомление о событии BioAPI_NOTIFY_INSERT
            получено в пределах указанной максимальной
            продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
</assert_condition>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается. -->
<assert_condition break_if_false="true">
        <description>
```

```
            Полученное уведомление о событии содержит
            достоверное описание категории модуля.
        </description>
        <or>
            <equal_to var1="_unitCategory"
            var2="__BioAPI_CATEGORY_ARCHIVE"/>
            <equal_to var1="_unitCategory"
            var2="__BioAPI_CATEGORY_MATCHING_ALG"/>
            <equal_to var1="_unitCategory"
            var2="__BioAPI_CATEGORY_PROCESSING_ALG"/>
            <equal_to var1="_unitCategory"
            var2="__BioAPI_CATEGORY_SENSOR"/>
        </or>
</assert_condition>

<!-- Вызов функции BioSPI_BSPAttach и явное присоединение
только одного модуля. -->
<invoke function="BioSPI_BSPAttach">
        <only_if>
            <equal_to var1="_unitCategory"
            var2="__BioAPI_CATEGORY_ARCHIVE"/>
        </only_if>
        <input name="BSPUuid" var="bspUuid"/>
        <input name="Version" value="32"/>
        <input name="Unit_1_UnitCategory"
        var="_unitCategory" />
        <input name="Unit_1_UnitID" var="_unitID"/>
        <input name="Unit_2_UnitCategory"
        var="__BioAPI_CATEGORY_MATCHING_ALG" />
        <input name="Unit_2_UnitID"
        var="__BioAPI_DONT_INCLUDE"/>
        <input name="Unit_3_UnitCategory"
        var="__BioAPI_CATEGORY_PROCESSING_ALG" />
        <input name="Unit_3_UnitID"
        var="__BioAPI_DONT_INCLUDE"/>
```

```
            <input name="Unit_4_UnitCategory"
            var="__BioAPI_CATEGORY_SENSOR" />
            <input name="Unit_4_UnitID"
            var="__BioAPI_DONT_INCLUDE"/>
            <input name="NumUnits" value="4" />
            <input name="BSPHandle" var="_bsphandle"/>
            <return setvar="return"/>
        </invoke>
        <invoke function="BioSPI_BSPAttach">
            <only_if>
                <equal_to var1="_unitCategory"
                var2="__BioAPI_CATEGORY_MATCHING_ALG"/>
            </only_if>
            <input name="BSPUuid" var="bspUuid"/>
            <input name="Version" value="32"/>
            <input name="Unit_1_UnitCategory"
            var="__BioAPI_CATEGORY_ARCHIVE" />
            <input name="Unit_1_UnitID"
            var="__BioAPI_DONT_INCLUDE"/>
            <input name="Unit_2_UnitCategory"
            var="_unitCategory" />
            <input name="Unit_2_UnitID" var="_unitID"/>
            <input name="Unit_3_UnitCategory"
            var="__BioAPI_CATEGORY_PROCESSING_ALG" />
            <input name="Unit_3_UnitID"
            var="__BioAPI_DONT_INCLUDE"/>
            <input name="Unit_4_UnitCategory"
            var="__BioAPI_CATEGORY_SENSOR" />
            <input name="Unit_4_UnitID"
            var="__BioAPI_DONT_INCLUDE"/>
            <input name="NumUnits" value="4" />
            <input name="BSPHandle" var="_bsphandle"/>
            <return setvar="return"/>
        </invoke>
        <invoke function="BioSPI_BSPAttach">
```

```
        <only_if>
            <equal_to var1="_unitCategory"
            var2="__BioAPI_CATEGORY_PROCESSING_ALG"/>
        </only_if>
        <input name="BSPUuid" var="bspUuid"/>
        <input name="Version" value="32"/>
        <input name="Unit_1_UnitCategory"
        var="__BioAPI_CATEGORY_ARCHIVE" />
        <input name="Unit_1_UnitID"
        var="__BioAPI_DONT_INCLUDE"/>
        <input name="Unit_2_UnitCategory"
        var="__BioAPI_CATEGORY_MATCHING_ALG" />
        <input name="Unit_2_UnitID"
        var="__BioAPI_DONT_INCLUDE"/>
        <input name="Unit_3_UnitCategory"
        var="_unitCategory" />
        <input name="Unit_3_UnitID" var="_unitID"/>
        <input name="Unit_4_UnitCategory"
        var="__BioAPI_CATEGORY_SENSOR" />
        <input name="Unit_4_UnitID"
        var="__BioAPI_DONT_INCLUDE"/>
        <input name="NumUnits" value="4" />
        <input name="BSPHandle" var="_bsphandle"/>
        <return setvar="return"/>
    </invoke>
    <invoke function="BioSPI_BSPAttach">
        <only_if>
            <equal_to var1="_unitCategory"
            var2="__BioAPI_CATEGORY_SENSOR"/>
        </only_if>
        <input name="BSPUuid" var="bspUuid"/>
        <input name="Version" value="32"/>
        <input name="Unit_1_UnitCategory"
        var="__BioAPI_CATEGORY_ARCHIVE" />
        <input name="Unit_1_UnitID"
        var="__BioAPI_DONT_INCLUDE"/>
```

```
        <input name="Unit_2_UnitCategory"
        var="__BioAPI_CATEGORY_MATCHING_ALG" />
        <input name="Unit_2_UnitID"
        var="__BioAPI_DONT_INCLUDE"/>
        <input name="Unit_3_UnitCategory"
        var="__BioAPI_CATEGORY_PROCESSING_ALG" />
        <input name="Unit_3_UnitID"
        var="__BioAPI_DONT_INCLUDE"/>
        <input name="Unit_4_UnitCategory"
        var="_unitCategory" />
        <input name="Unit_4_UnitID" var="_unitID"/>
        <input name="NumUnits" value="4" />
        <input name="BSPHandle" var="_bsphandle"/>
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_BSPAttach возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_EnableEvents для разрешения или
запрещения событий. -->
<invoke function="BioSPI_EnableEvents" >
        <input name="BSPHandle" var="_bsphandle"/>
```

```
        <input name="EventNotifyInsert"
        var="eventnotifyinsert"/>
        <input name="EventNotifyRemove"
        var="eventnotifyremove"/>
        <input name="EventNotifyFault"
        var="eventnotifyfault"/>
        <input name="EventNotifySourcePresent"
        var="eventnotifysourcepresent"/>
        <input name="EventNotifySourceRemoved"
        var="eventnotifysourceremoved"/>
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition break_if_false="true">
        <description>
                Функция BioSPI_EnableEvents возвращает значение
                BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<set name="_enableEventsCalled" value="true" />
<set name="_eventFlag" value="false" />

<!-- Уведомление о событии, которое было запрещено, не
должно быть получено в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="timeout" var="_eventFlag"/>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
```

```
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
            <description>
                ПБУ не передает ни одного запрещенного
                уведомления о событии.
            </description>
            <not var="_eventFlag" />
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>

    <!-- Данный процесс будет вызван в ответ на входящий вызов
    функции BioSPI_ModuleEventHandler, предоставленной испытуемым
    компонентом ПБУ. В данном процессе значения глобальных
    переменных "_unitID", "_insert", "_sourcePresent" и "_eventtype"
    должны соответствовать входным значениям параметра. -->
    <activity name="EventHandler" atomic="true">
        <input name="BSPUuid"/>
        <input name="UnitID"/>
        <input name="UnitSchema_BspUuid" />
        <input name="UnitSchema_UnitManagerUuid" />
        <input name="UnitSchema_UnitId" />
        <input name="UnitSchema_UnitCategory" />
        <input name="UnitSchema_UnitProperties" />
        <input name="UnitSchema_VendorInformation" />
        <input name="UnitSchema_EventNotifyInsert" />
        <input name="UnitSchema_EventNotifyRemove" />
        <input name="UnitSchema_EventNotifyFault" />
```

```
<input name="UnitSchema_EventNotifySourcePresent" />
<input name="UnitSchema_EventNotifySourceRemoved" />
<input name="UnitSchema_UnitPropertyID" />
<input name="UnitSchema_UnitProperty" />
<input name="UnitSchema_HardwareVersion" />
<input name="UnitSchema_FirmwareVersion" />
<input name="UnitSchema_SoftwareVersion" />
<input name="UnitSchema_HardwareSerialNumber" />
<input name="UnitSchema_AuthenticatedHardware" />
<input name="UnitSchema_MaxBspDbSize" />
<input name="UnitSchema_MaxIdentify" />
<input name="EventType" />
<output name="return"/>

<!-- Проверить совместимость полученного уведомления о
событии с маской событий, установленной с помощью функции
BioSPI_EnableEvents. -->
<invoke activity="checkForUnexpectedEvent">
     <only_if>
          <same_as var1="_enableEventsCalled" val-
          ue2="true"/>
          <existing var="_unitID"/>
     </only_if>
     <input name="UnitID" var="UnitID"/>
     <input name="EventType" var="EventType"/>
</invoke>

<!-- Присвоить начальное значение глобальной переменной
"_unitID" если:
     - оно не присвоено;
     - уведомлением о событии является BioAPI_NOTIFY_INSERT
     или BioAPI_NOTIFY_SOURCE_PRESENT;-->
<set name="_unitID" var="UnitID">
     <only_if>
          <not>
               <existing var="_unitID"/>
```

```
                    </not>
                    <or>
                         <equal_to var1="EventType"
                              var2="__BioAPI_NOTIFY_INSERT"/>
                         <equal_to var1="EventType"
                              var2="__BioAPI_NOTIFY_SOURCE_PRESENT"/>
                    </or>
               </only_if>
          </set>

          <!-- Установка категории модуля. -->
          <set name="_unitCategory" var="UnitSchema_UnitCategory">
               <only_if>
                    <not>
                         <existing var="_unitCategory"/>
                    </not>
                    <equal_to var1="EventType"
                         var2="__BioAPI_NOTIFY_INSERT"/>
               </only_if>
          </set>

          <invoke activity="EventHandlerSetGlobalData"
               package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                    break_on_break="true">
               <only_if>
                    <existing var="_unitID"/>
               </only_if>
               <input name="UnitID" var="UnitID"/>
               <input name="EventType" var="EventType"/>
          </invoke>

          <set name="return" var="__BioAPI_OK"/>
     </activity>

     <activity name="checkForUnexpectedEvent" atomic="true">
```

```
<input name="UnitID"/>
<input name="EventType"/>
<set name="_eventFlag" value="true">
     <only_if>
          <equal_to var1="_unitID" var2="UnitID"/>
          <or>
               <and>
                    <equal_to var1="EventType"
                         var2="__BioAPI_NOTIFY_INSERT"/>
                    <not var="_eventNotifyInsert" />
               </and>
               <and>
                    <equal_to var1="EventType"
                         var2="__BioAPI_NOTIFY_REMOVE"/>
                    <not var="_eventNotifyRemove" />
               </and>
               <and>
                    <equal_to var1="EventType"
                         var2="__BioAPI_NOTIFY_FAULT"/>
                    <not var="_eventNotifyFault" />
               </and>
               <and>
                    <equal_to var1="EventType"
                         var2="__BioAPI_NOTIFY_SOURCE_
                         PRESENT"/>
                    <not var="_eventNotifySourcePresent"
                    />
               </and>
               <and>
                    <equal_to var1="EventType"
                         var2="__BioAPI_NOTIFY_SOURCE_
                         REMOVED"/>
                    <not var="_eventNotifySourceRemoved" />
               </and>
          </or>
     </only_if>
```

```
        </set>
    </activity>
</package>
```

8.27 Утверждение 8b *BioSPI_EnableEvents_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью испытания функции *BioSPI_EnableEvents* с недействительным дескриптором модуля.

Выдержки

Подпункт 9.3.3.1

BioAPI_RETURN BioAPI BioSPI_EnableEvents
(BioAPI_HANDLE BSPHandle,
BioAPI_EVENT_MASK Events).

Пункт 8.3.1

Данная функция с помощью маски EventMask разрешает события, поступающие со всех модулей БиоАПИ, выбранных в сессии присоединения ПБУ, определенной с помощью дескриптора ПБУ. События, не удовлетворяющие условиям маски, запрещаются. События, поступающие с других модулей БиоАПИ, прямо или косвенно управляемые рассматриваемым ПБУ (возможно, выбранные в других, а не в данной сессии присоединения), не подвергаются воздействию.

Возвращаемое значение: Значение BioAPI_RETURN указывает на успешное выполнение функции или определяет тип ошибки. Значение BioAPI_OK соответствует успешному выполнению. Остальные значения определяют тип ошибки.

Параметры: BSPHandle (входной параметр) – дескриптор присоединенного ПБУ.

Ссылки: 9.3.3.1 и 8.3.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_EnableEvents* с недействительным дескриптором ПБУ. Предполагается завершение функции с ошибкой.

4) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_EnableEvents* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
package name="04ed0838-0ccf-1085-b64e-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_ EnableE-
          vents_InvalidBSPHandle" (см. далее элемент
          <description> утверждения).
     </description>

     <assertion name="BioSPI_EnableEvents_InvalidBSPHandle"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               испытания функции BioSPI_EnableEvents с
               недействительным дескриптором модуля.
               Следующий текст соответствует приведенному в
               9.3.3.1, 8.3.1.1 и 8.3.1.2 спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioSPI_EnableEvents
                    (BioAPI_HANDLE BSPHandle,
                    BioAPI_EVENT_MASKEvents).

               Примечание - Подробное определение функции
               BioAPI_EnableEvents приведено в 8.3.1.
```

```
        ---------------------------------------------------
        Подпункт 8.3.1.1:
        Данная функция с помощью маски Event Mask разрешает
        события, поступающие со всех модулей БиоАПИ, выбранных
        в сессии присоединения ПБУ, определенной с помощью
        дескриптора ПБУ. События, не удовлетворяющие условиям
        маски, запрещаются. События, поступающие с других
        модулей БиоАПИ, прямо или косвенно управляемые
        рассматриваемым ПБУ (возможно, выбранные в других, а
        не в данной сессии присоединения), не подвергаются
        воздействию.
        Возвращаемое значение: Значение BioAPI_RETURN указывает
        на успешное выполнение функции или определяет тип
        ошибки. Значение BioAPI_OK соответствует успешному
        выполнению. Остальные значения определяют тип ошибки.
        ---------------------------------------------------
        Подпункт 8.3.1.2:
        Параметры: BSPHandle (входной параметр) - дескриптор
        присоединенного ПБУ.
        ---------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Вызвать функцию BioSPI_EnableEvents с недействи-
        тельным дескриптором ПБУ. Предполагается завершение
        функции с ошибкой.
        4) Отсоединить и выгрузить испытуемый ПБУ.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>
```

```
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Указание на разрешение уведомления о событии
BioAPI_NOTIFY_INSERT. -->
<input name="_eventNotifyInsert"/>

<!-- Указание на разрешение уведомления о событии
BioAPI_NOTIFY_REMOVE. -->
<input name="_eventNotifyRemove"/>

<!-- Указание на разрешение уведомления о событии
BioAPI_NOTIFY_FAULT. -->
<input name="_eventNotifyFault"/>

<!-- Указание на разрешение уведомления о событии
BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_eventNotifySourcePresent"/>

<!-- Указание на разрешение уведомления о событии
BioAPI_NOTIFY_SOURCE_REMOVED. -->
<input name="_eventNotifySourceRemoved"/>

<!-- Время ожидания для уведомления о событии
BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeouttime" />

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_EnableEvents">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="eventnotifyinsert"
     var="_eventNotifyInsert"/>
     <input name="eventnotifyremove"
     var="_eventNotifyRemove"/>
```

```
            <input name="eventnotifyfault"
            var="_eventNotifyFault"/>
            <input name="eventnotifysourcepresent"
            var="_eventNotifySourcePresent"/>
            <input name="eventnotifysourceremoved"
            var="_eventNotifySourceRemoved"/>
            <input name="inserttimeouttime"
            var="_inserttimeouttime" />
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_EnableEvents">
        <input name="bspUuid"/>
        <input name="eventnotifyinsert" />
        <input name="eventnotifyremove" />
        <input name="eventnotifyfault" />
        <input name="eventnotifysourcepresent" />
        <input name="eventnotifysourceremoved" />
        <input name="inserttimeouttime" />

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
```

```
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<!-- Вызов функции BioSPI_EnableEvents с недействительным
дескриптором модуля. -->
<invoke function="BioSPI_EnableEvents" >
     <input name="BSPHandle" value="0" />
     <input name="EventNotifyInsert"
     var="eventnotifyinsert" />
     <input name="EventNotifyRemove"
     var="eventnotifyremove" />
     <input name="EventNotifyFault"
     var="eventnotifyfault" />
     <input name="EventNotifySourcePresent"
     var="eventnotifysourcepresent" />
     <input name="EventNotifySourceRemoved"
     var="eventnotifysourceremoved" />
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
     <description>
```

```
                    Функция BioSPI_EnableEvents возвращает значение
                    BioAPIERR_INVALID_BSP_HANDLE.
               </description>
               <equal_to var1="return"
                    var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
          </assert_condition>

          <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
          <invoke activity="DetachAndUnload"
                    package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
               <input name="bspUuid" var="bspUuid" />
               <input name="BSPHandle" var="_bsphandle" />
          </invoke>
     </activity>
</package>
```

8.28 Утверждение 9а *BioSPI_Capture_AuditData*

Описание: Проверку данного утверждения проводят с целью испытания функции *BioSPI_Capture* определенным значением параметра AuditData, отличающимся от NULL.

Выдержки

Подпункт 9.3.4.1

BioAPI_RETURN BioAPI BioSPI_Capture

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_PURPOSE Purpose,

BioAPI_BIR_SUBTYPE Subtype,

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*BioAPI_BIR_HANDLE *CapturedBIR,*

int32_t Timeout,

*BioAPI_BIR_HANDLE *AuditData).*

Подпункт 8.4.1.1

Если значение параметра AuditData отличается от NULL, может быть возвращена ЗБИ с типом «исходный биометрический образец». ПБУ может возвратить значение дескриптора BioAPI_UNSUPPORTED_BIR_HANDLE для указания, что параметр AuditData не поддерживается, или значение BioAPI_INVALID_BIR_HANDLE для указания, что контрольные данные недоступны.

Параметры: AuditData (выходной, необязательный параметр) – дескриптор ЗБИ, содержащий исходный биометрический образец. Эти данные могут быть использованы для предоставления идентификационных данных человека, использующего биометрическое устройство.

Если указатель имеет входное значение NULL, то контрольные данные не были собраны. Не все ПБУ поддерживают функцию сбора контрольных данных.

ПБУ может вернуть значение дескриптора BioAPI_UNSUPPORTED_BIR_HANDLE для указания, что параметр AuditData не поддерживается, или значение BioAPI_INVALID_BIR_HANDLE для указания, что контрольные данные недоступны.

Подраздел 7.47

BioAPI_OPTIONS_MASK

#defineBioAPI_RAW (0x00000001)

Если это значение установлено, то оно указывает, что ПБУ поддерживает возврат исходных биометрических образцов и контрольных данных.

Подпункт А.4.6.2.1

Возврат исходных данных.

Функции, включающие в себя получение биометрических данных с датчика, могут дополнительно поддерживать возврат этих исходных данных для их отображения или проверки.

Если возврат исходных данных поддерживается, то выходной параметр AuditData содержит указатель этих данных. Если возврат исходных данных не поддерживается, то ПБУ возвращает значение "–1".

Ссылки: 9.3.4.1, 8.4.1.1, 7.47 и A.4.6.2.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture* с параметром AuditData, которому присвоено значение, отличающееся от NULL.

4) Вызвать функцию *BioSPI_GetHeaderFromHandle* для проверки полученного дескриптора ЗБИ контрольных данных. Если контрольные данные поддерживаются, то степень обработки контрольных данных ЗБИ должна соответствовать исходному биометрическому образцу. Если контрольные данные не поддерживаются, предполагается, что функция возвратит значение BioAPI_UNSUPPORTED_BIR_HANDLE.

5) Отсоединить и выгрузить испытуемый ПБУ.

Ожидаемые результаты: Если контрольные данные поддерживаются, то степень обработки контрольных данных ЗБИ должна соответствовать исходному биометрическому образцу. В противном случае вызов *BioSPI_Capture* возвращает особое значение дескриптора BioAPI_UNSUPPORTED_BIR_HANDLE.

Пакет языка утверждения

```
<package name="02704c50-0cd8-1085-96cb-0002a5d5fd2e">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение
        "BioSPI_Capture_AuditData" (см. далее элемент
        <description> утверждения).
    </description>

    <assertion name="BioSPI_Capture_AuditData" model="BSPTesting">
        <description>
```

Проверку данного утверждения проводят с целью испытания функции BioSPI_Capture с определенным значением параметра AuditData, отличающимся от NULL. Следующий текст соответствует приведенному в 9.3.4.1, 8.4.1.1, 7.47 и A.4.6.2.1 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_Capture
     (BioAPI_HANDLE BSPHandle,
     BioAPI_BIR_PURPOSE Purpose,
     BioAPI_BIR_SUBTYPE Subtype,
     const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
     *OutputFormat,
     BioAPI_BIR_HANDLE *CapturedBIR,
     int32_t Timeout,
     BioAPI_BIR_HANDLE *AuditData).
```

Примечание – Подробное определение функции BioAPI_Capture приведено в 8.4.1.

Подпункт 8.4.1.1:
Если AuditData имеет значение, отличающееся от NULL, может быть возвращена ЗБИ с типом «исходный биометрический образец». ПБУ может возвратить значение дескриптора BioAPI_UNSUPPORTED_BIR_HANDLE для указания, что параметр AuditData не поддерживается, или значение BioAPI_INVALID_BIR_HANDLE для указания, что контрольные данные недоступны.
Параметры: AuditData (выходной, необязательный параметр) – дескриптор ЗБИ, содержащий исходный биометрический образец. Эти данные могут быть использованы для предоставления идентификационных данных человека, использующего биометрическое устройство. Если указатель имеет входное значение NULL, то контрольные данные не были собраны. Не все ПБУ поддерживают функцию сбора контрольных данных.

ПБУ может возвратить значение дескриптора BioAPI_UNSUPPORTED_BIR_HANDLE для указания, что параметр AuditData не поддерживается, или значение BioAPI_INVALID_BIR_HANDLE для указания, что контрольные данные недоступны.

Подраздел 7.47:

BioAPI_OPTIONS_MASK

#define BioAPI_RAW (0x00000001)

Если это значение установлено, то оно указывает, что ПБУ поддерживает возврат исходных биометрических образцов и контрольных данных.

Подпункт A.4.6.2.1:

Возврат исходных данных.

Функции, включающие в себя получение биометрических данных с датчика, могут дополнительно поддерживать возврат этих исходных данных для их отображения или проверки.

Если возврат исходных данных поддерживается, то выходной параметр AuditData содержит указатель этих данных. Если возврат исходных данных не поддерживается, то ПБУ возвращает значение "-1".

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию BioSPI_Capture с параметром AuditData, которому присвоено значение, отличающееся от NULL.

4) Активизировать функцию BioSPI_GetHeaderFromHandle для проверки полученного дескриптора ЗБИ контрольных данных. Если контрольные данные поддерживаются, то степень обработки контрольных данных ЗБИ должна

```
соответствовать исходному биометрическому образцу.
Если контрольные данные не поддерживаются, то
предполагается, что функция возвращает значение
BioAPI_UNSUPPORTED_BIR_HANDLE.
5) Отсоединить и выгрузить испытуемый ПБУ.
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события
BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
<input name="_capturetimeout"/>
<!-- Указание на поддержку испытуемым ПБУ контрольных
данных.-->
<input name="_supportAuditData" />

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="CapturedBIR_AuditDataPresent">
    <input name="bspUuid" var="_bspUuid"/>
    <input name="inserttimeouttime"
    var="_inserttimeout"/>
    <input name="nosourcepresentsupported"
    var="_noSourcePresentSupported" />
```

```
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
            <input name="supportAuditData"
            var="_supportAuditData" />
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="CapturedBIR_AuditDataPresent">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>
        <input name="supportAuditData" />

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
```

```
            break_on_break="true">
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>
<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
            setvar="eventtimeoutflag">
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Испытуемый ПБУ не поддерживает уведомление о
            событии BioAPI_NOTIFY_SOURCE_PRESENT или
            уведомление о событии было получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
</assert_condition>
```

```
<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Capture со значением параметра AuditData,
отличающимся от NULL. Дескриптор полученной ЗБИ хранится
в переменной "capturedbir". -->
<invoke function="BioSPI_Capture">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <output name="CapturedBIR"
     setvar="capturedbir_handle"/>
     <output name="AuditData" setvar="auditbir_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Capture возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии «FAIL» и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии «PASS». -->
```

```
<assert_condition break_if_false="true">
     <description>
          Выходным дескриптором ПБУ AuditData является
          или достоверное значение, или значение
          BioAPI_INVALID_BIR_HANDLE (когда контрольные
          данные поддерживаются), или значение
          BioAPI_UNSUPPORTED_BIR_HANDLE (когда контрольные
          данные не поддерживаются).
     </description>
     <or>
          <and>
               <same_as var1="supportAuditData"
               value2="true" />
               <or>
                    <equal_to var1="auditbir_handle"
                         var2="__BioAPI_INVALID_BIR_HANDLE"/>
                    <greater_than_or_equal_to
                    var1="auditbir_handle" value2="0" />
               </or>
          </and>
          <and>
               <same_as var1="supportAuditData"
               value2="false" />
               <equal_to var1="auditbir_handle"
                    var2="__BioAPI_UNSUPPORTED_BIR_HANDLE" />
          </and>
     </or>
</assert_condition>

<!-- Вызов процесса для проверки степени обработки
контрольных данных ЗБИ с заданным дескриптором. -->
<invoke activity="check_auditdata_type" >
     <only_if>
          <same_as var1="supportAuditData" value2="true" />
          <greater_than_or_equal_to
          var1="auditbir_handle" value2="0" />
```

```
            </only_if>
            <input name="bsphandle" var="_bsphandle"/>
            <input name="auditbirhandle" var="auditbir_handle" />
        </invoke>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>

    <activity name="check_auditdata_type">
        <input name="bsphandle" />
        <input name="auditbirhandle" />

        <!-- Вызов функции BioSPI_GetHeaderFromHandle. -->
        <invoke function="BioSPI_GetHeaderFromHandle">
            <input name="BSPHandle" var="_bsphandle"/>
            <input name="Handle" var="auditbirhandle"/>
            <output name="ProcessedLevel"
            setvar="processedLevel"/>
            <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
                break_if_false="true">
            <description>
```

```
                Функция BioSPI_GetHeaderFromHandle возвращает
                значение BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Степень обработки контрольных данных ЗБИ
                соответствует исходному биометрическому
                образцу.
            </description>
            <equal_to var1="processedLevel"
                var2="__BioAPI_BIR_DATA_TYPE_RAW"/>
        </assert_condition>
    </activity>
</package>
```

8.29 Утверждение 9b *BioSPI_Capture_ReturnQuality*

Описание: Проверку данного утверждения проводят с целью инициализации функции *BioSPI_Capture*, а также для определения содержания в заголовке зарегистрированной ЗБИ достоверного значения качества (в диапазоне от 0 до 100).

Выдержки

Подпункт 9.3.4.1

BioAPI_RETURN BioAPI BioSPI_Capture

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_PURPOSE Purpose,

BioAPI_BIR_SUBTYPE Subtype,

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*BioAPI_BIR_HANDLE *CapturedBIR,*

int32_t Timeout,

*BioAPI_BIR_HANDLE *AuditData).*

Подпункт 8.4.1.1

Даная функция обеспечивает получение образцов для установленной цели и возвращение ЗБИ типа «промежуточный биометрический образец» (при необходимости вызова функции обработки) или «обработанный биометрический образец» (если функция обработки не требуется).

Подраздел 7.49

BioAPI_QUALITY – значение, определяющее значение качества биометрических данных в ЗБИ.

Значение качества представляет собой целое число в диапазоне от 0 до 100, кроме следующих значений:

Значение "–1": значение BioAPI_QUALITY не было установлено ПБУ (соответствующая информация должна быть приведена в документации разработчика ПБУ).

Значение "–2": параметр BioAPI_QUALITY не поддерживается ПБУ.

Подраздел 7.47

#defineBioAPI_QUALITY_INTERMEDIATE (0x00000004)

Если установлено данное значение, то ПБУ поддерживает возврат значения качества в заголовке ЗБИ для промежуточных биометрических образцов.

#defineBioAPI_QUALITY_PROCESSED (0x00000008)

Если установлено данное значение, то ПБУ поддерживает возврат значения качества в заголовке ЗБИ для обработанных биометрических образцов.

Подпункт А.4.6.2.2

Возвращаемое значение типа Quality

После получения биометрических данных с датчика, ПБУ может вычислить значение качества, относящееся к этим данным, которые он включает в заголовок возвращенной ЗБИ CapturedBIR (и дополнительно – в AuditData).

Если данный параметр поддерживается, то поле заголовка, содержащее значение качества, содержит положительноечисло в диапазоне от 1 до 100.

Если данный параметр не поддерживается, то поле содержит значение "–2". Такая ситуация может возникнуть в процессе выполнения функций *BioAPI_Capture* и *BioAPI_Enroll*.

Ссылки: 9.3.4.1, 8.4.1.1, 7.49 , 7.47 и A.4.6.2.2.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture*.

4) Вызвать функцию *BioSPI_GetHeaderFromHandle*.

5) Проверить значение качества возвращенной ЗБИ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: В заголовке полученной ЗБИ содержится достоверное значение качества в диапазоне от 0 до 100, если качество поддерживается, или "–2", если качество не поддерживается.

Пакет языка утверждения

```
<package name="03f601f0-0cd8-1085-bd59-0002a5d5fd2e">
     <author>
          ISO/IECJTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_Capture_ReturnQuality" (см. далее элемент
          <description> утверждения).
     </description>

     <assertion name="BioSPI_Capture_ReturnQuality"
     model="BSPTesting">
          <description>
```

Проверку данного утверждения проводят с целью инициализации функции BioSPI_Capture,а также для определения содержания в заголовке полученной ЗБИ достоверного значения качества (в диапазоне от 0 до 100). Следующий текст соответствует приведенному в 9.3.4.1, 8.4.1, 7.47, 7.49 и А.4.6.2.2 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_Capture
     (BioAPI_HANDLE BSPHandle,
     BioAPI_BIR_PURPOSE Purpose,
     BioAPI_BIR_SUBTYPE Subtype,
     const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
     *OutputFormat,
     BioAPI_BIR_HANDLE *CapturedBIR,
     int32_t Timeout,
     BioAPI_BIR_HANDLE *AuditData).
```

Примечание — Подробное определение функции BioAPI_Capture приведено в 8.4.1.

Подраздел 7.49

BioAPI_QUALITY - значение, определяющее значение качества биометрических данных в ЗБИ.

Значение качества представляет собой целое число в диапазоне от 0 до 100, кроме следующих значений:

Значение "-1": значение BioAPI_QUALITY не было установлено ПБУ (соответствующая информация должна быть приведена в документации разработчика ПБУ).

Значение "-2": параметр BioAPI_QUALITY не поддерживается ПБУ.

Подраздел 7.47:

```
#define BioAPI_QUALITY_INTERMEDIATE (0x00000004)
```

Если установлено данное значение, то ПБУ поддерживает возврат значения качества в заголовке ЗБИ для промежуточных биометрических образцов.

#define BioAPI_QUALITY_PROCESSED (0x00000008)

Если установлено данное значение, то ПБУ поддерживает возврат значения качества в заголовке ЗБИ для обработанных биометрических образцов.

Подпункт А.4.6.2.2:

Возвращаемое значение типа Quality.

После получения биометрических данных с датчика, ПБУ может вычислить значение качества, относящееся к этим данным, которые он включает в заголовок возвращенной ЗБИ CapturedBIR (и дополнительно – в AuditData).

Если данный параметр поддерживается, то поле заголовка, содержащее значение качества, содержит положительное число в диапазоне от 1 до 100.

Если данный параметр не поддерживается, то поле содержит значение "-2". Такая ситуация может возникнуть в процессе выполнения функций BioAPI_Capture и BioAPI_Enroll.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Capture.
4) Вызвать функцию BioSPI_GetHeaderFromHandle.
5) Проверить значение качества возвращенной ЗБИ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

</description>

```
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
<input name="_capturetimeout"/>
<!-- Указание на поддержку испытуемым ПБУ значения
качества в промежуточном биометрическом образце. -->
<input name="_intermediateQualitySupported" />

<!-- Указание на поддержку испытуемым ПБУ значения
качества в обработанном биометрическом образце. -->
<input name="_processedQualitySupported" />

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_Capture_ReturnQuality">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
```

```
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
            <input name="intermediateQualitySupported"
            var="_intermediateQualitySupported" />
            <input name="processedQualitySupported"
            var="_processedQualitySupported" />
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_Capture_ReturnQuality">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>
        <input name="intermediateQualitySupported" />
        <input name="processedQualitySupported" />

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
```

```
            break_on_break="true">
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
            setvar="eventtimeoutflag">
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается заклю-
чение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Испытуемый ПБУ не поддерживает уведомление о
            событии BioAPI_NOTIFY_SOURCE_PRESENT или
            уведомление о событии получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
```

```
</assert_condition>

<!-- ПБУ готов для регистрации. Вызвать функцию
BioSPI_Capture для создания образца. Дескриптор
полученной ЗБИ хранится в переменной "capturedbir". -->
<invoke function="BioSPI_Capture">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <input name="no_AuditData" value="true"/>
     <output name="CapturedBIR"
     setvar="capturedbir_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Capture возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetHeaderFromHandle. -->
<invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="capturedbir_handle"/>
```

```
        <output name="ProcessedLevel"
        setvar="processedLevel"/>
        <output name="Quality" setvar="quality"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
              break_if_false="true">
        <description>
              Функция BioSPI_GetHeaderFromHandle возвращает
              значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <invoke activity="check_quality_supported"
              package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
              break_on_break="true" >
        <only_if>
              <same_as var1="processedQualitySupported"
                   value2="true"/>
              <same_as var1="intermediateQualitySupported"
                   value2="true"/>
        </only_if>
        <input name="quality" var="quality"/>
    </invoke>

    <invoke activity="check_quality_ps_ins"
              break_on_break="true" >
        <only_if>
```

```
            <same_as var1="processedQualitySupported"
                value2="true" />
            <same_as var1="intermediateQualitySupported"
                value2="false" />
        </only_if>
        <input name="quality" var="quality" />
        <input name="processedLevel" var="processedLevel" />
    </invoke>

    <invoke activity="check_quality_pns_is"
            break_on_break="true" >
        <only_if>
            <same_as var1="processedQualitySupported"
                value2="false" />
            <same_as var1="intermediateQualitySupported"
                value2="true" />
        </only_if>
        <input name="quality" var="quality" />
        <input name="processedLevel" var="processedLevel" />
    </invoke>

    <invoke activity="check_quality_not_supported"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            break_on_break="true" >
        <only_if>
            <same_as var1="processedQualitySupported"
                value2="false"/>
            <same_as var1="intermediateQualitySupported"
                value2="false"/>
        </only_if>
        <input name="quality" var="quality"/>
    </invoke>

    <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
    <invoke activity="DetachAndUnload"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
```

```
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>

    <!-- Данный процесс проверяет возвращаемое значение качества.
    Процесс используется, если ПБУ поддерживает качество
    обработанных не промежуточных биометрических образцов. -->
    <activity name="check_quality_ps_ins" >
        <input name="quality" />
        <input name="processedLevel" />

        <invoke activity="check_quality_not_supported"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <only_if>
                <equal_to var1="processedLevel"
                    var2="__BioAPI_BIR_DATA_TYPE_INTERMEDIATE" />
            </only_if>
            <input name="quality" var="quality" />
        </invoke>

        <invoke activity="check_quality_supported"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <only_if>
                <equal_to var1="processedLevel"
                    var2="__BioAPI_BIR_DATA_TYPE_PROCESSED" />
            </only_if>
            <input name="quality" var="quality" />
        </invoke>
    </activity>

    <!-- Данный процесс проверяет возвращаемое значение качества.
    Процесс используется, если ПБУ поддерживает качество обрабо-
    танных не промежуточных биометрических образцов. -->
    <activity name="check_quality_pns_is" >
```

```
        <input name="quality" />
        <input name="processedLevel" />

        <invoke activity="check_quality_supported"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <only_if>
                <equal_to var1="processedLevel"
                    var2="__BioAPI_BIR_DATA_TYPE_INTERMEDIATE" />
            </only_if>
            <input name="quality" var="quality" />
        </invoke>

        <invoke activity="check_quality_not_supported"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <only_if>
                <equal_to var1="processedLevel"
                    var2="__BioAPI_BIR_DATA_TYPE_PROCESSED" />
            </only_if>
            <input name="quality" var="quality" />
        </invoke>
    </activity>
</package>
```

8.30 Утверждение 9c BioSPI_Capture_IntermediateProcessedBIR

Описание: Проверку данного утверждения проводят с целью определения, какую степень обработки (промежуточного или обработанного биометрического образца) имеет ЗБИ, возвращенная функцией *BioSPI_Capture*.

Выдержки

Подпункт 9.3.4.1

BioAPI_RETURN BioAPI BioSPI_Capture

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_PURPOSE Purpose,

BioAPI_BIR_SUBTYPE Subtype,

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*BioAPI_BIR_HANDLE *CapturedBIR,*

int32_t Timeout,

*BioAPI_BIR_HANDLE *AuditData).*

Подпункт 8.4.1.1

Даная функция обеспечивает получение образцов для установленного назначения и возвращает или ЗБИ типа «промежуточный биометрический образец» (если в дальнейшем должна быть вызвана функция обработки), или ЗБИ типа «обработанный биометрический образец» (если функция обработки не требуется).

Подпункт 8.4.1.2

CapturedBIR (выходной параметр) – дескриптор ЗБИ, содержащий полученные данные. Эти данные могут быть или промежуточным биометрическим образцом (в зависимости от назначения могут использоваться только одной из функций *Process* или *CreateTemplate*), или обработанным биометрическим образцом (в зависимости от назначения могут использоваться функциями *VerifyMatch* или *IdentifyMatch*).

Ссылки: 9.3.4.1. 8.4.1.1 и 8.4.1.2.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture*.

4) Вызвать функцию *BioSPI_GetHeaderFromHandle*.

5) Проверить степень обработки возвращенной ЗБИ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Степень обработки возвращенной ЗБИ соответствует промежуточному или обработанному биометрическому образцу.

Пакет языка утверждения

```
<package name="055ddb08-0cd6-1085-a6d3-0002a5d5fd2e">
```

```
<author>
     ISO/IEC JTC1 SC37
</author>

<description>
     Данный пакет содержит утверждение "BioSPI_Capture_ Inter-
     mediateProcessedBIR" (см. далее элемент <description> ут-
     верждения).
</description>

<assertion name="BioSPI_Capture_IntermediateProcessedBIR"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью
          определения, какую степень обработки (промежуточного
          или обработанного биометрического образца) имеет
          ЗБИ, возвращенная функцией BioSPI_Capture.
          Следующий текст соответствует приведенному в
          9.3.4.1, 8.4.1.1 и 8.4.1.2 спецификации БиоАПИ 2.0.
          ------------------------------------------------
          BioAPI_RETURN BioAPI BioSPI_Capture
               (BioAPI_HANDLE BSPHandle,
               BioAPI_BIR_PURPOSE Purpose,
               BioAPI_BIR_SUBTYPE Subtype,
               const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
               *OutputFormat,
               BioAPI_BIR_HANDLE *CapturedBIR,
               int32_t Timeout,
               BioAPI_BIR_HANDLE *AuditData).

          Примечание - Подробное определение функции
          BioAPI_Capture приведено в 8.4.1.
          ------------------------------------------------
          Подпункт 8.4.1.1:
          Даная функция обеспечивает получение образцов для
          установленного назначения и возвращает или ЗБИ типа
```

```
        «промежуточный биометрический образец» (если в
        дальнейшем должна быть вызвана функция обработки),
        или ЗБИ типа «обработанный биометрический образец»
        (если функция обработки не требуется).
        ---------------------------------------------------
        Подпункт 8.4.1.2:
        CapturedBIR (выходной параметр) - дескриптор ЗБИ,
        содержащий полученные данные. Это данные могут быть
        или промежуточным биометрическим образцом (в
        зависимости от назначения могут быть использованы
        только одной функцией Process или CreateTemplate),
        или обработанным биометрическим образцом (в
        зависимости от назначения могут быть использованы
        функциями VerifyMatch или IdentifyMatch).
        ---------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Вызвать функцию BioSPI_Capture.
        4) Вызвать функцию BioSPI_GetHeaderFromHandle.
        5) Проверить степень обработки возвращенной ЗБИ.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>
```

```
        <!-- Указание на отсутствие поддержки испытуемым ПБУ
        уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
        <input name="_noSourcePresentSupported" />

        <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
        PRESENT. -->
        <input name="_sourcepresenttimeout"/>

        <!-- Время ожидания для функции BioSPI_Capture. -->
        <input name="_capturetimeout"/>

        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="CapturedBIR_Datatype">
             <input name="bspUuid" var="_bspUuid"/>
             <input name="inserttimeouttime"
             var="_inserttimeout"/>
             <input name="nosourcepresentsupported"
             var="_noSourcePresentSupported" />
             <input name="sourcepresenttimeouttime"
             var="_sourcepresenttimeout"/>
             <input name="capturetimeouttime"
             var="_capturetimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
</assertion>

<activity name="CapturedBIR_Datatype">
```

```
<input name="bspUuid"/>
<input name="inserttimeouttime"/>
<input name="nosourcepresentsupported" />
<input name="sourcepresenttimeouttime"/>
<input name="capturetimeouttime"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->

<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
```

```
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию BioSPI_
Capture для создания образца. Дескриптор полученной ЗБИ
хранится в переменной "capturedbir". -->
<invoke function="BioSPI_Capture">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
          var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0" />
     <input name="Timeout" var="capturetimeouttime"/>
     <input name="no_AuditData" value="true"/>
     <output name="CapturedBIR"
     setvar="capturedbir_handle"/>
     <return setvar="return"/>
</invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Capture возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызов функции BioSPI_GetHeaderFromHandle. -->
<invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="capturedbir_handle"/>
     <output name="ProcessedLevel"
     setvar="processedLevel"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_GetHeaderFromHandle возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
```

```
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Степень обработки возвращенной ЗБИ соответствует
                промежуточному или обработанному биометрическому
                образцу.
            </description>
            <or>
                <equal_to var1="processedLevel"
                    var2="__BioAPI_BIR_DATA_TYPE_INTERMEDIATE"/>
                <equal_to var1="processedLevel"
                    var2="__BioAPI_BIR_DATA_TYPE_PROCESSED"/>
            </or>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.31 Утверждение 9d *BioSPI_Capture_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью определения возвращения ошибки BioAPIERR_INVALID_BSP_HANDLE при вызове функции *BioSPI_Capture* с недостоверным дескриптором ПБУ.

Выдержки

Подпункт 9.3.4.1

BioAPI_RETURN BioAPI BioSPI_Capture

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_PURPOSE Purpose,

BioAPI_BIR_SUBTYPE Subtype,

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*BioAPI_BIR_HANDLE *CapturedBIR,*

int32_t Timeout,

*BioAPI_BIR_HANDLE *AuditData).*

Подпункт 8.4.1.2

Параметры: BSPHandle (входной параметр) – дескриптор присоединенного ПБУ.

Ссылки: 9.3.4.1 и 8.4.1.2.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture* с недостоверным дескриптором ПБУ.

4) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_INVALID_BSP_HANDLE.

5) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_Capture* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="0244f2a8-0cf2-1085-8443-0002a5d5fd2e">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
```

Данный пакет содержит утверждение "BioSPI_Capture_
InvalidBSPHandle" (см. далее элемент <description>
утверждения).
</description>

<assertion name="BioSPI_Capture_InvalidBSPHandle"
model="BSPTesting">
<description>
Проверку данного утверждения проводят с целью
определения возвращения ошибки BioAPIERR_INVALID_
BSP_HANDLE при вызове функции BioSPI_Capture с
недостоверным дескриптором ПБУ.
Приведенный ниже текст соответствует приведенному в
9.3.4.1 спецификации БиоАПИ 2.0.

BioAPI_RETURN BioAPI BioSPI_Capture
(BioAPI_HANDLE BSPHandle,
BioAPI_BIR_PURPOSE Purpose,
BioAPI_BIR_SUBTYPE Subtype,
const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
*OutputFormat,
BioAPI_BIR_HANDLE *CapturedBIR,
int32_t Timeout,
BioAPI_BIR_HANDLE *AuditData).

Примечание - Подробное определение функции
BioAPI_Capture приведено в 8.4.1.

Подпункт 8.4.1.2:
Параметры: BSPHandle (входной параметр) - дескриптор
присоединенного ПБУ.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.

```
        3) Вызвать функцию BioSPI_Capture с недостоверным
        дескриптором ПБУ.
        4) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPIERR_INVALID_BSP_HANDLE.
        5) Отсоединить и выгрузить испытуемый ПБУ.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>

    <!-- Указание на отсутствие поддержки испытуемым ПБУ
    уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
    <input name="_noSourcePresentSupported" />

    <!-- Время ожидания для события
    BioAPI_NOTIFY_SOURCE_PRESENT. -->
    <input name="_sourcepresenttimeout"/>

    <!-- Время ожидания для функции BioSPI_Capture. -->
    <input name="_capturetimeout"/>

    <!--Вызов основного процесса данного утверждения со
    значениями входных параметров, определенными значениями
    параметров утверждения. -->
    <invoke activity="BioSPI_Capture_InvalidBSPHandle">
        <input name="bspUuid" var="_bspUuid"/>
        <input name="inserttimeouttime"
        var="_inserttimeout"/>
```

```
            <input name="nosourcepresentsupported"
            var="_noSourcePresentSupported" />
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_Capture_InvalidBSPHandle">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
```

```
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
    </invoke>

    <set name="eventtimeoutflag" value="false"/>

    <!-- Если испытуемый ПБУ поддерживает уведомление о событии
    BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
    уведомления в пределах указанной максимальной
    продолжительности. -->
    <wait_until timeout_var="sourcepresenttimeouttime"
              setvar="eventtimeoutflag">
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent" />
    </wait_until>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
              break_if_false="true">
        <description>
            Испытуемый ПБУ не поддерживает уведомление о
            событии BioAPI_NOTIFY_SOURCE_PRESENT или
            уведомление о событии получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
    </assert_condition>
```

```
        <!-- ПБУ готов к получению ЗБИ. Вызвать функцию
        BioSPI_Capture для создания образца. Дескриптор
        полученной ЗБИ хранится в переменной "capturedbir". -->
        <invoke function="BioSPI_Capture">
             <input name="BSPHandle" value="0"/>
             <input name="Purpose"
                  var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
             <input name="Subtype" value="0" />
             <input name="Timeout" var="capturetimeouttime"/>
             <input name="no_AuditData" value="true"/>
             <output name="CapturedBIR" set-
             var="capturedbir_handle"/>
             <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
             <description>
                  Функция BioSPI_Capture возвращает значение
                  BioAPIERR_INVALID_BSP_HANDLE.
             </description>
             <equal_to var1="return"
                  var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                  package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
             <input name="bspUuid" var="bspUuid" />
             <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.32 Утверждение 10а *BioSPI_CreateTemplate_PayloadSupported*

Описание: Проверку данного утверждения проводят с целью испытания функции *BioSPI_CreateTemplate* с достоверными параметрами и полезной информацией.

Выдержки

Подпункт 9.3.4.2

BioAPI_RETURN BioAPI BioSPI_CreateTemplate
(BioAPI_HANDLE BSPHandle,
*const BioAPI_INPUT_BIR *CapturedBIR,*
*const BioAPI_INPUT_BIR *ReferenceTemplate,*
*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*
*BioAPI_BIR_HANDLE *NewTemplate,*
*const BioAPI_DATA *Payload,*
*BioAPI_UUID *TemplateUUID).*

Подпункт 8.4.5.1

Если Payload (полезная информация) относится к ReferenceTemplate (контрольному шаблону), то она может быть возвращена при условии успешной верификации, если FMRAchieved (вероятность ошибки ложного совпадения) имеет допустимое значение. Данный процесс управляется политикой ПБУ и должен быть указан в его схеме.

Подраздел 7.47

#define BioAPI_PAYLOAD (0x00000080)

Если данное значение установлено, то ПБУ поддерживает перенос полезной информации (сохраняет полезную информацию при выполнении функций *BioAPI_Enroll* или *BioAPI_CreateTemplate* и возвращает ее при успешном выполнении функций *BioAPI_Verify* или *BioAPI_VerifyMatch*).

Подпункт А.4.6.2.6

Если функция предусмотрена, то ПБУ должен ввести в реестр компонентов максимальный размер поля для полезной информации. Если максимальный размер поля равен нулю, это указывает на то, что перенос полезной информации

не поддерживается. Если размер входной полезной информации превышает установленный размер, то функция должна возвратить код ошибки.

Ссылки: 9.3.4.2, 8.4.5.1, 7.47 и А.4.6.2.6.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture* с назначением BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY для получения ЗБИ для регистрации.

4) Вызвать функцию *BioSPI_CreateTemplate* с предоставлением полезной информации.

5) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

6) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_CreateTemplate* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="04a01118-0cf9-1085-96d4-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_CreateTemplate_PayloadSupported" (см. далее
          элемент <description> утверждения).
     </description>

     <assertion name="BioSPI_CreateTemplate_PayloadSupported"
     model="BSPTesting">
```

<description>

Проверку данного утверждения проводят с целью испытания функции BioSPI_CreateTemplate с достоверными параметрами и полезной информацией. Следующий текст соответствует приведенному в 9.3.4.2, A.4.6.2.6, C.3.4.4 и 7.47 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_CreateTemplate
    (BioAPI_HANDLE BSPHandle,
    const BioAPI_INPUT_BIR *CapturedBIR,
    const BioAPI_INPUT_BIR *ReferenceTemplate,
    const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
    *OutputFormat,
    BioAPI_BIR_HANDLE *NewTemplate,
    const BioAPI_DATA *Payload,
    BioAPI_UUID *TemplateUUID).
```

Примечание – Подробное определение функции BioAPI_CreateTemplate приведено в 8.4.2.

Подпункт A.4.6.2.6

Если функция предусмотрена, то ПБУ должен ввести в реестр компонентов максимальный размер поля для полезной информации. Если максимальный размер равен нулю, это указывает на то, что перенос полезной информации не поддерживается. Если размер входной полезной информации превышает установленный размер, то функция должна возвратить код ошибки.

Подраздел 7.47:

```
#define BioAPI_PAYLOAD (0x00000080)
```

Если данное значение установлено, то ПБУ поддерживает перенос полезной информации (сохраняет полезную информацию при выполнении функций BioAPI_Enroll или

```
      BioAPI_CreateTemplate и возвращает ее при успешном
      выполнении функций BioAPI_Verify или BioAPI_VerifyMatch).

      Порядок действий при испытании на соответствие:

      1) Загрузить испытуемый ПБУ.
      2) Присоединить испытуемый ПБУ.
      3) Вызвать функцию BioSPI_Capture с назначением
      BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY для
      получения ЗБИ для регистрации.
      4) Вызвать функцию BioSPI_CreateTemplate с
      предоставлением полезной информации.
      5) Проверить возвращаемый код. Предполагается
      возвращение значения BioAPI_OK.
      6) Отсоединить и выгрузить испытуемый ПБУ.

      Если какие-либо промежуточные операции прошли
      неуспешно, то выдается заключение о соответствии
      "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события
BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
```

```
        <input name="_capturetimeout"/>

        <!-- Указание на поддержку испытуемым ПБУ полезной
        информации.-->
        <input name="_supportPayload" />
        <!-- Размер входной полезной информации не должен превышать
        максимальный размер поля, установленный для полезной
        информации, введенный ПБУ в реестр компонентов. -->
        <input name="_payload" />

        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_CreateTemplate_PayloadSupported">
             <input name="bspUuid" var="_bspUuid"/>
             <input name="inserttimeouttime"
             var="_inserttimeout"/>
             <input name="nosourcepresentsupported"
             var="_noSourcePresentSupported" />
             <input name="sourcepresenttimeouttime"
             var="_sourcepresenttimeout"/>
             <input name="capturetimeouttime"
             var="_capturetimeout"/>
             <input name="supportpayload" var="_supportPayload" />
             <input name="payload" var="_payload" />
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
    </assertion>
```

```
<activity name="BioSPI_CreateTemplate_PayloadSupported">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported" />
     <input name="sourcepresenttimeouttime"/>
     <input name="capturetimeouttime"/>
     <input name="supportpayload" />
     <input name="payload" />

     <!-- Данное утверждение использует BSPHandle со значением
     "1" для всех вызовов BioSPI, которые этого требуют. -->
     <set name="_bsphandle" value="1"/>

     <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
     предоставленных испытуемым ПБУ. Входное значение параметра
     "unitIDOrNull" равно "0", поэтому утверждение будет
     тестировать модуль датчика, выбранный ПБУ. -->
     <invoke activity="LoadAndAttach"
               package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
               break_on_break="true">
          <input name="bspUuid" var="bspUuid"/>
          <input name="bspVersion" value="32"/>
          <input name="unitIDOrNull" value="0"/>
          <input name="bspHandle" var="_bsphandle"/>
          <input name="eventtimeouttime"
          var="inserttimeouttime"/>
     </invoke>

     <set name="eventtimeoutflag" value="false"/>

     <!-- Если испытуемый ПБУ поддерживает уведомление о событии
     BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
     уведомления в пределах указанной максимальной
     продолжительности. -->
     <wait_until timeout_var="sourcepresenttimeouttime"
```

```
            setvar="eventtimeoutflag">
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent" />
    </wait_until>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Испытуемый ПБУ не поддерживает уведомление о
            событии BioAPI_NOTIFY_SOURCE_PRESENT или
            уведомление о событии получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
    </assert_condition>

    <!-- ПБУ готов к получению ЗБИ. Вызвать функцию
    BioSPI_Capture для создания образца. Дескриптор
    полученного ЗБИ хранится в переменной "capturedbir". -->
    <invoke function="BioSPI_Capture">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Purpose"
        var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <input name="no_AuditData" value="true"/>
        <output name="CapturedBIR"
        setvar="capturedbir_handle"/>
        <return setvar="return"/>
    </invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Capture возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Если степень обработки полученной ЗБИ соответствует
обработанному биометрическому образцу, то выполнение
процесса прерывается и выдается заключение о соответствии
"UNDECIDED". -->
<invoke activity="check_capturedBIR_datatype"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="BIRHandle" var="capturedbir_handle"/>
</invoke>

<!-- Вызов функции BioSPI_CreateTemplate. -->
<invoke function="BioSPI_CreateTemplate">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="CapturedBIR_Form"
     var="__BioAPI_BIR_HANDLE_INPUT" />
     <input name="CapturedBIR_BIRHandle"
     var="capturedbir_handle"/>
     <input name="Payload" var="payload" />
     <output name="NewTemplate"
     setvar="newTemplate_handle"/>
```

```
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "FAIL", в противном
    случае выдается заключение о соответствии "PASS". -->
    <assert_condition>
        <description>
            Функция BioSPI_CreateTemplate возвращает
            действительное значение, на основании которого
            можно определить, поддерживает ПБУ полезную
            информацию или нет.
        </description>
        <or>
            <and>
                <equal_to var1="return"
                var2="__BioAPI_OK"/>
                <same_as var1="supportpayload"
                value2="true"/>
            </and>
            <and>
                <equal_to var1="return"
                    var2="__BioAPIERR_BSP_UNABLE_TO_STORE
                    _PAYLOAD" />
                <same_as var1="supportpayload"
                value2="false"/>
            </and>
        </or>
    </assert_condition>

    <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
    <invoke activity="DetachAndUnload"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
        <input name="bspUuid" var="bspUuid" />
```

```
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.33 Утверждение 10b *BioSPI_CreateTemplate_BIRHeaderQuality*

Описание: Проверку данного утверждения проводят с целью испытания функции *BioSPI_CreateTemplate* с достоверными параметрами и возвращенным значением качества.

Выдержки

Подпункт 9.3.4.2

BioAPI_RETURN BioAPI BioSPI_CreateTemplate

(BioAPI_HANDLE BSPHandle,

*const BioAPI_INPUT_BIR *CapturedBIR,*

*const BioAPI_INPUT_BIR *ReferenceTemplate,*

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*BioAPI_BIR_HANDLE *NewTemplate,*

*const BioAPI_DATA *Payload,*

*BioAPI_UUID *TemplateUUID).*

Пункт 7.49.3

Значение качества представляет собой целое число в диапазоне от 0 до 100, кроме следующих значений:

Значение "–1": значение BioAPI_QUALITY не было установлено ПБУ (соответствующая информация должна быть приведена в документации разработчика ПБУ).

Значение "–2": параметр BioAPI_QUALITY не поддерживается ПБУ.

Подраздел 7.47

#define BioAPI_QUALITY_PROCESSED (0x00000008)

Если данное значение установлено, то ПБУ поддерживает возврат значения качества (в заголовке ЗБИ) для обработанных биометрических данных.

Подпункт А.4.6.2.2

После обработки ЗБИ может выполняться повторное вычисление значения качества, которое может быть включено в заголовок processedBIR (вновь созданная обработанная ЗБИ) и, кроме того, в заголовок AdaptedBIR (адаптированная ЗБИ). Это происходит в процессе выполнения функций *BioAPI_CreateTemplate*, *BioAPI_Process*, *BioAPI_Verify*, *BioAPI_VerifyMatch*, *BioAPI_Enroll*, и *BioAPI_Import* (ConstructedBIR – ЗБИ, созданная из импортированных биометрических данных).

ПБУ должен ввести в реестр компонентов информацию о том, поддерживает ли он вычисление качества для ЗБИ каждого типа: исходной, промежуточной и обработанной.

Ссылки: 9.3.4.2, 7.49.3, 7.47 и A.4.6.2.2.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture* для получения ЗБИ.

4) Вызвать функцию *BioSPI_CreateTemplate*.

5) Вызвать функцию *BioSPI_GetHeaderFromHandle* для NewTemplate (нового шаблона) и проверить значение поля Quality. Предполагается получение значения в диапазоне от 0 до 100.

6) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: В заголовке полученной ЗБИ содержится достоверное значение качества (в диапазоне от 0 до 100).

Пакет языка утверждения

```
<package name="00b5c728-0cfb-1085-8969-0002a5d5fd2e">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
```

```
     Данный пакет содержит утверждение
     "BioSPI_CreateTemplate_BIRHeaderQuality" (см. далее
     элемент <description> утверждения).
</description>

<assertion name="BioSPI_CreateTemplate_BIRHeaderQuality"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью
          испытания функции BioSPI_CreateTemplate с достоверными
          параметрами и возвращенным значением качества.
          Следующий текст соответствует приведенному в 9.3.4.2,
          7.49.3, 7.47 и A.4.6.2.2 спецификации БиоАПИ 2.0.
          ------------------------------------------------
          BioAPI_RETURN BioAPI BioSPI_CreateTemplate
               (BioAPI_HANDLE BSPHandle,
               const BioAPI_INPUT_BIR *CapturedBIR,
               const BioAPI_INPUT_BIR *ReferenceTemplate,
               const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
               *OutputFormat,
               BioAPI_BIR_HANDLE *NewTemplate,
               const BioAPI_DATA *Payload,
               BioAPI_UUID *TemplateUUID).

          Примечание - Подробное определение функции
          BioAPI_CreateTemplate приведено в 8.4.2.
          ------------------------------------------------
          Пункт 7.49.3:
          Значение качества представляет собой целое число в
          диапазоне от 0 до 100, кроме следующих значений:
          Значение "-1": значение BioAPI_QUALITY не было
          установлено ПБУ (соответствующая информация должна
          быть приведена в документации разработчика ПБУ).
          Значение "-2": параметр BioAPI_QUALITY
          не поддерживается ПБУ.
```

Подраздел 7.47:

#define BioAPI_QUALITY_PROCESSED (0x00000008)

Если данное значение установлено, то ПБУ поддерживает возврат значения качества (в заголовке ЗБИ) для обработанных биометрических данных.

Подпункт A.4.6.2.2:

После обработки ЗБИ может выполняться повторное вычисление значения качества, которое может быть включено в заголовок processedBIR (вновь созданная обработанная ЗБИ) и,кроме того, в заголовок AdaptedBIR (адаптированная ЗБИ).

Это происходит в процессе выполнения функций BioAPI_CreateTemplate, BioAPI_Process, BioAPI_Verify, BioAPI_VerifyMatch, BioAPI_Enroll, и BioAPI_Import (ConstructedBIR - ЗБИ, созданная из импортированных биометрических данных).

ПБУ должен ввести в реестр компонентов информацию о том, поддерживает ли он вычисление качества для ЗБИ каждого типа: исходной, промежуточной и обработанной.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Capture для получения ЗБИ.
4) Вызвать функцию BioSPI_CreateTemplate.
5) Вызвать функцию BioSPI_GetHeaderFromHandle для NewTemplate (новогошаблона), проверить значение поля Quality. Предполагается получение значения в диапазоне от 0 до 100.
6) Отсоединить и выгрузить испытуемый ПБУ.

```
      Если какие-либо промежуточные операции прошли
      неуспешно, то выдается заключение о соответствии
      "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
<input name="_capturetimeout"/>
<!-- Указание на поддержку испытуемым ПБУ возвращения
значения качества в обработанной ЗБИ. -->
<input name="_processedQualitySupported"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_CreateTemplate_ReturnQuality">
      <input name="bspUuid" var="_bspUuid"/>
      <input name="inserttimeouttime"
      var="_inserttimeout"/>
      <input name="nosourcepresentsupported"
      var="_noSourcePresentSupported"/>
      <input name="sourcepresenttimeouttime"
      var="_sourcepresenttimeout"/>
```

```
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
            <input name="processedQualitySupported"
            var="_processedQualitySupported"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_CreateTemplate_ReturnQuality">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported"/>
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>
        <input name="processedQualitySupported"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
```

```
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
             setvar="eventtimeoutflag">
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent"/>
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
             break_if_false="true">
        <description>
             Испытуемый ПБУ не поддерживает уведомление о
             событии BioAPI_NOTIFY_SOURCE_PRESENT или
             уведомление о событии получено в пределах
             указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
</assert_condition>
```

```
<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Capture для создания шаблона. Дескриптор
полученной ЗБИ хранится в переменной "capturedbir". -->
<invoke function="BioSPI_Capture">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <input name="no_AuditData" value="true"/>
     <output name="CapturedBIR"
     setvar="capturedbir_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Capture возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Если степень обработки полученной ЗБИ соответствует
обработанному биометрическому образцу, то выдается
заключение о соответствии «UNDECIDED» и выполнение
процесса прерывается. -->
<invoke activity="check_capturedBIR_datatype"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
```

```
          break_on_break="true" >
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="BIRHandle" var="capturedbir_handle"/>
</invoke>

<!-- Вызов функции BioSPI_CreateTemplate. -->
<invoke function="BioSPI_CreateTemplate">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="CapturedBIR_Form"
     var="__BioAPI_BIR_HANDLE_INPUT"/>
     <input name="CapturedBIR_BIRHandle"
     var="capturedbir_handle"/>
     <output name="NewTemplate"
     setvar="newTemplate_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_CreateTemplate возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetHeaderFromHandle с
предоставлением созданного шаблона ЗБИ. -->
<invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="newTemplate_handle"/>
```

```
        <output name="Quality" setvar="quality"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_GetHeaderFromHandle возвращает
            значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <invoke activity="check_quality_supported"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            break_on_break="true">
        <only_if>
            <same_as var1="processedQualitySupported"
            value2="true"/>
        </only_if>
        <input name="quality" var="quality"/>
    </invoke>

    <invoke activity="check_quality_not_supported"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            break_on_break="true">
        <only_if>
            <same_as var1="processedQualitySupported"
            value2="false"/>
        </only_if>
```

```
            <input name="quality" var="quality"/>
        </invoke>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid"/>
            <input name="BSPHandle" var="_bsphandle"/>
        </invoke>
    </activity>
</package>
```

8.34 Утверждение 10с *BioSPI_CreateTemplate_OutputBIRDataType*

Описание: Проверку данного утверждения проводят с целью проверки функции *BioSPI_CreateTemplate* с достоверными параметрами. Предполагается, что вновь созданный шаблон ЗБИ является обработанным биометрическим образцом.

Выдержки

Подпункт 9.3.4.2

BioAPI_RETURN BioAPI BioSPI_CreateTemplate

(BioAPI_HANDLE BSPHandle,

*const BioAPI_INPUT_BIR *CapturedBIR,*

*const BioAPI_INPUT_BIR *ReferenceTemplate,*

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*BioAPI_BIR_HANDLE *NewTemplate,*

*const BioAPI_DATA *Payload,*

*BioAPI_UUID *TemplateUUID).*

Подпункт 8.4.2.1

Данная функция принимает в качестве входного параметра ЗБИ, содержащую промежуточной биометрический образец, с целью создания нового шаблона регистрации.

Ссылки: 9.3.4.2 и 8.4.2.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture* с назначением BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY для получения ЗБИ для регистрации.

4) Вызвать функцию *BioSPI_CreateTemplate* без StoredTemplate (сохраненного шаблона) и Payload (полезной информации) со значением "0".

5) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

6) Вызвать функцию *BioSPI_GetHeaderFromHandle* с новым шаблоном ЗБИ, при этом предполагают, что степень обработки ЗБИ соотвествует обработанному биометрическому образцу.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Степень обработки нового шаблона ЗБИ соответствует обработанному биометрическому образцу.

Пакет языка утверждения

```
<package name="0193c730-0cf9-1085-b0a3-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_CreateTemplate_OutputBIRDataType" (см. далее
          элемент <description> утверждения).
     </description>

     <assertion name="BioSPI_CreateTemplate_OutputBIRDataType"
     model="BSPTesting">
          <description>
```

Проверку данного утверждения проводят с целью проверки функции BioSPI_CreateTemplate с достоверными параметрами. Предполагается, что новый шаблон ЗБИ является обработанным биометрическим образцом. Следующий текст соответствует приведенному в 9.3.4.2 и 8.4.2.1 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_CreateTemplate
     (BioAPI_HANDLE BSPHandle,
     const BioAPI_INPUT_BIR *CapturedBIR,
     const BioAPI_INPUT_BIR *ReferenceTemplate,
     const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
     *OutputFormat,
     BioAPI_BIR_HANDLE *NewTemplate,
     const BioAPI_DATA *Payload,
     BioAPI_UUID *TemplateUUID).
```

Примечание — Подробное определение функции BioAPI_CreateTemplate приведено в 8.4.2.

Подпункт 8.4.2.1:

Данная функция принимает в качестве входного параметра ЗБИ, содержащую промежуточной биометрический образец, с целью создания нового шаблона регистрации.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Capture с назначением BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY для получения ЗБИ для регистрации.
4) Вызвать функцию BioSPI_CreateTemplate без StoredTemplate (сохраненного шаблона) и Payload (полезной информации) со значением "0".

```
        5) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPI_OK.
        6) Вызвать функцию BioSPI_GetHeaderFromHandle с новым
        шаблоном ЗБИ, при этом предполагают, что степень
        обработки ЗБИ соотвествует обработанному
        биометрическому образцу.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT.-->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
<input name="_capturetimeout"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_CreateTemplate">
        <input name="bspUuid" var="_bspUuid"/>
```

```
            <input name="inserttimeouttime"
            var="_inserttimeout"/>
            <input name="nosourcepresentsupported"
            var="_noSourcePresentSupported"/>
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_CreateTemplate">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported"/>
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
```

```
            break_on_break="true">
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддеривает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
            setvar="eventtimeoutflag">
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent"/>
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Испытуемый ПБУ не поддерживает уведомление о
            событии BioAPI_NOTIFY_SOURCE_PRESENT или
            уведомление о событии получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
```

```
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Capture для создания шаблона. Дескриптор созданного
ЗБИ хранится в перменной "capturedbir". -->
<invoke function="BioSPI_Capture">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <input name="no_AuditData" value="true"/>
     <output name="CapturedBIR"
     setvar="capturedbir_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Capture возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Если степень обработки полученной ЗБИ соответствует
обработанному биометрическому образцу, то выдается
заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается. -->
<invoke activity="check_capturedBIR_datatype"
```

```
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            break_on_break="true" >
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="BIRHandle" var="capturedbir_handle"/>
    </invoke>

    <!-- Вызов функции BioSPI_CreateTemplate. -->
    <invoke function="BioSPI_CreateTemplate">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="CapturedBIR_Form"
        var="__BioAPI_BIR_HANDLE_INPUT" />
        <input name="CapturedBIR_BIRHandle"
        var="capturedbir_handle"/>
        <output name="NewTemplate"
        setvar="newTemplate_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_CreateTemplate возвращает
            значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Вызов функции BioSPI_GetHeaderFromHandle с
    предоставлением вновь созданного шаблона ЗБИ. -->
    <invoke function="BioSPI_GetHeaderFromHandle">
```

```
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="newTemplate_handle"/>
     <output name="ProcessedLevel"
     setvar="processedLevel"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается заклю-
чение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_GetHeaderFromHandle возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
     <description>
          Степень обработки вновь созданного шаблона ЗБИ
          соответствует обработанному биометрическому
          образцу.
     </description>
     <equal_to var1="processedLevel"
          var2="__BioAPI_BIR_DATA_TYPE_PROCESSED" />
</assert_condition>
```

```
        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPeUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.35 Утверждение 10d *BioSPI_CreateTemplate_OutputBIRPurpose*

Описание: Проверку данного утверждения проводят с целью определения совпадения назначения созданного шаблона с назначением полученной ЗБИ.

Выдержки

Подпункт 9.3.4.2

BioAPI_RETURN BioAPI BioSPI_CreateTemplate
(BioAPI_HANDLE BSPHandle,
*const BioAPI_INPUT_BIR *CapturedBIR,*
*const BioAPI_INPUT_BIR *ReferenceTemplate,*
*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*
*BioAPI_BIR_HANDLE *NewTemplate,*
*const BioAPI_DATA *Payload,*
*BioAPI_UUID *TemplateUUID).*

Подпункт 8.4.2.1

Данная функция принимает в качестве входного параметра ЗБИ, содержащую промежуточной биометрический образец, с целью создания нового шаблона регистрации.

Ссылки: 9.3.4.2 и 8.4.2.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture* с назначением BioAPI_PURPOSE_

ENROLL_FOR_VERIFICATION_ONLY для получения ЗБИ для регистрации.

4) Вызвать функцию *BioSPI_CreateTemplate* без StoredTemplate (сохраненного шаблона) и Payload (полезной информации) со значением "0".

5) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

6) Вызвать функцию *BioSPI_GetHeaderFromHandle* для зарегистрированной ЗБИ и полученного NewTemplate (нового шаблона) с целью их сравнения друг с другом. Предполагается, что они имеют одно и то же назначение BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY.

Утверждение ожидает уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT (если ПБУ заявляет о поддержке данного события).

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Назначение нового шаблона аналогично назначению полученной ЗБИ.

Пакет языка утверждения

```
<package name="03dbdaa0-0cf2-1085-99ed-0002a5d5fd2e">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_CreateTemplate_OutputBIRPurpose" (см. далее
          элемент <description> утверждения).
     </description>

     <assertion name="BioSPI_CreateTemplate_OutputBIRPurpose"
     model="BSPTesting">
          <description>
```

Проверку данного утверждения проводят с целью определения совпадения назначения созданного шаблона с назначением полученной ЗБИ.

Следующий текст соответствует приведенному в 9.3.4.2 и 8.4.2.1 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_CreateTemplate
     (BioAPI_HANDLE BSPHandle,
     const BioAPI_INPUT_BIR *CapturedBIR,
     const BioAPI_INPUT_BIR *ReferenceTemplate,
     const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
     *OutputFormat,
     BioAPI_BIR_HANDLE *NewTemplate,
     const BioAPI_DATA *Payload,
     BioAPI_UUID *TemplateUUID).
```

Примечание — Подробное определение функции BioAPI_CreateTemplate приведено в 8.4.2.

Подпункт 8.4.2.1:

Данная функция принимает в качестве входного параметра ЗБИ, содержащую промежуточной биометрический образец, с целью создания нового шаблона регистрации.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Capture с назначением BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY для получения ЗБИ для регистрации.
4) Вызвать функцию BioSPI_CreateTemplate без StoredTemplate (сохраненного шаблона) и Payload (полезной информации) со значением "0".

```
      5) Проверить возвращаемый код. Предполагается
      возвращение значения BioAPI_OK.
      6) Вызвать функцию BioSPI_GetHeaderFromHandle для
      зарегистрированной ЗБИ и полученного NewTemplate
      (нового шаблона) для их сравнения друг с другом.
      Предполагается, что они имеют одно и то же назначение
      BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY.
      Утверждение ожидает уведомления о событии BioAPI_
      NOTIFY_SOURCE_PRESENT (если ПБУ поддерживает данное
      событие).

      Если какие-либо промежуточные операции прошли
      неуспешно, то выдается заключение о соответствии
      "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_
SOURCE_PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
<input name="_capturetimeout"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
```

```
        <invoke activity="BioSPI_CreateTemplate">
            <input name="bspUuid" var="_bspUuid"/>
            <input name="inserttimeouttime"
            var="_inserttimeout"/>
            <input name="nosourcepresentsupported"
            var="_noSourcePresentSupported"/>
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_CreateTemplate">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported"/>
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
```

```
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent"/>
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
```

```
            уведомление о событии получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Capture для создания шаблона. Дескриптор созданной
ЗБИ хранится в переменной "capturedbir". -->
<invoke function="BioSPI_Capture">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Purpose"
        var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <input name="no_AuditData" value="true"/>
        <output name="CapturedBIR"
        setvar="capturedbir_handle"/>
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_Capture возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
```

```
<!-- Если степень обработки полученной ЗБИ соответствует
обработанному биометрическому образцу, то выдается
заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается. -->
<invoke activity="check_capturedBIR_datatype"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true" >
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="BIRHandle" var="capturedbir_handle"/>
</invoke>

<!-- Вызов функции BioSPI_CreateTemplate. -->
<invoke function="BioSPI_CreateTemplate">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="CapturedBIR_Form"
     var="__BioAPI_BIR_HANDLE_INPUT"/>
     <input name="CapturedBIR_BIRHandle"
     var="capturedbir_handle"/>
     <output name="NewTemplate"
     setvar="newTemplate_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_CreateTemplate возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
```

```
<!-- Вызов функции BioSPI_GetHeaderFromHandle для
получения ЗБИ. -->
<invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="capturedbir_handle"/>
     <output name="Purpose" setvar="purpose1"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_GetHeaderFromHandle возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_GetHeaderFromHandle для
созданного шаблона ЗБИ. -->
<invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="newTemplate_handle"/>
     <output name="Purpose" setvar="purpose2"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
```

```
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_GetHeaderFromHandle возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
     <description>
          Назначение созданного шаблона аналогично
          назначению полученной ЗБИ.
     </description>
     <equal_to var1="purpose1" var2="purpose2"/>
</assert_condition>

<!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
<invoke activity="DetachAndUnload"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
     <input name="bspUuid" var="bspUuid"/>
     <input name="BSPHandle" var="_bsphandle"/>
</invoke>
</activity>
</package>
```

8.36 Утверждение 10e *BioSPI_CreateTemplate_InputBIRDataType*

Описание: Проверку данного утверждения проводят с целью испытания функции *BioSPI_CreateTemplate* с предоставлением в качестве входного параметра ЗБИ, имеющей недостоверную степень обработки.

Выдержки

Подпункт 9.3.4.2

BioAPI_RETURN BioAPI BioSPI_CreateTemplate
(BioAPI_HANDLE BSPHandle,
*const BioAPI_INPUT_BIR *CapturedBIR,*
*const BioAPI_INPUT_BIR *ReferenceTemplate,*
*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*
*BioAPI_BIR_HANDLE *NewTemplate,*
*const BioAPI_DATA *Payload,*
*BioAPI_UUID *TemplateUUID).*

Подпункт 8.4.2.1

Данная функция принимает ЗБИ, содержащую биометрические данные в промежуточной форме, для создания нового шаблона регистрации. Новую ЗБИ создают с помощью CapturedBIR. Также дополнительно функция может выполнить адаптацию нового шаблона на основе существующего ReferenceTemplate.

Ссылки: 9.3.4.2 и 8.4.2.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll* с назначением BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY для получения обработанного шаблона.

4) Вызвать функцию *BioSPI_GetBIRHeaderFromHandle*.

5) Проверить, что степень обработки ЗБИ соответствует обработанному биометрическому образцу.

6) Вызвать функцию *BioSPI_CreateTemplate* с обработанной ЗБИ в качестве входного параметра.

7) Проверить, отличается ли возвращаемое значение от значения BioAPI_OK.

8) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_CreateTemplate* завершается ошибкой.

Пакет языка утверждения

```
<package name="6d543ea0-2ce9-11d9-9669-0800200c9a66">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение "BioSPI_CreateTemplate_
        InputBIRDataType" (см. далее элемент <description>
        утверждения).
    </description>

    <assertion name="BioSPI_CreateTemplate_InputBIRDataType"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью
            испытания функции BioSPI_CreateTemplate с
            предоставлением в качестве входного параметра ЗБИ,
            имеющей недостоверную степень обработки.
            Следующий текст соответствует приведенному в 9.3.4.2
            и 8.4.2.1 спецификации БиоАПИ 2.0.
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_CreateTemplate
                (BioAPI_HANDLE BSPHandle,
                const BioAPI_INPUT_BIR *CapturedBIR,
```

```
    const BioAPI_INPUT_BIR *ReferenceTemplate,
    const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
    *OutputFormat,
    BioAPI_BIR_HANDLE *NewTemplate,
    const BioAPI_DATA *Payload,
    BioAPI_UUID *TemplateUUID).
```

Примечание - Подробное определение функции BioAPI_CreateTemplate приведено в 8.4.2.

Подпункт 8.4.2.1:
Данная функция принимает ЗБИ, содержащую биометрические данные в промежуточной форме, для создания нового шаблона регистрации. Новая ЗБИ создается с помощью CapturedBIR. Также дополнительно функция может выполнить адаптацию нового шаблона на основе существующего ReferenceTemplate.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Enroll с назначением BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY для получения обработанного шаблона.
4) Вызвать функцию BioSPI_GetBIRHeaderFromHandle.
5) Проверить, что степень обработки ЗБИ соответствует обработанному биометрическому образцу.
6) Вызвать функцию BioSPI_CreateTemplate с обработанной ЗБИ в качестве входного параметра.
7) Проверить, отличается ли возвращаемое значение от значения BioAPI_OK.
8) Отсоединить и выгрузить испытуемый ПБУ.

```
     Если какие-либо промежуточные операции прошли
     неуспешно, то выдается заключение о соответствии
     "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
<input name="_capturetimeout"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_CreateTemplate_InputBIRDataType">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
</invoke>
```

```
        <!-- Процесс связан с функцией интерфейса обратного
        вызова инфраструктуры, предоставленной испытуемым
        компонентом ПБУ. При каждом входящем вызове функции
        связанный с ней процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
              package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
              function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_CreateTemplate_InputBIRDataType">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
            <input name="bspVersion" value="32"/>
            <input name="unitIDOrNull" value="0"/>
            <input name="bspHandle" var="_bsphandle"/>
            <input name="eventtimeouttime"
            var="inserttimeouttime"/>
        </invoke>
```

```
<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
```

```
        <input name="Timeout" var="capturetimeouttime"/>
        <output name="NewTemplate"
        setvar="capturedbir_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_Enroll возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Вызов функции BioSPI_GetHeaderFromHandle. -->
    <invoke function="BioSPI_GetHeaderFromHandle">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Handle" var="capturedbir_handle"/>
        <output name="ProcessedLevel"
        setvar="processedLevel"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
```

```
          break_if_false="true">
     <description>
          Функция BioSPI_GetHeaderFromHandle возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Степень обработки зарегистрированной ЗБИ соот-
          ветствует обработанному биометрическому образцу.
     </description>
     <equal_to var1="processedLevel"
     var2="__BioAPI_BIR_DATA_TYPE_PROCESSED"/>
</assert_condition>

<!-- Вызов функции BioSPI_CreateTemplate. -->
<invoke function="BioSPI_CreateTemplate">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="CapturedBIR_Form"
     var="__BioAPI_BIR_HANDLE_INPUT" />
     <input name="CapturedBIR_BIRHandle"
     var="capturedbir_handle"/>
     <output name="NewTemplate"
     setvar="newTemplate_handle"/>
     <return setvar="return"/>
</invoke>
```

```
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Функция BioSPI_CreateTemplate завершилась с
                ошибкой.
            </description>
            <not_equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_ModuleDetach и
        BioSPI_ModuleUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.37 Утверждение 10f *BioSPI_CreateTemplate_Inconsistent_Purpose*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_CreateTemplate* с недостоверным назначением входной ЗБИ.

Выдержки

Подпункт 9.3.4.2

BioAPI_RETURN BioAPI BioSPI_CreateTemplate

(BioAPI_HANDLE BSPHandle,

*const BioAPI_INPUT_BIR *CapturedBIR,*

*const BioAPI_INPUT_BIR *ReferenceTemplate,*

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*BioAPI_BIR_HANDLE *NewTemplate,*

*const BioAPI_DATA *Payload,*

*BioAPI_UUID *TemplateUUID).*

Подпункт 8.4.2.1

Данная функция принимает ЗБИ, содержащую биометрические данные в промежуточной форме, для создания нового шаблона регистрации. Новая ЗБИ создается с помощью CapturedBIR. Также дополнительно функция может выполнить адаптацию нового шаблона на основе существующего ReferenceTemplate.

Ссылки: 9.3.4.2 и 8.4.2.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture* с назначением BioAPI_PURPOSE_ VERIFY для получения ЗБИ.

4) Вызвать функцию *BioSPI_CreateTemplate* с обработанной ЗБИ в качестве входного параметра.

5) Проверить, соответствует ли возвращаемое значение значению Bio APIERR_INCONSISTENT_PURPOSE.

6) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_CreateTemplate* возвращает значение BioAPIERR_INCONSISTENT_PURPOSE.

Пакет языка утверждения

```
<package name="28ec1620-e995-11d9-b1d1-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
```

```
        Данный пакет содержит утверждение
        "BioSPI_CreateTemplate_Inconsistent_Purpose" (см. далее
        элемент <description> утверждения).
    </description>

    <assertion name="BioSPI_CreateTemplate_Inconsistent_Purpose"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью вызова
            функции BioSPI_CreateTemplate с недостоверным
            назначением входной ЗБИ.
            Следующий текст соответствует приведенному в 9.3.4.2
            и 8.4.2.1 спецификации БиоАПИ 2.0.
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_CreateTemplate
                (BioAPI_HANDLE BSPHandle,
                const BioAPI_INPUT_BIR *CapturedBIR,
                const BioAPI_INPUT_BIR *ReferenceTemplate,
                const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
                *OutputFormat,
                BioAPI_BIR_HANDLE *NewTemplate,
                const BioAPI_DATA *Payload,
                BioAPI_UUID *TemplateUUID).
            ------------------------------------------------
            Подпункт 8.4.2.1:
            Данная функция принимает ЗБИ, содержащую биометрические
            данные в промежуточной форме, для создания нового
            шаблона регистрации. Новая ЗБИ создается с помощью
            CapturedBIR. Также дополнительно функция может вы-
            полнить адаптацию нового шаблона на основе сущест-
            вующего ReferenceTemplate.
            ------------------------------------------------
            Порядок действий при испытании на соответствие:
```

```
      1) Загрузить испытуемый ПБУ.
      2) Присоединить испытуемый ПБУ.
      3) Вызвать функцию BioSPI_Capture с назначением
      BioAPI_PURPOSE_VERIFY для получения ЗБИ.
      4) Вызвать функцию BioSPI_CreateTemplate с
      обработанной ЗБИ в качестве входного параметра.
      5) Проверить, соответствует ли возвращаемое значение
      значению BioAPIERR_INCONSISTENT_PURPOSE.
      6) Отсоединить и выгрузить испытуемый ПБУ.

      Если какие-либо промежуточные операции прошли
      неуспешно, то выдается заключение о соответствии
      "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
<input name="_capturetimeout"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_CreateTemplate">
```

```
            <input name="bspUuid" var="_bspUuid"/>
            <input name="inserttimeouttime"
            var="_inserttimeout"/>
            <input name="nosourcepresentsupported"
            var="_noSourcePresentSupported"/>
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_CreateTemplate">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported"/>
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
```

```
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о
событии BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать
получения уведомления в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent"/>
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
```

```
        <not var="eventtimeoutflag"/>
    </assert_condition>

    <!-- ПБУ готов к получению ЗБИ. Вызвать функцию BioSPI_
    Capture для создания шаблона. Дескриптор созданной ЗБИ
    хранится в перменной "capturedbir". -->
    <invoke function="BioSPI_Capture">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Purpose"
        var="__BioAPI_PURPOSE_VERIFY"/>
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <input name="no_AuditData" value="true"/>
        <output name="CapturedBIR"
        setvar="capturedbir_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
              break_if_false="true">
        <description>
            Функция BioSPI_Capture возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Если степень обработки полученной ЗБИ соответствует об-
    работанному биометрическому образцу, то выдается заключение
```

```
о соответствии "UNDECIDED" и выполнение процесса
прерывается. -->
<invoke activity="check_capturedBIR_datatype"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true" >
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="BIRHandle" var="capturedbir_handle"/>
</invoke>

<!-- Вызов функции BioSPI_CreateTemplate. -->
<invoke function="BioSPI_CreateTemplate">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="CapturedBIR_Form"
     var="__BioAPI_BIR_HANDLE_INPUT"/>
     <input name="CapturedBIR_BIRHandle"
     var="capturedbir_handle"/>
     <output name="NewTemplate"
     setvar="newTemplate_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
     <description>
          Функция BioSPI_CreateTemplate возвращает
          значение BioAPIERR_INCONSISTENT_PURPOSE.
     </description>
     <equal_to var1="return"
          var2="__BioAPIERR_BSP_INCONSISTENT_PURPOSE"/>
</assert_condition>

<!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
<invoke activity="DetachAndUnload"
```

```
                    package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
                <input name="bspUuid" var="bspUuid"/>
                <input name="BSPHandle" var="_bsphandle"/>
            </invoke>
        </activity>
</package>
```

8.38 Утверждение 11a *BioSPI_Process_ValidParam*

Описание: Проверку данного утверждения проводят с целью испытания функции *BioSPI_Process* с достоверными параметрами и возвращенной степенью обработки.

Выдержки

Подпункт 9.3.4.3

BioAPI_RETURN BioAPI BioSPI_Process

(BioAPI_HANDLE BSPHandle,

*const BioAPI_INPUT_BIR *CapturedBIR,*

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*BioAPI_BIR_HANDLE *ProcessedBIR).*

Подпункт 8.4.3.1

Данная функция обрабатывает промежуточные данные, полученные путем вызова функции *BioAPI_Capture*, для последующей верификации или идентификации. Если присоединенный вызов ПБУ позволяет проводить обработку, то ПБУ создает обработанный биометрический образец ЗБИ. В противном случае параметру ProcessedBIR присваивается значение NULL и функция возвращает значение BioAPIERR_BSP_FUNCTION_NOT_SUPPORTED.

Ссылки: 9.3.4.3 и 8.4.3.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture* для получения промежуточной ЗБИ.

4) Вызвать функцию *BioSPI_Process* для создания обработанной ЗБИ. Предполагается возвращение значения BioAPI_OK.

5) Вызвать функцию *BioSPI_GetHeaderFromHandle* для обработанной ЗБИ. Предполагается, что обработанная ЗБИ будет иметь степень обработки, соответствующую обработанному биометрическому образцу.

6) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: ЗБИ, созданная функцией *BioSPI_Process*, имеет степень обработки, соответствующую обработанному биометрическому образцу.

Пакет языка утверждения

```
<package name="4ec34700-e9a0-11d9-8fc8-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение "BioSPI_Process_
        ValidParam " (см. далее элемент <description> утверждения).
    </description>

    <assertion name="BioSPI_Process_ValidParam"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью
            испытания функции BioSPI_Process с достоверными
            параметрами и возвращенной степенью обработки.
            Следующий текст соответствует приведенному в 9.3.4.3
            и 8.4.3.1 спецификации БиоАПИ 2.0.
            BioAPI_RETURN BioAPI BioSPI_Process
                (BioAPI_HANDLE BSPHandle,
                const BioAPI_INPUT_BIR *CapturedBIR,
```

```
            const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
            *OutputFormat,
            BioAPI_BIR_HANDLE *ProcessedBIR).
        ------------------------------------------------
        Подпункт 8.4.3.1:
        Данная функция обрабатывает промежуточные данные,
        полученные путем вызова функции BioAPI_Capture, для
        последующей верификации или идентификации. Если
        присоединенный вызов ПБУ позволяет проводить обработку,
        то ПБУ создает обработанный биометрический образец
        ЗБИ. В противном случае параметру ProcessedBIR
        присваивается значение NULL и функция возвращает
        значение BioAPIERR_BSP_FUNCTION_NOT_SUPPORTED.
        ------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Вызвать функцию BioSPI_Capture для получения
        промежуточной ЗБИ.
        4) Вызвать BioSPI_Process для создания обработанной
        ЗБИ. Предполагается возвращение значения BioAPI_OK.
        5) Вызвать функцию BioSPI_GetHeaderFromHandle для
        обработанной ЗБИ. Предполагается, что обработанная
        ЗБИ будет иметь степени обработки, соответствующую
        обработанному биометрическому образцу.
        6) Отсоединить и выгрузить испытуемый ПБУ.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>
```

```
<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
<input name="_capturetimeout"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_Process">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного
вызова инфраструктуры, предоставленной испытуемым
компонентом ПБУ. При каждом входящем вызове функции связанный
с ней процесс будет вызываться автоматически. -->
<bind activity="EventHandler"
     package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
```

```
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_Process">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
            <input name="bspVersion" value="32"/>
            <input name="unitIDOrNull" value="0"/>
            <input name="bspHandle" var="_bsphandle"/>
            <input name="eventtimeouttime"
            var="inserttimeouttime"/>
        </invoke>

        <set name="eventtimeoutflag" value="false"/>
        <!-- Если испытуемый ПБУ поддерживает уведомление о событии
        BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
        уведомления о событии в пределах указанной максимальной
        продолжительности. -->
        <wait_until timeout_var="sourcepresenttimeouttime"
```

```
            setvar="eventtimeoutflag">
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Испытуемый ПБУ не поддерживает уведомление о
            событии BioAPI_NOTIFY_SOURCE_PRESENT или
            уведомление о событии получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию BioSPI_
Capture для создания шаблона. Дескриптор созданной ЗБИ
хранится в переменной "capturedbir". -->
<invoke function="BioSPI_Capture">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Purpose"
        var="__BioAPI_PURPOSE_VERIFY"/>
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <input name="no_AuditData" value="true"/>
        <output name="CapturedBIR"
        setvar="capturedbir_handle"/>
        <return setvar="return"/>
</invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Capture возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Если степень обработки полученной ЗБИ соответствует
обработанному биометрическому образцу, то выдается
заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается.-->
<invoke activity="check_capturedBIR_datatype"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true" >
     <input name="BSPHandle" var="_bsphandle" />
     <input name="BIRHandle" var="capturedbir_handle" />
</invoke>

<!-- Вызов функции BioSPI_Process. -->
<invoke function="BioSPI_Process">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="CapturedBIR_Form"
     var="__BioAPI_BIR_HANDLE_INPUT" />
     <input name="CapturedBIR_BIRHandle"
     var="capturedbir_handle"/>
     <output name="ProcessedBIR"
     setvar="processedbir_handle"/>
     <return setvar="return"/>
```

```
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
     <description>
          Функция BioSPI_Process возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetHeaderFromHandle. -->
<invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="processedbir_handle"/>
     <output name="ProcessedLevel"
     setvar="processedLevel"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_GetHeaderFromHandle возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
```

```
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Степень обработки ЗБИ соответствует обработанному
                биометрическому образцу.
            </description>
            <equal_to var1="processedLevel"
            var2="__BioAPI_BIR_DATA_TYPE_PROCESSED"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.39 Утверждение 11b *BioSPI_Process_BIRHeaderQuality*

Описание: Проверку данного утверждения проводят с целью испытания функции *BioSPI_Process* с достоверными параметрами и проверки, является ли достоверным возвращенное значение качества.

Выдержки

Подпункт 9.3.4.3

BioAPI_RETURN BioAPI BioSPI_Process

(BioAPI_HANDLE BSPHandle,

*const BioAPI_INPUT_BIR *CapturedBIR,*

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*BioAPI_BIR_HANDLE *ProcessedBIR).*

Подпункт 8.4.3.1

Данная функция обрабатывает промежуточные данные, полученные путем вызова функции *BioAPI_Capture*, для последующей верификации или идентификации. Если присоединенный вызов ПБУ позволяет проводить обработку, то ПБУ создает обработанный биометрический образец ЗБИ. В противном случае параметру ProcessedBIR присваивается значение NULL и функция возвращает значение BioAPIERR_BSP_FUNCTION_NOT_SUPPORTED.

Подпункт 7.49.3

Значение качества представляет собой целое число в диапазоне от 0 до 100, кроме следующих значений:

Значение "–1": значение BioAPI_QUALITY не было установлено ПБУ (соответствующая информация приведена в документации разработчика ПБУ).

Значение "–2": параметр BioAPI_QUALITY не поддерживается ПБУ.

Подраздел 7.47

#define BioAPI_QUALITY_PROCESSED (0x00000008)

Если данное значение установлено, то ПБУ поддерживает возврат значения качества (в заголовке ЗБИ) для обработанных биометрических данных.

Подпункт А.4.6.2.2

После обработки ЗБИ может быть выполнено повторное вычисление значения качества, которое может быть включено в заголовок ProcessedBIR, и, кроме того, в заголовок AdaptedBIR. Это происходит в процессе выполнения функций *BioAPI_CreateTemplate*, *BioAPI_Process*, *BioAPI_Verify*, *BioAPI_VerifyMatch*, *Bio API_Enroll*, и *BioAPI_Import* (ConstructedBIR – ЗБИ, содержащая импортированные биометрические данные).

ПБУ должен ввести в реестр компонентов информацию о том, поддерживает ли он вычисление качества для ЗБИ каждого типа: исходной, промежуточной и обработанной.

Ссылки: 9.3.4.3, 8.4.3.1, 7.49.3, 7.47 и А.4.6.2.2.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture* для получения промежуточной ЗБИ.

4) Вызвать функцию *BioSPI_Process* для создания обработанной ЗБИ. Предполагается, что функция возвратит значение BioAPI_OK.

5) Вызвать функцию *BioSPI_GetHeaderFromHandle* для обработанной ЗБИ. Проверить достоверность значения качества.

6) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: ЗБИ, созданная функцией *BioSPI_Process*, имеет достоверное значение качества.

Пакет языка утверждения

```
<package name="211668e0-e9a6-11d9-bcc8-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_Process_
          BIRHeaderQuality" (см. далее элемент
          <description> утверждения).
     </description>

     <assertion name="BioSPI_Process_BIRHeaderQuality"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью испытания
               функции BioSPI_Process с достоверными параметрами и
               проверки, является ли достоверным возвращенное
               значение качества.
```

Следующий текст соответствует приведенному в 9.3.4.3, 8.4.3.1, 7.49.3, 7.47 и A.4.6.2.2 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_Process
     (BioAPI_HANDLE BSPHandle,
     const BioAPI_INPUT_BIR *CapturedBIR,
     const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
     *OutputFormat,
     BioAPI_BIR_HANDLE *ProcessedBIR).
```

Подпункт 8.4.3.1:

Данная функция обрабатывает промежуточные данные, полученные путем вызова функции BioAPI_Capture, для последующей верификации или идентификации. Если присоединенный вызов ПБУ позволяет проводить обработку, то ПБУ создает обработанный биометрический образец ЗБИ. В противном случае параметру ProcessedBIR присваивается значение NULL и функция возвращает значение BioAPIERR_BSP_FUNCTION_NOT_SUPPORTED.

Подпункт 7.49.3

Значение качества представляет собой целое число в диапазоне от 0 до 100, кроме следующих значений:

Значение "-1": значение BioAPI_QUALITY не было установлено ПБУ (соответствующая информация приведена в документации разработчика ПБУ).

Значение "-2": параметр BioAPI_QUALITY не поддерживается ПБУ.

Подраздел 7.47:

```
#define BioAPI_QUALITY_PROCESSED (0x00000008)
```

Если данное значение установлено, то ПБУ поддерживает возврат значения качества (в заголовке ЗБИ) для обработанных биометрических данных.

Подпункт A.4.6.2.2

После обработки ЗБИ может быть выполнено повторное вычисление значения качества, которое может быть включено в заголовок ProcessedBIR (вновь созданная обработанная ЗБИ), и, кроме того, в заголовок AdaptedBIR (адаптированная ЗБИ). Это происходит в процессе выполнения функций BioAPI_CreateTemplate, BioAPI_Process, BioAPI_Verify, BioAPI_VerifyMatch, BioAPI_Enroll и BioAPI_Import (ConstructedBIR - ЗБИ, содержащая импортированные биометрические данные). ПБУ должен внести в реестр компонентов информацию о том, поддерживает ли он вычисление качества для ЗБИ каждого типа: исходной, промежуточной и обработанной.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Capture для получения промежуточной ЗБИ.
4) Вызвать функцию BioSPI_Process для создания обработанной ЗБИ. Предполагается, что функция возвратит значение BioAPI_OK.
5) Вызвать функцию BioSPI_GetHeaderFromHandle для обработанной ЗБИ. Проверить достоверность значения качества.
6) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

</description>

<!-- УУИД испытуемого ПБУ. -->

```
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
<input name="_capturetimeout"/>

<!-- Указание на поддержку испытуемым ПБУ значения
качества в обработанной ЗБИ. -->
<input name="_processedQualitySupported" />

<!--Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_Process">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
     <input name="processedQualitySupported"
     var="_processedQualitySupported" />
</invoke>
```

```
        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_Process">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>
        <input name="processedQualitySupported" />

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                  package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                  break_on_break="true">
             <input name="bspUuid" var="bspUuid"/>
             <input name="bspVersion" value="32"/>
             <input name="unitIDOrNull" value="0"/>
             <input name="bspHandle" var="_bsphandle"/>
             <input name="eventtimeouttime"
             var="inserttimeouttime"/>
```

```
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомления о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Capture для создания шаблона. Дескриптор созданной
ЗБИ хранится в переменной "capturedbir". -->
<invoke function="BioSPI_Capture">
     <input name="BSPHandle" var="_bsphandle"/>
```

```
        <input name="Purpose"
        var="__BioAPI_PURPOSE_VERIFY"/>
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <input name="no_AuditData" value="true"/>
        <output name="CapturedBIR"
        setvar="capturedbir_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
              break_if_false="true">
        <description>
              Функция BioSPI_Capture возвращает значение
              BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Если степень обработки полученной ЗБИ соответствует
    обработанному биометрическому образцу, то выдается
    заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается. -->
    <invoke activity="check_capturedBIR_datatype"
              package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
              break_on_break="true" >
        <input name="BSPHandle" var="_bsphandle" />
        <input name="BIRHandle" var="capturedbir_handle" />
    </invoke>
```

```
<!-- Вызов функции BioSPI_Process. -->
<invoke function="BioSPI_Process">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="CapturedBIR_Form"
     var="__BioAPI_BIR_HANDLE_INPUT" />
     <input name="CapturedBIR_BIRHandle"
     var="capturedbir_handle"/>
     <output name="ProcessedBIR"
     setvar="processedbir_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Process возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetHeaderFromHandle с
предоставлением обработанной ЗБИ. -->
<invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="processedbir_handle"/>
     <output name="Quality" setvar="quality"/>
     <return setvar="return"/>
</invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_GetHeaderFromHandle возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<invoke activity="check_quality_supported"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <only_if>
          <same_as var1="processedQualitySupported"
          value2="true" />
     </only_if>
     <input name="quality" var="quality" />
</invoke>

<invoke activity="check_quality_not_supported"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true" >
     <only_if>
          <same_as var1="processedQualitySupported"
          value2="false" />
     </only_if>
     <input name="quality" var="quality" />
</invoke>
```

```
        <!-- Вызов функции BioSPI_BSPDetach и BioSPI_BSPUnload -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.40 Утверждение 11c *BioSPI_Process_OutputBIRPurpose*

Описание: Проверку данного утверждения проводят с целью испытания функции *BioSPI_Process* с достоверными параметрами и проверки совпадения назначения обработанной ЗБИ с назначением зарегистрированной ЗБИ.

Выдержки

Подпункт 9.3.4.3

BioAPI_RETURN BioAPI BioSPI_Process

(BioAPI_HANDLE BSPHandle,

*const BioAPI_INPUT_BIR *CapturedBIR,*

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*BioAPI_BIR_HANDLE *ProcessedBIR).*

Подпункт 8.4.3.1

Данная функция обрабатывает промежуточные данные, полученные путем вызова функции *BioAPI_Capture*, для последующей верификации или идентификации. Если присоединенный вызов ПБУ позволяет проводить обработку, то ПБУ создает обработанный биометрический образец ЗБИ. В противном случае параметру ProcessedBIR присваивается значение NULL и функция возвращает значение BioAPIERR_BSP_FUNCTION_NOT_SUPPORTED.

Пункт 7.12.3

e) функции *BioAPI_Process*, *BioAPI_CreateTemplate* и *BioAPI_ProcessWithAuxData* не используют назначение в качестве входного параметра, а считывают поле назначения из заголовка входной ЗБИ;

f) функция *BioAPI_Process* может принимать в качестве входных данных любую промежуточную ЗБИ с назначениями BioAPI_PURPOSE_VERIFY и BioAPI_ PURPOSE_IDENTIFY и должна возвращать ЗБИ с тем же назначением, что и входная ЗБИ.

Ссылки: 9.3.4.3, 8.4.3.1 и 7.12.3.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture* с назначением BioAPI_PURPOSE_ ENROLL_FOR_VERIFICATION_ONLY для получения ЗБИ для регистрации.

4) Вызвать функцию *BioSPI_Process*.

5) Проверить возвращаемый код. Предполагается, что функция возвратит значение BioAPI_OK.

6) Вызвать функцию *BioSPI_GetHeaderFromHandle* для сравнения CapturedBIR с Processed. Предполагается, что они имеют одно и то же назначение.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: ЗБИ, созданная функцией *BioSPI_Process*, имеет назначение BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY.

Пакет языка утверждения

```
<package name="e1bb4f20-ed61-11d9-9344-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
```

```
     Данный пакет содержит утверждение "BioSPI_Process_
     OutputBIRPurpose" (см. далее элемент <description>
     утверждения).
</description>

<assertion name="BioSPI_Process_OutputBIRPurpose"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью
          испытания функции BioSPI_Process с достоверными
          параметрами и проверки совпадения назначения
          обработанной ЗБИ с назначением зарегистрированной ЗБИ.
          Следующий текст соответствует приведенному в 8.4.3.1
          и 7.12.3 спецификации БиоАПИ 2.0.
          ------------------------------------------------
          BioAPI_RETURN BioAPI BioSPI_Process
               (BioAPI_HANDLE BSPHandle,
               const BioAPI_INPUT_BIR *CapturedBIR,
               const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
               *OutputFormat,
               BioAPI_BIR_HANDLE *ProcessedBIR).
          ------------------------------------------------
          Подпункт 8.4.3.1
          Данная функция обрабатывает промежуточные данные,
          полученные путем вызова функции BioAPI_Capture, для
          последующей верификации или идентификации. Если
          присоединенный вызов ПБУ позволяет проводить обработку,
          то ПБУ создает обработанный биометрический образец
          ЗБИ. В противном случае параметру ProcessedBIR
          присваивается значение NULL и функция возвращает
          значение BioAPIERR_BSP_FUNCTION_NOT_SUPPORTED.
          ------------------------------------------------
          Пункт 7.12.3:
          e) функции BioAPI_Process, BioAPI_CreateTemplate и
          BioAPI_ProcessWithAuxData не используют назначение в
```

```
качестве входного параметра, а считывают поле
назначения из заголовка входной ЗБИ;
f) функция BioAPI_Process может принимать
в качестве входных данных любую
промежуточную ЗБИ с назначениями BioAPI_PURPOSE_
VERIFY и BioAPI_PURPOSE_IDENTIFY и должна возвращать
ЗБИ с тем же назначением, что и входная ЗБИ.
-------------------------------------------------
Порядок действий при испытании на соответствие:
1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Capture с назначением
BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY для
получения ЗБИ для регистрации.
4) Вызвать функцию BioSPI_Process.
5) Проверить возвращаемый код. Предполагается, что
функция возвратит значение BioAPI_OK.
6) Вызвать функцию BioSPI_GetHeaderFromHandle для
CapturedBIR и Processed BIR с целью их сравнения
друг с другом. Предполагается, что они имеют одно и
то же назначение.

Если какие-либо промежуточные операции прошли
неуспешно, то выдается заключение о соответствии
"UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
```

```
        <input name="_noSourcePresentSupported" />

        <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
        PRESENT. -->
        <input name="_sourcepresenttimeout"/>

        <!-- Время ожидания для функции BioSPI_Capture. -->
        <input name="_capturetimeout"/>

        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_Process">
            <input name="bspUuid" var="_bspUuid"/>
            <input name="inserttimeouttime"
            var="_inserttimeout"/>
            <input name="nosourcepresentsupported"
            var="_noSourcePresentSupported"/>
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_Process">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
```

```
<input name="nosourcepresentsupported"/>
<input name="sourcepresenttimeouttime"/>
<input name="capturetimeouttime"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent"/>
</wait_until>
```

```
<!-- Выдача заключения о соответствии. Если условие, определенное далее в элементе <description>, ложное, выдается заключение о соответствии "UNDECIDED" и выполнение процесса прерывается, в противном случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о событии BioAPI_NOTIFY_SOURCE_PRESENT или уведомление о событии получено в пределах указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию BioSPI_Capture для верификации. Дескриптор созданной ЗБИ хранится в переменной "capturedbir". -->
<invoke function="BioSPI_Capture">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_VERIFY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <input name="no_AuditData" value="true"/>
     <output name="CapturedBIR"
     setvar="capturedbir_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, определенное далее в элементе <description>, ложное, выдается заключение о соответствии "UNDECIDED" и выполнение процесса прерывается, в противном случае выдается заключение о соответствии "PASS". -->
```

```
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Capture возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Если степень обработки полученной ЗБИ соответствует
обработанному биометрическому образцу, то выдается
заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается. -->
<invoke activity="check_capturedBIR_datatype"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true" >
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="BIRHandle" var="capturedbir_handle"/>
</invoke>

<!-- Вызов функции BioSPI_Process. -->
<invoke function="BioSPI_Process">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="CapturedBIR_Form"
     var="__BioAPI_BIR_HANDLE_INPUT"/>
     <input name="CapturedBIR_BIRHandle"
     var="capturedbir_handle"/>
     <output name="ProcessedBIR" set-
     var="processedbir_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
```

```
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Process возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetHeaderFromHandle с
предоставлением полученной ЗБИ. -->
<invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="capturedbir_handle"/>
     <output name="Purpose" setvar="purpose1"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_GetHeaderFromHandle возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetHeaderFromHandle для вновь
созданной обработанной ЗБИ. -->
```

```
<invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="processedbir_handle"/>
     <output name="Purpose" setvar="purpose2"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_GetHeaderFromHandle возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
     <description>
          Назначение обработанной ЗБИ совпадает с
          назначением полученной ЗБИ.
     </description>
     <equal_to var1="purpose1" var2="purpose2"/>
</assert_condition>

<!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
```

```
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid"/>
            <input name="BSPHandle" var="_bsphandle"/>
        </invoke>
    </activity>
</package>
```

8.41 Утверждение 11d *BioSPI_Process_BuildsProcessedBIR*

Описание: Проверку данного утверждения проводят с целью испытания функции *BioSPI_Process* с достоверными параметрами и проверки возвращаемой степени обработки.

Выдержки

Подпункт 9.3.4.3

BioAPI_RETURN BioAPI BioSPI_Process

(BioAPI_HANDLE BSPHandle,

*const BioAPI_INPUT_BIR *CapturedBIR,*

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*BioAPI_BIR_HANDLE *ProcessedBIR).*

Подпункт 8.4.3.1

Данная функция обрабатывает промежуточные данные, полученные путем вызова функции *BioAPI_Capture*, для последующей верификации или идентификации. Если присоединенный вызов ПБУ позволяет проводить обработку, то ПБУ создает обработанный биометрический образец ЗБИ. В противном случае параметру ProcessedBIR присваивается значение NULL и функция возвращает значение BioAPIERR_BSP_FUNCTION_NOT_SUPPORTED.

Ссылки: 9.3.4.3 и 8.4.3.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture* для получения промежуточной ЗБИ.

4) Вызвать функцию *BioSPI_Process* для создания обработанной ЗБИ. Предполагается, что функция возвратит значение BioAPI_OK.

5) Вызвать функцию *BioSPI_GetHeaderFromHandle* для обработанной ЗБИ. Проверить достоверность типа данных.

6) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: ЗБИ, созданная функцией *BioSPI_Process* из промежуточной ЗБИ, является обработанной.

Пакет языка утверждения

```
<package name="f2ce6540-ed66-11d9-9618-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение
        "BioSPI_Process_BuildsProcessedBIR" (см. далее элемент
        <description> утверждения).
    </description>

    <assertion name="BioSPI_Process_BuildsProcessedBIR"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью испытания
            функции BioSPI_Process с достоверными параметрами и
            проверки возвращаемой степени обработки.
            Следующий текст соответствует приведенному в 8.4.3.1
            и C.3.4.2 спецификации БиоАПИ 2.0.
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_Process
                (BioAPI_HANDLE BSPHandle,
                const BioAPI_INPUT_BIR *CapturedBIR,
```

```
    const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
    *OutputFormat,
    BioAPI_BIR_HANDLE *ProcessedBIR).
-------------------------------------------------
Подпункт 8.4.3.1
Данная функция обрабатывает промежуточные данные,
полученные путем вызова функции BioAPI_Capture, для
последующей верификации или идентификации. Если
присоединенный вызов ПБУ позволяет проводить обработку,
то ПБУ создает обработанный биометрический образец
ЗБИ. В противном случае параметру ProcessedBIR
присваивается значение NULL и функция возвращает
значение BioAPIERR_BSP_FUNCTION_NOT_SUPPORTED.
-------------------------------------------------
Подпункт C.3.4.2:
Функция всегда принимает в качестве входного параметра
промежуточную ЗБИ и может выполнять обработку до
получения конечного шаблона определенного типа,
соответствующего указанному назначению.
-------------------------------------------------
Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Capture для получения про-
межуточной ЗБИ.
4) Вызвать функцию BioSPI_Process для создания обра-
ботанной ЗБИ. Предполагается, что функция возвратит
значение BioAPI_OK.
5) Вызвать функцию BioSPI_GetHeaderFromHandle для об-
работанной ЗБИ. Проверить достовереность типа данных.
6) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли
неуспешно, то выдается заключение о соответствии
"UNDECIDED".
```

```
</description>

<!--УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
<input name="_capturetimeout"/>

<!--Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_Process">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported"/>
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
</invoke>
```

```
    <!-- Процесс связан с функцией интерфейса обратного вызова
    инфраструктуры, предоставленной испытуемым компонентом
    ПБУ. При каждом входящем вызове функции связанный с ней
    процесс будет вызываться автоматически. -->
    <bind activity="EventHandler"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_Process">
    <input name="bspUuid"/>
    <input name="inserttimeouttime"/>
    <input name="nosourcepresentsupported" />
    <input name="sourcepresenttimeouttime"/>
    <input name="capturetimeouttime"/>

    <!-- Данное утверждение использует BSPHandle со значением
    "1" для всех вызовов BioSPI, которые этого требуют. -->
    <set name="_bsphandle" value="1"/>

    <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
    предоставленных испытуемым ПБУ. Входное значение параметра
    "unitIDOrNull" равно "0", поэтому утверждение будет
    тестировать модуль датчика, выбранный ПБУ. -->
    <invoke activity="LoadAndAttach"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            break_on_break="true">
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
    </invoke>

    <set name="eventtimeoutflag" value="false"/>
```

```
<!-- Если испытуемый ПБУ поддерживает уведомление о
событии BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать
получения уведомления о событии в пределах указанной
максимальной продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Capture для создания шаблона. Дескриптор созданной
ЗБИ хранится в перменной "capturedbir". -->
<invoke function="BioSPI_Capture">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_VERIFY"/>
     <input name="Subtype" value="0"/>
```

```
     <input name="Timeout" var="capturetimeouttime"/>
     <input name="no_AuditData" value="true"/>
     <output name="CapturedBIR"
     setvar="capturedbir_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и
выполнение процесса прерывается. -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Capture возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Если степень обработки полученной ЗБИ соответствует
обработанному биометрическому образцу, то выдается
заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается.-->
<invoke activity="check_capturedBIR_datatype"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true" >
     <input name="BSPHandle" var="_bsphandle" />
     <input name="BIRHandle" var="capturedbir_handle" />
</invoke>

<!-- Вызов функции BioSPI_Process. -->
<invoke function="BioSPI_Process">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="CapturedBIR_Form"
     var="__BioAPI_BIR_HANDLE_INPUT" />
```

```
        <input name="CapturedBIR_BIRHandle"
        var="capturedbir_handle"/>
        <output name="ProcessedBIR"
        setvar="processedbir_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_Process возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Вызов функции BioSPI_GetHeaderFromHandle с
    предоставлением обработанной ЗБИ. -->
    <invoke function="BioSPI_GetHeaderFromHandle">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Handle" var="processedbir_handle"/>
        <output name="ProcessedLevel"
        setvar="processedLevel"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
```

```
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
                break_if_false="true">
            <description>
                Функция BioSPI_GetHeaderFromHandle возвращает
                значение BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Выдача заключения о соответствии. Если условие, оп-
        ределенное далее в элементе <description>, ложное,
        выдается заключение о соответствии «FAIL», в противном
        случае выдается заключение о соответствии «PASS». -->
        <assert_condition>
            <description>
                Степень обработки выходной ЗБИ соответствует
                обработанному биометрическому образцу.
            </description>
            <equal_to var1="processedLevel"
                var2="__BioAPI_BIR_DATA_TYPE_PROCESSED"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.42 Утверждение 11e *BioSPI_Process_InputBIRDataType*

Описание: Проверку данного утверждения проводят с целью испытания функции *BioSPI_Process* с обработанной ЗБИ в качестве входного параметра. В этом случае вызов функции *BioSPI_Process* должен завершиться с ошибкой.

Выдержки

Подпункт 9.3.4.3

BioAPI_RETURN BioAPI BioSPI_Process

(BioAPI_HANDLE BSPHandle,

*const BioAPI_INPUT_BIR *CapturedBIR,*

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*BioAPI_BIR_HANDLE *ProcessedBIR).*

Подпункт 8.4.3.1

Данная функция обрабатывает промежуточные данные, полученные путем вызова функции *BioAPI_Capture*, для последующей верификации или идентификации. Если присоединенный вызов ПБУ позволяет проводить обработку, то ПБУ создает обработанный биометрический образец ЗБИ. В противном случае параметру ProcessedBIR присваивается значение NULL и функция возвращает значение BioAPIERR_BSP_FUNCTION_NOT_SUPPORTED.

Ссылки: 9.3.4.3 и 8.4.3.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Capture* с назначением BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY для получения ЗБИ для регистрации.

4) Вызвать функцию *BioSPI_GetBIRFromHandle*.

5) Вызвать функцию *BioSPI_Process*, если функция *BioSPI_Capture* вернула промежуточную ЗБИ.

6) Вызвать функцию *BioSPI_Process* с обработанной ЗБИ в качестве входного параметра.

7) Проверить, отличается ли возвращенное значение от значения BioAPI_OK.

8) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_Process* завершается с ошибкой.

Пакет языка утверждения

```
<package name="3cf96080-ed6b-11d9-9acf-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение "BioSPI_Process_
        InputBIRDataType" (см. далее элемент <description>
        утверждения).
    </description>

    <assertion name="BioSPI_Process_InputBIRDataType"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью
            испытания функции BioSPI_Process с обработанной ЗБИ
            в качестве входного параметра. В этом случае вызов
            функции BioSPI_Process должен завершиться с ошибкой.
            Следующий текст соответствует приведенному в С.3.4.2
            и 2.5.3.3 спецификации БиоАПИ 2.0.
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_Process
                (BioAPI_HANDLE BSPHandle,
                const BioAPI_INPUT_BIR *CapturedBIR,
                const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
                *OutputFormat,
```

BioAPI_BIR_HANDLE *ProcessedBIR).

Подпункт 8.4.3.1

Данная функция обрабатывает промежуточные данные, полученные путем вызова функции BioAPI_Capture, для последующей верификации или идентификации. Если присоединенный вызов ПБУ позволяет проводить обработку, то ПБУ создает обработанный биометрический образец ЗБИ. В противном случае параметру ProcessedBIR присваивается значение NULL и функция возвращает значение BioAPIERR_BSP_FUNCTION_NOT_SUPPORTED.

Подпункт C.3.4.2:

Функция всегда принимает в качестве входного параметра промежуточную ЗБИ и может выполнять обработку до получения конечного шаблона определенного типа, соответствующего указанному назначению.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Capture с назначением BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY для получения ЗБИ для регистрации.
4) Вызвать функцию BioSPI_GetBIRFromHandle.
5) Вызвать функцию BioSPI_Process, если функция BioSPI_Capture вернула промежуточную ЗБИ.
6) Вызвать функцию BioSPI_Process с обработанной ЗБИ в качестве входного параметра.
7) Проверить, отличается ли возвращенное значение от значения BioAPI_OK.
8) Отсоединить и выгрузить испытуемый ПБУ.

```
     Если какие-либо промежуточные операции прошли
     неуспешно, то выдается заключение о соответствии
     "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
<input name="_capturetimeout"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_Process_InputBIRDataType">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
</invoke>
```

```
        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_Process_InputBIRDataType">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                  package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                  break_on_break="true">
             <input name="bspUuid" var="bspUuid"/>
             <input name="bspVersion" value="32"/>
             <input name="unitIDOrNull" value="0"/>
             <input name="bspHandle" var="_bsphandle"/>
             <input name="eventtimeouttime"
             var="inserttimeouttime"/>
        </invoke>
```

```
<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию BioSPI_
Capture для создания шаблона. Дескриптор созданной ЗБИ
хранится в переменной "capturedbir". -->
<invoke function="BioSPI_Capture">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_VERIFY"/>
```

```
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <input name="no_AuditData" value="true"/>
        <output name="CapturedBIR"
        setvar="capturedbir_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
              break_if_false="true">
        <description>
              Функция BioSPI_Capture возвращает значение
              BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Вызов функции BioSPI_GetHeaderFromHandle с полученной
    ЗБИ в качестве входного параметра. -->
    <invoke function="BioSPI_GetHeaderFromHandle">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Handle" var="capturedbir_handle"/>
        <output name="ProcessedLevel"
        setvar="processedLevel"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
```

```
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_GetHeaderFromHandle возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Использовать полученную ЗБИ, если она обработана. -->
<set name="_processedbir_handle"
var="capturedbir_handle">
     <only_if>
          <equal_to var1="processedLevel"
          var2="__BioAPI_BIR_DATA_TYPE_PROCESSED" />
     </only_if>
</set>

<!-- Полная обработка промежуточной ЗБИ. -->
<invoke activity="process_bir"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true" >
     <only_if>
          <not_equal_to var1="processedLevel"
          var2="__BioAPI_BIR_DATA_TYPE_PROCESSED" />
     </only_if>
     <input name="BSPHandle" var="_bsphandle" />
     <input name="CapturedBIR_BIRHandle"
     var="capturedbir_handle" />
</invoke>

<!-- Вызов функции BioSPI_Process собработанной ЗБИ в
качестве входного параметра. -->
<invoke function="BioSPI_Process">
```

```
                <input name="BSPHandle" var="_bsphandle"/>
                <input name="CapturedBIR_Form"
                var="__BioAPI_BIR_HANDLE_INPUT" />
                <input name="CapturedBIR_BIRHandle"
                var="_processedbir_handle"/>
                <output name="ProcessedBIR"
                setvar="processedbir_handle"/>
                <return setvar="return"/>
            </invoke>

            <!-- Выдача заключения о соответствии. Если условие, оп-
            ределенное далее в элементе <description>, ложное,
            выдается заключение о соответствии "FAIL", в противном
            случае выдается заключение о соответствии "PASS". -->
            <assert_condition>
                <description>
                    Функция BioSPI_Process возвращает код ошибки.
                </description>
                <not_equal_to var1="return" var2="__BioAPI_OK"/>
            </assert_condition>

            <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
            <invoke activity="DetachAndUnload"
                    package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
                <input name="bspUuid" var="bspUuid" />
                <input name="BSPHandle" var="_bsphandle" />
            </invoke>
        </activity>
</package>
```

8.43 Утверждение 12a *BioSPI_VerifyMatch_ValidParam*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_VerifyMatch* с достоверными значениями входных параметров.

Выдержки

Подпункт 9.3.4.5

BioAPI_RETURN BioAPI BioSPI_VerifyMatch
(BioAPI_HANDLE BSPHandle,
BioAPI_FMR MaxFMRRequested,
*const BioAPI_INPUT_BIR *ProcessedBIR,*
*const BioAPI_INPUT_BIR *ReferenceTemplate,*
*BioAPI_BIR_HANDLE *AdaptedBIR,*
*BioAPI_BOOL *Result,*
*BioAPI_FMR *FMRAchieved,*
*BioAPI_DATA *Payload).*

Подпункт 8.4.5.1

Данная функция выполняет верификацию, то есть сравнение двух ЗБИ: ProcessedBIR и ReferenceTemplate. ProcessedBIR представляет собой обработанную ЗБИ, специально созданную для проведения верификации. ReferenceTemplate (контрольный шаблон) создается при регистрации.

Ссылки: 9.3.4.5 и 8.4.5.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll* для создания шаблона.

4) Зарегистрировать вновь созданную ЗБИ.

5) Вызвать функцию *BioSPI_Process* с полученной ЗБИ в качестве входного параметра, если рассматриваемая ЗБИ является промежуточной.

6) Использовать обработанную ЗБИ для ее сравнения с сохраненным шаблоном.

7) Проверить возвращаемый код.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_VerifyMatch* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="688aad60-ee30-11d9-a62c-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_VerifyMatch_
          ValidParam" (см. далее элемент <description> утверждения).
     </description>

     <assertion name="BioSPI_VerifyMatch_ValidParam"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               определения возвращения значения BioAPI_OK при вызове
               функции BioSPI_VerifyMatch с достоверными значениями
               входных параметров.
               Следующий текст соответствует приведенному в 8.4.5.1
               и 9.3.4.5 спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioSPI_VerifyMatch
                    (BioAPI_HANDLE BSPHandle,
                    BioAPI_FMR MaxFMRRequested,
                    const BioAPI_INPUT_BIR *ProcessedBIR,
                    const BioAPI_INPUT_BIR *ReferenceTemplate,
                    BioAPI_BIR_HANDLE *AdaptedBIR,
                    BioAPI_BOOL *Result,
                    BioAPI_FMR *FMRAchieved,
                    BioAPI_DATA *Payload).
```

```
        Данная функция выполняет верификацию, то есть сравнение
        двух ЗБИ: ProcessedBIR и ReferenceTemplate. Processed
        BIR представляет собой обработанную ЗБИ, специально
        созданную для проведения верификации. ReferenceTemplate
        (контрольный шаблон) создается при регистрации.
        ---------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Вызвать функцию BioSPI_Enroll для создания шаблона.
        4) Зарегистрировать вновь созданную ЗБИ.
        5) Вызвать функцию BioSPI_Process с полученной ЗБИ в
        качестве входного параметра, если рассматриваемая
        ЗБИ является промежуточной.
        6) Использовать обработанную ЗБИ для сравнения с со-
        храненным шаблоном.
        7) Проверить возвращаемый код.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />
```

```
<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функций BioSPI_Capture и
BioSPI_Enroll. -->
<input name="_capturetimeout"/>

<!-- Порог сопоставления для функции BioSPI_VerifyMatch. -->
<input name="_maxFMRRequested" />

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_VerifyMatch_ValidParam">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
     <input name="maxFMRRequested"
     var="_maxFMRRequested"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически. -->
<bind activity="EventHandler"
     package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
     function="BioSPI_EventHandler"/>
```

```
</assertion>

<activity name="BioSPI_VerifyMatch_ValidParam">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported" />
     <input name="sourcepresenttimeouttime"/>
     <input name="capturetimeouttime"/>
     <input name="maxFMRRequested" />

     <!-- Данное утверждение использует BSPHandle со значением
     "1" для всех вызовов BioSPI, которые этого требуют. -->
     <set name="_bsphandle" value="1"/>

     <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
     предоставленных испытуемым ПБУ. Входное значение параметра
     "unitIDOrNull" равно "0", поэтому утверждение будет
     тестировать модуль датчика, выбранный ПБУ. -->
     <invoke activity="LoadAndAttach"
               package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
               break_on_break="true">
          <input name="bspUuid" var="bspUuid"/>
          <input name="bspVersion" value="32"/>
          <input name="unitIDOrNull" value="0"/>
          <input name="bspHandle" var="_bsphandle"/>
          <input name="eventtimeouttime"
          var="inserttimeouttime"/>
     </invoke>

     <set name="eventtimeoutflag" value="false"/>

     <!-- Если испытуемый ПБУ поддерживает уведомление о событии
     BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
     уведомления о событии в пределах указанной максимальной
     продолжительности. -->
     <wait_until timeout_var="sourcepresenttimeouttime"
```

```
            setvar="eventtimeoutflag">
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent" />
    </wait_until>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED", и выполне-
    ние процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Испытуемый ПБУ не поддерживает уведомление о
            событии BioAPI_NOTIFY_SOURCE_PRESENT или
            уведомление о событии получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
    </assert_condition>

    <!-- Вызов функции BioSPI_Enroll для создания шаблона. -->
    <invoke function="BioSPI_Enroll">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Purpose"
        var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <output name="NewTemplate"
        setvar="template_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
```

```
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided" >
     <description>
          Функция BioSPI_Enroll возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Повторный вызов функции BioSPI_Capture для получения
вновь созданной ЗБИ. -->
<invoke function="BioSPI_Capture">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_VERIFY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <input name="no_AuditData" value="true"/>
     <output name="CapturedBIR"
     setvar="capturedbir_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Capture возвращает значение
          BioAPI_OK.
     </description>
```

```
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Вызов функции BioSPI_GetHeaderFromHandle с полученной
    ЗБИ в качестве входного параметра. -->
    <invoke function="BioSPI_GetHeaderFromHandle">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Handle" var="capturedbir_handle"/>
        <output name="ProcessedLevel"
        setvar="processedLevel"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_GetHeaderFromHandle возвращает
            значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Инициализировать глобальную переменную
    "_processedbir_handle" значением переменной
    "capturedbir_handle". -->
    <set name="_processedbir_handle" var="capturedbir_handle" />

    <!-- Если полученная ЗБИ имеет промежуточную степень
    обработки, следует вызвать функцию BioSPI_Process. -->
    <invoke activity="process_bir"
```

```
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            break_on_break="true" >
        <only_if>
            <not_equal_to var1="processedLevel"
            var2="__BioAPI_BIR_DATA_TYPE_PROCESSED" />
        </only_if>
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="CapturedBIR_BIRHandle"
        var="capturedbir_handle"/>
</invoke>

<!-- Вызов функции BioSPI_VerifyMatch для верификации
обработанной ЗБИ с использованием сохраненного шаблона. -->
<invoke function="BioSPI_VerifyMatch" >
        <input name="BSPHandle" var="_bsphandle" />
        <input name="MaxFMRRequested" var="maxFMRRequested" />
        <input name="ProcessedBIR_Form"
        var="__BioAPI_BIR_HANDLE_INPUT" />
        <input name="ProcessedBIR_BIRHandle"
        var="_processedbir_handle" />
        <input name="ReferenceTemplate_Form"
        var="__BioAPI_BIR_HANDLE_INPUT" />
        <input name="ReferenceTemplate_BIRHandle"
        var="template_handle" />
        <input name="no_AdaptedBIR" value="true" />
        <input name="no_Payload" value="true" />
        <output name="Result" setvar="result" />
        <output name="FMRAchieved" setvar="fmrAchieved" />
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
```

```
            <description>
                Функция BioSPI_VerifyMatch возвращает значение
                BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.44 Утверждение 12b *BioSPI_VerifyMatch_Payload*

Описание: Проверку данного утверждения проводят с целью испытания поддержки полезной информации в функции *BioSPI_VerifyMatch*. Предполагается, что функция возвратит значение BioAPI_OK.

Выдержки

Подпункт 9.3.4.5

BioAPI_RETURN BioAPI BioSPI_VerifyMatch

(BioAPI_HANDLE BSPHandle,

BioAPI_FMR MaxFMRRequested,

*const BioAPI_INPUT_BIR *ProcessedBIR,*

*const BioAPI_INPUT_BIR *ReferenceTemplate,*

*BioAPI_BIR_HANDLE *AdaptedBIR,*

*BioAPI_BOOL *Result,*

*BioAPI_FMR *FMRAchieved,*

*BioAPI_DATA *Payload).*

Подпункт 8.4.5.1

Данная функция выполняет верификацию, то есть сравнение двух ЗБИ: ProcessedBIR и ReferenceTemplate. ProcessedBIR представляет собой обработанную ЗБИ, созданную для проведения верификации. ReferenceTemplate (контрольный шаблон) создается при регистрации.

Payload (выходной, необязательный параметр) создается в том случае, если с ReferenceTemplate связана полезная информация, которая возвращается в выделенную в памяти структуру BioAPI_DATA, если значение FMRAchieved соответствует политике ПБУ.

Подраздел 7.47

#define BioAPI_PAYLOAD (0x00000080)

Если данное значение установлено, то ПБУ поддерживает перенос полезной информации (сохраняет полезную информацию при выполнении функций *BioAPI_Enroll* или *BioAPI_CreateTemplate* и возвращает ее при успешном выполнении функций *BioAPI_Verify* или *BioAPI_VerifyMatch*).

Ссылки: 9.3.4.5, 8.4.5.1 и 7.47.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll* для создания шаблона.

4) Зарегистрировать вновь созданную ЗБИ.

5) Вызвать функцию *BioSPI_Process* с полученной ЗБИ, если рассматриваемая ЗБИ является промежуточной.

6) Вызвать фукнцию *BioSPI_VerifyMatch* для верификации полученной ЗБИ относительно ЗБИ сохраненного шаблона.

7) Проверить выходной параметр Payload. Предполагается, что значение этого параметра будет соответствовать значению входного параметра функции *BioSPI_Enroll.*

8) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_VerifyMatch* возвращает значение BioAPI_OK. В случае совпадения функция *BioSPI_VerifyMatch* возвратит полезную информацию, аналогичную исходной.

Пакет языка утверждения

```
<package name="692ebe20-ee47-11d9-bd34-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_VerifyMatch_
          Payload" (см. далее элемент <description> утверждения).
     </description>

     <assertion name="BioSPI_VerifyMatch_Payload"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               испытания поддержки полезной информации в функции
               BioSPI_VerifyMatch. Предполагается, что функция
               возвращает значение BioAPI_OK.
               Следующий текст соответствует приведенному в
               8.4.5.1, С.53 и 7.47 спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioSPI_VerifyMatch
                    (BioAPI_HANDLE BSPHandle,
                    BioAPI_FMR MaxFMRRequested,
                    const BioAPI_INPUT_BIR *ProcessedBIR,
                    const BioAPI_INPUT_BIR *ReferenceTemplate,
                    BioAPI_BIR_HANDLE *AdaptedBIR,
                    BioAPI_BOOL *Result,
                    BioAPI_FMR *FMRAchieved,
                    BioAPI_DATA *Payload).
```

Данная функция выполняет верификацию, то есть сравнение двух ЗБИ: ProcessedBIR и ReferenceTemplate. ProcessedBIR представляет собой обработанную ЗБИ, созданную для проведения верификации. ReferenceTemplate (контрольный шаблон) создается при регистрации.

Подраздел 7.47:

#define BioAPI_PAYLOAD (0x00000080)

Если данное значение установлено, то ПБУ поддерживает перенос полезной информации (сохраняет полезную информацию при выполнении функций BioAPI_Enroll или BioAPI_CreateTemplate и возвращает ее при успешном выполнении функции BioAPI_Verify или BioAPI_VerifyMatch).

Подраздел C.5.3:

Полезная информация предоставляется приложению после успешной верификации или идентификации (после выполнения функции BioAPI_Verify или BioAPI_VerifyMatch). ПБУ использует политику предоставления полезной информации только в том случае, если достигнутая фактическая вероятность ошибки ложного совпадения ОЛС меньше порогового значения, установленного в записи ПБУ реестра.

Подпункт 8.4.5.1:

Payload (выходной, необязательный параметр) - если с ReferenceTemplate связана полезная информация, то она возвращается в выделенную в памяти структуру BioAPI_DATA в том случае, если значение FMRAchieved соответствует политике ПБУ.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Enroll для создания шаблона.
4) Зарегистрировать вновь созданную ЗБИ.

```
        5) Вызвать функцию BioSPI_Process с предоставлением
        полученной ЗБИ, если рассматриваемая ЗБИ является
        промежуточной.
        6) Вызвать функцию BioSPI_VerifyMatch для верификации
        полученной ЗБИ относительно ЗБИ сохраненного шаблона.
        7) Проверить выходной параметр Payload.
        Предполагается, что значение этого параметра будет
        соответствовать значению входного параметра функции
        BioSPI_Enroll.
        8) Отсоединить и выгрузить испытуемый ПБУ.
        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>

    <!-- Указание на отсутствие поддержки испытуемым ПБУ
    уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
    <input name="_noSourcePresentSupported" />

    <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
    PRESENT. -->
    <input name="_sourcepresenttimeout"/>

    <!-- Время ожидания для функций BioSPI_Capture и
    BioSPI_Enroll. -->
    <input name="_capturetimeout"/>

    <!-- Порог сопоставления для функции BioSPI_VerifyMatch. -->
```

```
<input name="_maxFMRRequested" />

<!-- Указание на поддержку испытуемым ПБУ полезной
информации. -->
<input name="_payloadSupported"/>

<!-- Политика ПБУ в отношении полезной информации. -->
<input name="_payloadPolicy" />

<!-- Полезная информация. -->
<input name="_payload" />

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_VerifyMatch">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
     <input name="maxFMRRequested"
     var="_maxFMRRequested"/>
     <input name="payloadSupported"
     var="_payloadSupported"/>
     <input name="payloadPolicy" var="_payloadPolicy"/>
     <input name="payload" var="_payload"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом ПБУ.
```

```
        При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
              package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
              function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_VerifyMatch">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>
        <input name="maxFMRRequested" />
        <input name="payloadSupported"/>
        <input name="payloadPolicy" />
        <input name="payload" />

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют.-->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
            <input name="bspVersion" value="32"/>
            <input name="unitIDOrNull" value="0"/>
            <input name="bspHandle" var="_bsphandle"/>
            <input name="eventtimeouttime"
            var="inserttimeouttime"/>
```

```
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности.-->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- Вызов функции BioSPI_Enroll для создания шаблона. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
```

```
        <input name="Timeout" var="capturetimeouttime"/>
        <input name="Payload" var="payload" />
        <output name="NewTemplate"
        setvar="template_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED", в противном
    случае выдается заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
    break_if_false="true">
        <description>
            Функция BioSPI_Enroll возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Повторный вызов функции BioSPI_Capture для получения
    вновь созданной ЗБИ.-->
    <invoke function="BioSPI_Capture">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Purpose"
        var="__BioAPI_PURPOSE_VERIFY"/>
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <input name="no_AuditData" value="true"/>
        <output name="CapturedBIR"
        setvar="capturedbir_handle"/>
        <return setvar="return"/>
    </invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие, определенное далее в элементе <description>, ложное, выдается заключение о соответствии "UNDECIDED" и выполнение процесса прерывается, в противном случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_Capture возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetHeaderFromHandle с полученной ЗБИ в качестве входного параметра. -->
<invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="capturedbir_handle"/>
     <output name="ProcessedLevel"
     setvar="processedLevel"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, определенное далее в элементе <description>, ложное, выдается заключение о соответствии "UNDECIDED" и выполнение процесса прерывается, в противном случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_GetHeaderFromHandle возвращает
          значение BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
```

```
</assert_condition>
<!-- Присвоение глобальной переменной "_processedbir_
handle" значения переменной "capturedbir_handle". -->
<set name="_processedbir_handle" var="capturedbir_handle" />

<!-- Если полученная ЗБИ имеет промежуточную степень
обработки, следует обработать данную ЗБИ. -->
<invoke activity="process_bir"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true" >
     <only_if>
          <not_equal_to var1="processedLevel"
               var2="__BioAPI_BIR_DATA_TYPE_PROCESSED" />
     </only_if>
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="CapturedBIR_BIRHandle"
     var="capturedbir_handle"/>
</invoke>

<!-- Вызов функции BioSPI_VerifyMatch для верификации
обработанной ЗБИ с ЗБИ сохраненного шаблона.-->
<invoke function="BioSPI_VerifyMatch" >
     <input name="BSPHandle" var="_bsphandle" />
     <input name="MaxFMRRequested" var="maxFMRRequested" />
     <input name="ProcessedBIR_Form"
     var="__BioAPI_BIR_HANDLE_INPUT" />
     <input name="ProcessedBIR_BIRHandle"
     var="_processedbir_handle" />
     <input name="ReferenceTemplate_Form"
     var="__BioAPI_BIR_HANDLE_INPUT" />
     <input name="ReferenceTemplate_BIRHandle"
     var="template_handle" />
     <input name="no_AdaptedBIR" value="true" />
     <output name="Payload" setvar="output_payload" />
```

```
        <output name="Result" setvar="result" />
        <output name="FMRAchieved" setvar="fmrAchieved" />
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "FAIL", в противном
    случае выдается заключение о соответствии "PASS". -->
    <assert_condition>
        <description>
            Функция BioSPI_VerifyMatch возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>
    <!-- Проверка возвращаемой полезной информации (если ПБУ
    поддерживает полезную информацию). -->
    <invoke activity="payloadSupport_checkPayload"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            break_on_break="true">
        <only_if>
            <same_as var1="payloadSupported"
            value2="true"/>
        </only_if>
        <input name="inputPayload" var="payload" />
        <input name="outputPayload" var="output_payload"/>
        <input name="result" var="result"/>
        <input name="payloadPolicy" var="payloadPolicy"/>
        <input name="fmrAchieved" var="fmrAchieved"/>
    </invoke>

    <!-- Проверка возвращаемой полезной информации (если ПБУ
    не поддерживает полезную информацию). -->
    <invoke activity="payloadNotSupport_checkPayload"
```

```
                    package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                    break_on_break="true">
               <only_if>
                    <same_as var1="payloadSupported"
                    value2="false"/>
               </only_if>
               <input name="outputPayload" var="output_payload"/>
          </invoke>

          <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
          <invoke activity="DetachAndUnload"
                    package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
               <input name="bspUuid" var="bspUuid" />
               <input name="BSPHandle" var="_bsphandle" />
          </invoke>
     </activity>
</package>
```

8.45 Утверждение 12c *BioSPI_VerifyMatch_Inconsistent_Purpose*

Описание: Проверку данного утверждения проводят с целью испытания функции *BioSPI_VerifyMatch* с предоставлением в качестве входного параметра ЗБИ, назначение которой является недействительным для данной функции. Предполагается, что функция возвратит значение BioAPIERR_BSP_INCONSISTENT_PURPOSE.

Выдержки

Подпункт 9.3.4.5

BioAPI_RETURN BioAPI BioSPI_VerifyMatch

(BioAPI_HANDLE BSPHandle,

BioAPI_FMR MaxFMRRequested,

*const BioAPI_INPUT_BIR *ProcessedBIR,*

*const BioAPI_INPUT_BIR *ReferenceTemplate,*

*BioAPI_BIR_HANDLE *AdaptedBIR,*

*BioAPI_BOOL *Result,*

*BioAPI_FMR *FMRAchieved,*

*BioAPI_DATA *Payload).*

Подпункт 8.4.5.1

Данная функция выполняет верификацию, то есть сравнение двух ЗБИ: ProcessedBIR и ReferenceTemplate. ProcessedBIR представляет собой обработанную ЗБИ, созданную для проведения верификации. ReferenceTemplate создается при регистрации.

Пункт 11.2.3

#define BioAPIERR_INCONSISTENT_PURPOSE (0x000115)

Назначение, указанное в ЗБИ, и требуемое назначение несовместимы с выполняемой функцией.

Ссылки: 9.3.4.5, 8.4.5.1 и 11.2.3.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll* для создания шаблона.

4) Вызвать функцию *BioSPI_Enroll* для создания нового шаблона.

5) Вызвать функцию *BioSPI_VerifyMatch* с предоставлением в качестве входных параметров двух вышеуказанных шаблонов. При этом оба шаблона имеют назначение BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY.

6) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_BSP_INCONSISTENT_PURPOSE.

7) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_VerifyMatch* возвращает значение BioAPIERR_BSP_INCONSISTENT_PURPOSE.

Пакет языка утверждения

```
<package name="9108ec70-2e9b-11d9-9669-0800200c9a66">
    <author>
```

```
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение "BioSPI_VerifyMatch_
        Inconsistent_Purpose" (см. далее элемент <description>
        утверждения).
    </description>

    <assertion name="BioSPI_VerifyMatch_Inconsistent_Purpose"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью испытания
            функции BioSPI_VerifyMatch с предоставлением в качестве
            входного параметра ЗБИ, назначение которой является
            недействительным для данной функции. Предполагается,
            что функция возвратит значение BioAPIERR_BSP_
            INCONSISTENT_PURPOSE.
            Следующий текст соответствует приведенному в 8.4.5.1
            и 11.2.3 спецификации БиоАПИ 2.0.
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_VerifyMatch
                (BioAPI_HANDLE BSPHandle,
                BioAPI_FMR MaxFMRRequested,
                const BioAPI_INPUT_BIR *ProcessedBIR,
                const BioAPI_INPUT_BIR *ReferenceTemplate,
                BioAPI_BIR_HANDLE *AdaptedBIR,
                BioAPI_BOOL *Result,
                BioAPI_FMR *FMRAchieved,
                BioAPI_DATA *Payload).

            Данная функция выполняет верификацию, то есть
            сравнение двух ЗБИ: ProcessedBIR и ReferenceTemplate.
            ProcessedBIR представляет собой обработанную ЗБИ,
            созданную для проведения верификации.
```

```
ReferenceTemplate создается при регистрации.
-------------------------------------------------
Пункт 11.2.3:
#defineBioAPIERR_INCONSISTENT_PURPOSE (0x000115)
Назначение, указанное в ЗБИ, и требуемое назначение
несовместимы с выполняемой функцией.
-------------------------------------------------
Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Enroll для создания шаблона.
4) Вызвать функцию BioSPI_Enroll для создания нового
шаблона.
5) Вызвать функцию BioSPI_VerifyMatch с предоставлением
в качестве входных параметров двух вышеуказанных
шаблонов. При этом оба шаблона имеют назначение
BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY.
6) Проверить возвращаемый код. Предполагается воз-
вращение значения BioAPIERR_BSP_INCONSISTENT_
PURPOSE.
7) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли
неуспешно, то выдается заключение о соответствии
"UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>
```

```
<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Capture. -->
<input name="_capturetimeout"/>

<!-- Порог сопоставления для функции BioSPI_VerifyMatch. -->
<input name="_maxFMRRequested" />

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_VerifyMatch">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
     <input name="maxFMRRequested"
     var="_maxFMRRequested"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически. -->
```

```
    <bind activity="EventHandler"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_VerifyMatch">
    <input name="bspUuid"/>
    <input name="inserttimeouttime"/>
    <input name="nosourcepresentsupported" />
    <input name="sourcepresenttimeouttime"/>
    <input name="capturetimeouttime"/>
    <input name="maxFMRRequested" />

    <!-- Данное утверждение использует BSPHandle со значением
    "1" для всех вызовов BioSPI, которые этого требуют. -->
    <set name="_bsphandle" value="1"/>

    <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
    предоставленных испытуемым ПБУ. Входное значение параметра
    "unitIDOrNull" равно "0", поэтому утверждение будет
    тестировать модуль датчика, выбранный ПБУ. -->
    <invoke activity="LoadAndAttach"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            break_on_break="true">
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
    </invoke>

    <set name="eventtimeoutflag" value="false"/>

    <!-- Если испытуемый ПБУ поддерживает уведомление о событии
    BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
```

```
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- Вызов функции BioSPI_Enroll для создания шаблона. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <output name="NewTemplate" set-
     var="template_handle1"/>
     <return setvar="return"/>
</invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
     <description>
          Функция BioSPI_Enroll возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Повторный вызов функции BioSPI_Enroll для создания
нового шаблона. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <output name="NewTemplate"
     setvar="template_handle2"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
     <description>
          Функция BioSPI_Enroll возвращает значение
          BioAPI_OK.
```

```
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>

    <!-- Вызов функции BioSPI_VerifyMatch с предоставлением в
    качестве входных параметров двух шаблонов. Предполагается,
    что функция завершится возвращением кода ошибки
    BioAPIERR_BSP_INCONSISTENT_PURPOSE. -->
    <invoke function="BioSPI_VerifyMatch" >
        <input name="BSPHandle" var="_bsphandle" />
        <input name="MaxFMRRequested" var="maxFMRRequested" />
        <input name="ProcessedBIR_Form"
        var="__BioAPI_BIR_HANDLE_INPUT" />
        <input name="ProcessedBIR_BIRHandle"
        var="template_handle2" />
        <input name="ReferenceTemplate_Form"
        var="__BioAPI_BIR_HANDLE_INPUT" />
        <input name="ReferenceTemplate_BIRHandle"
        var="template_handle1" />
        <input name="no_AdaptedBIR" value="true" />
        <input name="no_Payload" value="true" />
        <output name="Result" setvar="result" />
        <output name="FMRAchieved" setvar="fmrAchieved" />
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "FAIL", в противном
    случае выдается заключение о соответствии "PASS". -->
    <assert_condition>
        <description>
            Функция BioSPI_VerifyMatch возвращает значение
            BioAPIERR_INCONSISTENT_PURPOSE.
        </description>
```

```
            <equal_to var1="return"
                var2="__BioAPIERR_BSP_INCONSISTENT_PURPOSE"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.46 Утверждение 13a *BioSPI_Enroll_ValidParam*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_Enroll* с достоверными значениями входных параметров.

Выдержки

Подпункт 9.3.4.7

BioAPI_RETURN BioAPI BioSPI_Enroll

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_PURPOSE Purpose,

BioAPI_BIR_SUBTYPE Subtype,

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat*

*const BioAPI_INPUT_BIR *ReferenceTemplate,*

*BioAPI_BIR_HANDLE *NewTemplate,*

*const BioAPI_DATA *Payload,*

int32_t Timeout,

*BioAPI_BIR_HANDLE *AuditData,*

*BioAPI_UUID *TemplateUUID).*

Подпункт 8.4.7.1

Данная функция получает биометрические данные с присоединенного устройства (модуля датчика) для создания ProcessedBIR (обработанной ЗБИ) с целью регистрации.

Раздел А.4

Данную функцию поддерживают как ПБУ верификации, так и ПБУ идентификации.

Ссылки: 9.3.4.7, 8.4.7.1 и А.4.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll* для получения и регистрации ЗБИ.

4) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

5) Вызвать функцию *BioSPI_GetHeaderFromHandle* для проверки степени обработки ЗБИ нового шаблона. Степень обработки должна соответствовать обработанному биометрическому образцу.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_Enroll* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="0b5ebb60-eefb-11d9-990c-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение "BioSPI_Enroll_ Valid-
        Param" (см. далее элемент <description> утверждения).
    </description>
```

```
<assertion name="BioSPI_Enroll_ValidParam" model="BSPTesting">
    <description>
        Проверку данного утверждения проводят с целью
        определения возвращения значения BioAPI_OK при
        вызове функции BioSPI_Enroll с достоверными значениями
        входных параметров.
        Следующий текст соответствует приведенному в А.4 и
        8.4.7.1 спецификации БиоАПИ 2.0.
        ------------------------------------------------
        BioAPI_RETURN BioAPI BioSPI_Enroll
            (BioAPI_HANDLE BSPHandle,
            BioAPI_BIR_PURPOSE Purpose,
            BioAPI_BIR_SUBTYPE Subtype,
            const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
            *OutputFormat,
            const BioAPI_INPUT_BIR *ReferenceTemplate,
            BioAPI_BIR_HANDLE *NewTemplate,
            const BioAPI_DATA *Payload,
            int32_t Timeout,
            BioAPI_BIR_HANDLE *AuditData,
            BioAPI_UUID *TemplateUUID).

        Данная функция получает биометрические данные с
        присоединенного устройства (модуля датчика) для
        создания ProcessedBIR (обработанной ЗБИ) с целью
        регистрации.
        ------------------------------------------------
        Раздел А.4:
        Данную функцию поддерживают как ПБУ верификации, так
        и ПБУ идентификации.
        ------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
```

```
        3) Вызвать функцию BioSPI_Enroll для получения и
        регистрации ЗБИ.
        4) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPI_OK.
        5) Вызвать функцию BioSPI_GetHeaderFromHandle для
        проверки степени обработки ЗБИ нового шаблона. Степень
        обработки должна соответствовать обработанному
        биометрическому образцу.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT.-->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Enroll. -->
<input name="_capturetimeout"/>

<!--Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_Enroll">
```

```
            <input name="bspUuid" var="_bspUuid"/>
            <input name="inserttimeouttime"
            var="_inserttimeout"/>
            <input name="nosourcepresentsupported"
            var="_noSourcePresentSupported" />
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_Enroll">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>
        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
```

```
            break_on_break="true">
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
            setvar="eventtimeoutflag">
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Испытуемый ПБУ не поддерживает уведомление о
            событии BioAPI_NOTIFY_SOURCE_PRESENT или
            уведомление о событии получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
```

```
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <output name="NewTemplate"
     setvar="newtemplate_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
     <description>
          Функция BioSPI_Enroll возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetHeaderFromHandle. -->
     <invoke function="BioSPI_GetHeaderFromHandle">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Handle" var="newtemplate_handle"/>
     <output name="ProcessedLevel"
     setvar="processedLevel"/>
     <return setvar="return"/>
</invoke>
```

```
        <!-- Выдача заключения о соответствии. Если условие, оп-
        ределенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
                break_if_false="true">
            <description>
                Функция BioSPI_GetHeaderFromHandle возвращает
                значение BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Выдача заключения о соответствии. Если условие, оп-
        ределенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Степень обработки зарегистрированной ЗБИ соот-
                ветствует обработанному биометрическому образцу.
            </description>
            <equal_to var1="processedLevel"
            var2="__BioAPI_BIR_DATA_TYPE_PROCESSED"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
```

```
</package>
```

8.47 Утверждение 13b *BioSPI_Enroll_Payload*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_Enroll* с полезной информацией.

Выдержки

Подпункт 9.3.4.7

BioAPI_RETURN BioAPI BioSPI_Enroll

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_PURPOSE Purpose,

BioAPI_BIR_SUBTYPE Subtype,

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*const BioAPI_INPUT_BIR *ReferenceTemplate,*

*BioAPI_BIR_HANDLE *NewTemplate,*

*const BioAPI_DATA *Payload,*

int32_t Timeout,

*BioAPI_BIR_HANDLE *AuditData,*

*BioAPI_UUID *TemplateUUID).*

Подпункт 8.4.7.1

Данная функция получает биометрические данные с присоединенного устройства (модуля датчика) для создания ProcessedBIR с целью регистрации.

Параметры: Payload (входной, необязательный параметр) указывает на данные, которые будут сохранены ПБУ. Данный параметр игнорируется, если указатель содержит значение NULL.

Подпункт А.4.6.2.6

Если функция предусмотрена, то ПБУ должен внести в реестр компонентов максимальный размер поля для полезной информации. Если максимальный

размер равен нулю, то это указывает на то, что перенос полезной информации не поддерживается. Если размер входной полезной информации превышает установленный размер, то функция должна возвратить код ошибки.

Подраздел 7.47

#define BioAPI_PAYLOAD (0x00000080)

Если данное значение установлено, то ПБУ поддерживает перенос полезной информации (сохраняет полезную информацию при выполнении функций *BioAPI_Enroll* или *BioAPI_CreateTemplate* и возвращает ее при успешном выполнении функций *BioAPI_Verify* или *BioAPI_VerifyMatch*).

Ссылки: 9.3.4.7, 8.4.7.1, A.4.6.2.6 и 7.47.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll* для получения и регистрации ЗБИ с полезной информацией, которая отличается от значения NULL.

4) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_Enroll* возвращает значение BioAPI_OK (полезная информация принимается).

Пакет языка утверждения

```
<package name="e8969d40-ef05-11d9-9098-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение "BioSPI_Enroll_Payload"
        (см. далее элемент <description> утверждения).
    </description>
```

```
<assertion name="BioSPI_Enroll_Payload" model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью
          определения возвращаения значения BioAPI_OK при
          вызове функции BioSPI_Enroll с полезной информацией.
          Следующий текст соответствует приведенному в 7.47,
          A.4.6.2.6 и 8.4.7.1 спецификации БиоАПИ 2.0.
          ------------------------------------------------
          BioAPI_RETURN BioAPI BioSPI_Enroll
               (BioAPI_HANDLE BSPHandle,
               BioAPI_BIR_PURPOSE Purpose,
               BioAPI_BIR_SUBTYPE Subtype,
               const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
               *OutputFormat,
               const BioAPI_INPUT_BIR *ReferenceTemplate,
               BioAPI_BIR_HANDLE *NewTemplate,
               const BioAPI_DATA *Payload,
               int32_t Timeout,
               BioAPI_BIR_HANDLE *AuditData,
               BioAPI_UUID *TemplateUUID).

               Данная функция получает биометрические данные с
               присоединенного устройства (модуля датчика) для
               создания ProcessedBIR с целью регистрации.
          ------------------------------------------------
          Подпункт 8.4.7.1:
          Параметры: Payload (входной, необязательный параметр)
          указывает на данные, которые будут сохранены ПБУ.
          Данный параметр игнорируется, если указатель
          содержит значение NULL.
          ------------------------------------------------
          Подпункт A.4.6.2.6
          Если функция предусмотрена, то ПБУ должен внести в
          реестр компонентов максимальный размер поля для
```

```
        полезной информации. Если максимальный размер равен
        нулю, то это указывает на то, что перенос полезной
        информации не поддерживается. Если размер входной
        полезной информации превышает установленный размер,
        то функция должна возвратить код ошибки.
        ------------------------------------------------
        Подраздел 7.47:
        #define BioAPI_PAYLOAD (0x00000080)
        Если данное значение установлено, то ПБУ поддерживает
        перенос полезной информации (сохраняет полезную
        информацию при выполнении функций BioAPI_Enroll или
        BioAPI_CreateTemplate и возвращает ее при успешном
        выполнении функций BioAPI_Verify или
        BioAPI_VerifyMatch).
        ------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Вызвать функцию BioSPI_Enroll для получения и
        регистрации ЗБИ с полезной информацией, которая
        отличается от значения NULL.
        4) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPI_OK.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>
```

```
<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Enroll. -->
<input name="_capturetimeout"/>

<!-- Указание на поддержку испытуемым ПБУ полезной
информации. -->
<input name="_supportPayload" />

<!-- Размер полезной информации не должен превышать
максимальный размер поля для полезной информации, внесенный
ПБУ в реестр компонентов.-->
<input name="_payload" />

<!--Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_Enroll">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
     <input name="supportpayload" var="_supportPayload" />
     <input name="payload" var="_payload" />
```

```
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_Enroll">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>
        <input name="supportpayload" />
        <input name="payload" />

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                  package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                  break_on_break="true">
             <input name="bspUuid" var="bspUuid"/>
             <input name="bspVersion" value="32"/>
             <input name="unitIDOrNull" value="0"/>
             <input name="bspHandle" var="_bsphandle"/>
```

```
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ заявляет о поддержке уведомления
о событии BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать
получения уведомления о событии в пределах указанной
максимальной продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов кполучению ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
```

```
        <input name="Purpose"
        var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <input name="Payload" var="payload" />
        <output name="NewTemplate"
        setvar="newtemplate_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "FAIL" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition break_if_false="true">
        <description>
            Если ПБУ поддерживает полезную информацию, то
            функция BioSPI_Enroll возвращает значение
            BioAPI_OK, в противном случае функция
            BioSPI_Enroll возвращает значение
            BioAPIERR_BSP_UNABLE_TO_STORE_PAYLOAD.
        </description>
        <or>
            <and>
                <equal_to var1="return"
                var2="__BioAPI_OK"/>
                <same_as var1="supportpayload"
                value2="true" />
            </and>
            <and>
                <equal_to var1="return"
                    var2="__BioAPIERR_BSP_UNABLE_TO_STORE_
                    PAYLOAD" />
```

```
                    <same_as var1="supportpayload"
                    value2="false" />
                </and>
            </or>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.48 Утверждение 13c *BioSPI_Enroll_AuditData*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_Enroll* со значением параметра AuditData, отличающимся от NULL. Предполагается, что функция возвратит контрольные данные, если ПБУ поддерживает контрольные данные.

Выдержки

Подпункт 9.3.4.7

BioAPI_RETURN BioAPI BioSPI_Enroll

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_PURPOSE Purpose,

BioAPI_BIR_SUBTYPE Subtype,

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*const BioAPI_INPUT_BIR *ReferenceTemplate,*

*BioAPI_BIR_HANDLE *NewTemplate,*

*const BioAPI_DATA *Payload,*

int32_t Timeout,

*BioAPI_BIR_HANDLE *AuditData,*

*BioAPI_UUID *TemplateUUID).*

Подпункт 8.4.7.1

Данная функция получает биометрические данные с присоединенного устройства (модуля датчика) для создания ProcessedBIR с целью регистрации.

AuditData (выходной, необязательный параметр) – дескриптор ЗБИ, содержащей контрольные биометрические данные, которые могут быть использованы для предоставления биометрических данных личности. Если входным значением указателя является NULL, то это указывает на то, что контрольные данные не собраны. Не все ПБУ поддерживают сбор контрольных данных. ПБУ может возвратить значение дескриптора ЗБИ BioAPI_UNSUPPORTED_BIR_HANDLE для того, чтобы указать, что параметр AuditData не поддерживается или значение BioAPI_INVALID_BIR_HANDLE, чтобы указать, что контрольные данные недоступны.

Подраздел 7.47

#define BioAPI_RAW (0x00000001)

Если установлено данное значение, это указывает на то, что ПБУ поддерживает возврат необработанных (контрольных) данных.

Подпункт А.4.6.2.1

Функции, обеспечивающие получение биометрических данных с датчика, могут дополнительно поддерживать возврат исходных данных для визуализации или контроля. Если данная функция поддерживает возврат исходных данных, то выходной параметр AuditData должен содержать указатель этих данных. Если возврат исходных данных не поддерживается, то ПБУ возвращает значение "–1".

Ссылки: 9.3.4.7, 8.4.7.1, 7.47 и А.4.6.2.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll* для получения и регистрации ЗБИ с параметром AuditData, которому присвоено значение, отличающееся от NULL.

4) Проверить возвращаемый код. Если контрольные данные поддерживаются, то предполагается, что возвращаемый дескриптор ЗБИ будет иметь значение BioAPI_UNSUPPORTED_BIR_HANDLE. Если контрольные данные не поддерживаются, то предполагается, что возвращаемый дескриптор будет иметь значение BioAPI_INVALID_BIR_HANDLE.

5) Вызвать функцию *BioSPI_GetHeaderFromHandle* для проверки типа контрольных данных, если эти данные поддерживаются.Предполагается, что типом контрольных данных будет RAW (необработанные данные, исходный биометрический образец).

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Если контрольные данные поддерживаются, то вызов функции BioSPI_Enroll возвращает ЗБИ контрольных данных, которая имеет степень обработки, соответствующую исходному биометрическому образцу.

Пакет языка утверждения

```
<package name="b40a5260-ef14-11d9-a4fe-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_Enroll_AuditData" (см. далее элемент
          <description> утверждения).
     </description>

     <assertion name="BioSPI_Enroll_AuditData" model="BSPTesting">
          <description>
```

Проверку данного утверждения проводят с целью вызова функции BioSPI_Enroll со значением параметра AuditData, отличающимся от NULL. Предполагается, что функция возвращает контрольные данные, если ПБУ поддерживает контрольные данные.

Следующий текст соответствует приведенному в 8.4.7.1 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_Enroll
    (BioAPI_HANDLE BSPHandle,
    BioAPI_BIR_PURPOSE Purpose,
    BioAPI_BIR_SUBTYPE Subtype,
    const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
    *OutputFormat,
    const BioAPI_INPUT_BIR *ReferenceTemplate,
    BioAPI_BIR_HANDLE *NewTemplate,
    const BioAPI_DATA *Payload,
    int32_t Timeout,
    BioAPI_BIR_HANDLE *AuditData,
    BioAPI_UUID *TemplateUUID).
```

Данная функция получает биометрические данные с присоединенного устройства (модуля датчика) для создания ProcessedBIR с целью регистрации.

Подпункт 8.4.7.1:

AuditData (выходной, необязательный параметр) - дескриптор ЗБИ, содержащей контрольные биометрические данные, которые могут использоваться для предоставления биометрических данных личности. Если входным значением указателя является NULL, то это указывает на то, что контрольные данные не собраны. Не все ПБУ поддерживают сбор контрольных данных. ПБУ может возвратить значение дескриптора ЗБИ BioAPI_UNSUPPORTED_BIR_HANDLE для того, чтобы указать, что параметр AuditData не

поддерживается, или значение BioAPI_INVALID_BIR_HANDLE, чтобы указать, что контрольные данные недоступны.

Подраздел 7.47: BioAPI_OPTIONS_MASK
#define BioAPI_RAW (0x00000001)
Если данное значение установлено, это указывает на то, что ПБУ поддерживает возврат необработанных (контрольных) данных.

Подпункт А.4.6.2.1
Функции, обеспечивающие получение биометрических данных от датчика, могут дополнительно поддерживать возврат исходных данных для визуализации или контроля. Если данная функция поддерживает возврат исходных данных, то выходной параметр AuditData должен содержать указатель этих данных. Если возврат исходных данных не поддерживается, то ПБУ возвращает значение "-1".

Порядок действий при испытании на соответствие:
1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Enroll для получения и регистрации ЗБИ с параметром AuditData, которому присвоено значение, отличающееся от NULL.
4) Проверить возвращаемый код. Если контрольные данные поддерживаются, то предполагается, что возвращаемый дескриптор ЗБИ будет иметь значение BioAPI_UNSUPPORTED_BIR_HANDLE. Если контрольные данные не поддерживаются, то предполагается, что возвращаемый дескриптор будет иметь значение BioAPI_INVALID_BIR_HANDLE.
5) Вызвать функцию BioSPI_GetHeaderFromHandle для проверки типа контрольных данных, если эти данные поддерживаются.Предполагается, что типом контольных

```
        данных будет RAW (необработанные данные, исходный
        биометрический образец).

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".

    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>

    <!-- Указание на отсутствие поддержки испытуемым ПБУ
    уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
    <input name="_noSourcePresentSupported" />

    <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
    PRESENT. -->
    <input name="_sourcepresenttimeout"/>

    <!-- Время ожидания для функции BioSPI_Enroll. -->
    <input name="_capturetimeout"/>

    <!-- Указание на поддержку испытуемым ПБУ контрольных
    данных. -->
    <input name="_supportAuditData" />

    <!-- Вызов основного процесса данного утверждения со
    значениями входных параметров, определенными значениями
    параметров утверждения. -->
    <invoke activity="BioSPI_Enroll">
        <input name="bspUuid" var="_bspUuid"/>
```

```
            <input name="inserttimeouttime"
            var="_inserttimeout"/>
            <input name="nosourcepresentsupported"
            var="_noSourcePresentSupported" />
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
            <input name="supportAuditData"
            var="_supportAuditData" />
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_Enroll">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>
        <input name="supportAuditData" />

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
```

```
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
     package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
     break_on_break="true">
          <input name="bspUuid" var="bspUuid"/>
          <input name="bspVersion" value="32"/>
          <input name="unitIDOrNull" value="0"/>
          <input name="bspHandle" var="_bsphandle"/>
          <input name="eventtimeouttime"
          var="inserttimeouttime"/>
     </invoke>

     <set name="eventtimeoutflag" value="false"/>

     <!--Если испытуемый ПБУ поддерживает уведомление о
     событии BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать
     получения уведомления о событии в пределах указанной
     максимальной продолжительности. -->
     <wait_until timeout_var="sourcepresenttimeouttime"
               setvar="eventtimeoutflag">
          <or var1="nosourcepresentsupported"
          var2="_sourcePresent" />
     </wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
```

```
            уведомление о событии получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Purpose"
        var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <output name="AuditData" setvar="auditbir_handle" />
        <output name="NewTemplate"
        setvar="newtemplate_handle"/>
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_Enroll возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
```

```
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition break_if_false="true">
     <description>
          Если контрольные данные поддерживаются, то
          выходной дескриптор ЗБИ AuditData является или
          достоверным дескриптором ЗБИ, или значением
          BioAPI_INVALID_BIR_HANDLE. Если контрольные
          данные не поддерживаются, то выходной дескриптор
          ЗБИ AuditDataBIR является значением BioAPI_
          UNSUPPORTED_BIR_HANDLE.
     </description>
     <or>
          <and>
               <same_as var1="supportAuditData"
               value2="true" />
               <or>
                    <equal_to var1="auditbir_handle"
                         var2="__BioAPI_INVALID_BIR_HANDLE" />
                    <greater_than_or_equal_to
                    var1="auditbir_handle" value2="0" />
               </or>
          </and>
          <and>
               <same_as var1="supportAuditData"
               value2="false" />
               <equal_to var1="auditbir_handle"
                    var2="__BioAPI_UNSUPPORTED_BIR_HANDLE" />
          </and>
     </or>
</assert_condition>

<!-- Проверка степени обработки контрольных данных ЗБИ. -->
<invoke activity="check_audit_data_type" >
```

```
            <only_if>
                <same_as var1="supportAuditData" value2="true" />
                <greater_than_or_equal_to
                var1="auditbir_handle" value2="0" />
            </only_if>
            <input name="BSPHandle" var="_bsphandle"/>
            <input name="Handle" var="auditbir_handle"/>
        </invoke>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>

    <activity name="check_audit_data_type" >
        <input name="BSPHandle" />
        <input name="Handle" />
        <!-- Вызов функции BioSPI_GetHeaderFromHandle. -->
            <invoke function="BioSPI_GetHeaderFromHandle">
            <input name="BSPHandle" var="_bsphandle"/>
            <input name="Handle" var="Handle"/>
            <output name="ProcessedLevel"
            setvar="processedLevel"/>
            <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
                break_if_false="true">
```

```
            <description>
                Функция BioSPI_GetHeaderFromHandle возвращает
                значение BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Степень обработки контрольных данных ЗБИ соот-
                ветствует исходному биометрическому образцу.
            </description>
            <equal_to var1="processedLevel"
            var2="__BioAPI_BIR_DATA_TYPE_RAW"/>
        </assert_condition>
    </activity>
</package>
```

8.49 Утверждение 13d *BioSPI_Enroll_BIRHeaderQuality*

Описание: Проверку данного утверждения проводят с целью определения возвращения достоверного значения качества в заголовке нового шаблона ЗБИ при вызове функции *BioSPI_Enroll*.

Выдержки

Подпункт 9.3.4.7

BioAPI_RETURN BioAPI BioSPI_Enroll

(BioAPI_HANDLE BSPHandle,

BioAPI_BIR_PURPOSE Purpose,

BioAPI_BIR_SUBTYPE Subtype,

*const BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,*

*const BioAPI_INPUT_BIR *ReferenceTemplate,*

*BioAPI_BIR_HANDLE *NewTemplate,*

*const BioAPI_DATA *Payload,*

int32_t Timeout,

*BioAPI_BIR_HANDLE *AuditData,*

*BioAPI_UUID *TemplateUUID).*

Подпункт 8.4.7.1

Данная функция получает биометрические данные с присоединенного устройства (модуля датчика) для создания ProcessedBIR с целью регистрации.

Подпункт 7.49.3

Результаты измерения качества возвращаются как целые числа в диапазоне от 0 до 100, кроме следующих значений:

Значение "–1": BioAPI_QUALITY не было установлено ПБУ (соответствующая информация должна быть приведена в документации разработчика ПБУ).

Значение "–2": BioAPI_QUALITY не поддерживается ПБУ.

Подраздел 7.47

#define BioAPI_QUALITY_INTERMEDIATE (0x00000004)

Если данное значение установлено, то ПБУ поддерживает возврат значения качества (в заголовке ЗБИ) для промежуточных биометрических данных.

#define BioAPI_QUALITY_PROCESSED (0x00000008)

Если данное значение установлено, то ПБУ поддерживает возврат значения качества (в заголовке ЗБИ) для обработанных биометрических данных.

Подпункт А.4.6.2.2

Возвращаемое значение качества

После получения биометрических данных с датчика ПБУ может вычислить относительное значение качества для полученных биометрических данных, которое будет включено в заголовок возвращаемой CapturedBIR (и необязательной AuditData). Если функция возврата значения качества поддерживается, то поле заголовка будет содержать положительное значение от 1 до 100. Если

функция возврата значения качества не поддерживается, то значение поля будет установлено на "–2". Значение устанавливается в процессе выполнения функций *BioSPI_Capture* и *BioSPI_Enroll*.

После обработки ЗБИ может быть выполнено повторное вычисление значения качества, которое может быть включено в заголовок processedBIR (вновь созданная обработанная ЗБИ), и, кроме того, в заголовок AdaptedBIR (адаптированная ЗБИ). Это происходит в процессе выполнения функций *BioAPI_CreateTemplate*, *BioAPI_Process*, *BioAPI_Verify*, *BioAPI_VerifyMatch*, *BioAPI_Enroll* и *BioAPI_Import* (ConstructedBIR – ЗБИ, созданная из импортированных биометрических данных).

ПБУ должен ввести в реестр компонентов информацию о том, поддерживает ли он вычисление качества для ЗБИ каждого типа: исходной, промежуточной и обработанной.

Ссылки: 9.3.4.7, 8.4.7.1, 7.49.3, 7.47 и A.4.6.2.2.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll*.

4) Вызвать функцию *BioSPI_GetHeaderFromHandle* для ЗБИ нового шаблона. Проверить поле Quality (значение качества). Предполагается, что значение качества будет находиться в диапазоне от 0 до 100.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: ЗБИ, созданная функцией *BioSPI_Enroll* имеет достоверное значение в поле качества.

Пакет языка утверждения

```
<package name="6f727320-ef1a-11d9-9143-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>
```

```
<description>
     Данный пакет содержит утверждение
     "BioSPI_Enroll_BIRHeaderQuality" (см. далее элемент
     <description> утверждения).
</description>

<assertion name="BioSPI_Enroll_BIRHeaderQuality"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью
          определения возвращения достоверного значения качества
          в заголовке нового шаблона ЗБИ при вызове функции
          BioSPI_Enroll.
          Следующий текст соответствует приведенному в 7.49.3
          и 8.4.7.1 спецификации БиоАПИ 2.0.
          ---------------------------------------------------
          BioAPI_RETURN BioAPI BioSPI_Enroll
               (BioAPI_HANDLE BSPHandle,
               BioAPI_BIR_PURPOSE Purpose,
               BioAPI_BIR_SUBTYPE Subtype,
               const BioAPI_BIR_BIOMETRIC_DATA_FORMAT
               *OutputFormat,
               const BioAPI_INPUT_BIR *ReferenceTemplate,
               BioAPI_BIR_HANDLE *NewTemplate,
               const BioAPI_DATA *Payload,
               int32_t Timeout,
               BioAPI_BIR_HANDLE *AuditData,
               BioAPI_UUID *TemplateUUID).

          Данная функция получает биометрические данные с
          присоединенного устройства (модуля датчика) для
          создания ProcessedBIR с целью регистрации.

          Подпункт 7.49.3:
```

Результаты измерения качества возвращаются как целые числа в диапазоне от 0 до 100, за исключением следующих случаев:
Значение "-1": BioAPI_QUALITY не было установлено ПБУ (соответствующая информация должна быть приведена в документации разработчика ПБУ).
Значение "-2": BioAPI_QUALITY не поддерживается ПБУ.

Подраздел 7.47:
#define BioAPI_QUALITY_INTERMEDIATE (0x00000004)
Если данное значение установлено, то ПБУ поддерживает возврат значения качества (в заголовке ЗБИ) для промежуточных биометрических данных.
#define BioAPI_QUALITY_PROCESSED (0x00000008)
Если данное значение установлено, то ПБУ поддерживает возврат значения качества (в заголовке ЗБИ) для обработанных биометрических данных.

Подпункт А.4.6.2.2:
Возвращаемое значение качества
После получения биометрических данных с датчика ПБУ может вычислить относительное значение качества для полученных биометрических данных, которое будет включено в заголовок возвращаемой CapturedBIR (и необязательной AuditData). Если функция возврата значения качества поддерживается, то поле заголовка будет содержать положительное значение в диапазоне от 1 до 100. Если функция возврата значения качества не поддерживается, то значение поля
будет установлено на "-2". Значение устанавливается в процессе выполнения функций BioSPI_Capture и BioSPI_Enroll.
После обработки ЗБИ может быть выполнено повторное вычисление значениякачества, которое может быть включено в заголовок processedBIR (вновь созданная обработанная ЗБИ), и, кроме того, в заголовок AdaptedBIR

```
      (адаптированная ЗБИ). Это происходит в процессе
      выполнения функций BioAPI_CreateTemplate,
      BioAPI_Process, BioAPI_Verify, BioAPI_VerifyMatch,
      BioAPI_Enroll, и BioAPI_Import (ConstructedBIR - ЗБИ,
      созданная из импортированных биометрических данных).
      ---------------------------------------------------
      Порядок действий при испытании на соответствие:

      1) Загрузить испытуемый ПБУ.
      2) Присоединить испытуемый ПБУ.
      3) Вызвать функцию BioSPI_Enroll.
      4) Вызвать функцию BioSPI_GetHeaderFromHandle для
      ЗБИ нового шаблона. Проверить поле Quality (значение
      качества). Предполагается, что значение качества
      будет находиться в диапазоне от 0 до 100.

      Если какие-либо промежуточные операции прошли
      неуспешно, то выдается заключение о соответствии
      "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Enroll. -->
```

```
        <input name="_capturetimeout"/>

        <!-- Указание на поддержку испытуемым ПБУ возвращения
        значения качества в обработанной ЗБИ. -->
        <input name="_processedQualitySupported"/>

        <!--Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_Enroll">
             <input name="bspUuid" var="_bspUuid"/>
             <input name="inserttimeouttime"
             var="_inserttimeout"/>
             <input name="nosourcepresentsupported"
             var="_noSourcePresentSupported"/>
             <input name="sourcepresenttimeouttime"
             var="_sourcepresenttimeout"/>
             <input name="capturetimeouttime"
             var="_capturetimeout"/>
             <input name="processedQualitySupported"
             var="_processedQualitySupported"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_Enroll">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
```

```
<input name="nosourcepresentsupported"/>
<input name="sourcepresenttimeouttime"/>
<input name="capturetimeouttime"/>
<input name="processedQualitySupported"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent"/>
</wait_until>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <output name="NewTemplate"
     setvar="newtemplate_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается заклю-
чение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
```

```
            break_if_false="true">
        <description>
            Функция BioSPI_Enroll возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_GetHeaderFromHandle для проверки
степени обработки нового шаблона ЗБИ. -->
<invoke function="BioSPI_GetHeaderFromHandle">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Handle" var="newtemplate_handle"/>
        <output name="Quality" setvar="quality"/>
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_GetHeaderFromHandle возвращает
            значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<invoke activity="check_quality_supported"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            break_on_break="true">
        <only_if>
```

```
                    <same_as var1="processedQualitySupported"
                    value2="true" />
               </only_if>
               <input name="quality" var="quality" />
          </invoke>

          <invoke activity="check_quality_not_supported"
                    package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                    break_on_break="true" >
               <only_if>
                    <same_as var1="processedQualitySupported"
                    value2="false" />
               </only_if>
               <input name="quality" var="quality" />
          </invoke>

          <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
          <invoke activity="DetachAndUnload"
                    package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
               <input name="bspUuid" var="bspUuid"/>
               <input name="BSPHandle" var="_bsphandle"/>
          </invoke>
     </activity>
</package>
```

8.50 Утверждение 14а *BioSPI_Verify_ValidParam*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_Verify* с достоверными значениями входных параметров.

Выдержки

Подпункт 9.3.4.8

BioAPI_RETURN BioAPI BioSPI_Verify

(BioAPI_HANDLE BSPHandle,

BioAPI_FMR MaxFMRRequested,

*const BioAPI_INPUT_BIR *ReferenceTemplate,*

BioAPI_BIR_SUBTYPE Subtype,

*BioAPI_BIR_HANDLE *AdaptedBIR,*

*BioAPI_BOOL *Result,*

*BioAPI_FMR *FMRAchieved,*

*BioAPI_DATA *Payload,*

int32_t Timeout,

*BioAPI_BIR_HANDLE *AuditData).*

Подпункт 8.4.8.1

Данная функция получает биометрические данные с присоединенного устройства (модуля датчика) и сравнивает их с контрольным шаблоном ReferenceTemplate.

Приложение должно запросить максимальное значение критерия ОЛС (пороговое значение) для успешного сопоставления. Булево значение Result указывает, успешно или нет былапроведена верификация, а параметр FMRAchieved представляет собой значение ОЛС, определяющее степень близости сопоставляемых ЗБИ.

Раздел А.4

Данная функция должна поддерживаться как ПБУ верификации, так и ПБУ идентификации.

Ссылки: 9.3.4.8, 8.4.8.1 и А.4.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll* для создания шаблона.

4) Вызвать функцию *BioSPI_Verify* без адаптации и контрольных данных.

5) Проверить возвращаемый код.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_Verify* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="b78e5be0-efcb-11d9-b2c7-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_Verify_ValidParam" (см. далее элемент
          <description> утверждения).
     </description>

     <assertion name="BioSPI_Verify_ValidParam" model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               определения возвращения значения BioAPI_OK при
               вызове функции BioSPI_Verify с достоверными
               значениями входных параметров.
               Следующий текст соответствует приведенному в А.4 и
               8.4.8.1 спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioSPI_Verify
                    (BioAPI_HANDLE BSPHandle,
                    BioAPI_FMR MaxFMRRequested,
                    const BioAPI_INPUT_BIR *ReferenceTemplate,
                    BioAPI_BIR_SUBTYPE Subtype,
                    BioAPI_BIR_HANDLE *AdaptedBIR,
                    BioAPI_BOOL *Result,
                    BioAPI_FMR *FMRAchieved,
                    BioAPI_DATA *Payload,
                    int32_t Timeout,
                    BioAPI_BIR_HANDLE *AuditData).
```

```
            Данная функция получает биометрические данные с
            присоединенного устройства (модуля датчика) и
            сравнивает их с контрольным шаблоном Reference
            Template.
            Приложение должно запросить максимальное значение
            критерия ОЛС (пороговое значение) для успешного
            сопоставления. Булево значение Result указывает,
            успешно или нет была проведена верификация, а
            параметр FMRAchieved представляет собой значе-
            ние ОЛС, определяющее степень близости сопос-
            тавляемых ЗБИ.
        ------------------------------------------------------
        Раздел А.4:
        Данная функция должна поддерживаться как ПБУ
        верификации, так и ПБУ идентификации.
        ------------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Вызвать функцию BioSPI_Enroll для создания шаблона.
        4) Вызвать функцию BioSPI_Verify без адаптации и
        контрольных данных.
        5) Проверить возвращаемый код.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>
```

```
<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Enroll. -->
<input name="_capturetimeout"/>

<!-- Порог сопоставления для функции BioSPI_Verify. -->
<input name="_maxFMRRequested"/>

<!-- Время ожидания для функции BioSPI_Verify. -->
<input name="_verifytimeout"/>

<!--Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_Verify">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported"/>
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeouttime"
     var="_capturetimeout"/>
     <input name="maxFMRRequested"
     var="_maxFMRRequested"/>
     <input name="verifytimeout" var="_verifytimeout"/>
</invoke>
```

```
    <!-- Процесс связан с функцией интерфейса обратного вызова
    инфраструктуры, предоставленной испытуемым компонентом
    ПБУ. При каждом входящем вызове функции связанный с ней
    процесс будет вызываться автоматически. -->
    <bind activity="EventHandler"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_Verify">
    <input name="bspUuid"/>
    <input name="inserttimeouttime"/>
    <input name="nosourcepresentsupported"/>
    <input name="sourcepresenttimeouttime"/>
    <input name="capturetimeouttime"/>
    <input name="maxFMRRequested"/>
    <input name="verifytimeout"/>

    <!-- Данное утверждение использует BSPHandle со значением
    "1" для всех вызовов BioSPI, которые этого требуют. -->
    <set name="_bsphandle" value="1"/>

    <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
    предоставленных испытуемым ПБУ. Входное значение параметра
    "unitIDOrNull" равно "0", поэтому утверждение будет
    тестировать модуль датчика, выбранный ПБУ. -->
    <invoke activity="LoadAndAttach"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            break_on_break="true">
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
    </invoke>
```

```
<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent"/>
</wait_until>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- Вызов функции BioSPI_Enroll для создания шаблона.-->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
```

```
        <output name="NewTemplate"
        setvar="template_handle"/>
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided" >
        <description>
                Функция BioSPI_Enroll возвращает значение
                BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_Verify для верификации с вновь
созданным шаблоном. -->
<invoke function="BioSPI_Verify" >
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="MaxFMRRequested"
        var="maxFMRRequested"/>
        <input name="ReferenceTemplate_Form"
        var="__BioAPI_BIR_HANDLE_INPUT"/>
        <input name="ReferenceTemplate_BIRHandle"
        var="template_handle"/>
        <input name="Subtype" value="0"/>
        <input name="no_AdaptedBIR" value="true"/>
        <input name="Timeout" var="verifytimeout"/>
        <input name="no_AuditData" value="true"/>
        <input name="no_Result" value="false"/>
        <input name="no_Payload" value="false"/>
        <output name="Result" setvar="result"/>
        <output name="FMRAchieved" setvar="fmrAchieved"/>
        <output name="Payload" setvar="payload"/>
```

```
            <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Функция BioSPI_Verify возвращает значение
                BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid"/>
            <input name="BSPHandle" var="_bsphandle"/>
        </invoke>
    </activity>
</package>
```

8.51 Утверждение 14b*BioSPI_Verify_Payload*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_Verify* с предоставлением полезной информации.

Выдержки

Подпункт 9.3.4.8

BioAPI_RETURN BioAPI BioSPI_Verify

(BioAPI_HANDLE BSPHandle,

BioAPI_FMR MaxFMRRequested,

*const BioAPI_INPUT_BIR *ReferenceTemplate,*

BioAPI_BIR_SUBTYPE Subtype,

*BioAPI_BIR_HANDLE *AdaptedBIR,*

*BioAPI_BOOL *Result,*

*BioAPI_FMR *FMRAchieved,*

*BioAPI_DATA *Payload,*

int32_t Timeout,

*BioAPI_BIR_HANDLE *AuditData).*

Подпункт 8.4.8.1

Данная функция получает биометрические данные с присоединенного устройства (модуля датчика) и сравнивает их с контрольным шаблоном ReferenceTemplate.

Приложение должно запросить максимальное значение критерия ОЛС (пороговое значение) для успешного сопоставления. Булево значение Result указывает, успешно или нет была проведена верификация, а параметр FMRAchieved представляет собой значение ОЛС, определяющее степень близости сопоставляемых ЗБИ.

Параметры: Payload (выходной, необязательный параметр) – если с ReferenceTemplate связана полезная информация, то она возвращается в выделенную в памяти структуру BioAPI_DATA в том случае, если значение FMRAchieved соответствует политике ПБУ.

Подраздел 7.47

#define BioAPI_PAYLOAD (0x00000080)

Если данное значение установлено, то ПБУ поддерживает перенос полезной информации (сохраняет полезную информацию при выполнении функций *BioAPI_Enroll* или *BioAPI_CreateTemplate* и возвращает ее при успешном выполнении функции *BioAPI_Verify* или *BioAPI_VerifyMatch*).

Подпункт А.4.6.2.6

Если функция предусмотрена, то ПБУ должен ввести в реестр компонентов максимальный размер поля для полезной информации. Если размер входной

полезной информации превышает установленный размер, то функция должна возвратить код ошибки.

Ссылки: 9.3.4.8, 8.4.8.1, 7.47 и A.4.6.2.6.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll* для создания шаблона с использованием предоставленной полезной информации.

4) Вызвать функцию *BioSPI_Verify* для верификации полученной ЗБИ на основе сохраненной ЗБИ шаблона.

5) Проверить выходной параметр Payload (полезная информация). Предполагается, что значение этого параметра совпадает со значением входного параметра Payload функции *BioSPI_CreateTemplate*.

6) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_Verify* возвращает значение BioAPI_OK и возвращаемая полезная информация совпадает с исходной полезной информацией.

Пакет языка утверждения

```
<package name="32969ec0-eff8-11d9-9831-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение "BioSPI_Verify_Payload"
        (см. далее элемент <description> утверждения).
    </description>

    <assertion name="BioSPI_Verify_Payload" model="BSPTesting">
        <description>
```

Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции BioSPI_Verify с предоставлением полезной информации.

Следующий текст соответствует приведенному в 7.47, А.4.6.2.6 и 8.4.8.1 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_Verify
     (BioAPI_HANDLE BSPHandle,
     BioAPI_FMR MaxFMRRequested,
     const BioAPI_INPUT_BIR *ReferenceTemplate,
     BioAPI_BIR_SUBTYPE Subtype,
     BioAPI_BIR_HANDLE *AdaptedBIR,
     BioAPI_BOOL *Result,
     BioAPI_FMR *FMRAchieved,
     BioAPI_DATA *Payload,
     int32_t Timeout,
     BioAPI_BIR_HANDLE *AuditData).
```

Данная функция получает биометрические данные с присоединенного устройства (модуля датчика) и сравнивает их с контрольным шаблоном ReferenceTemplate. Приложение должно запросить максимальное значение критерия ОЛС (пороговое значение) для успешного сопоставления. Булево значение Result указывает, успешно или нет была проведена верификация, а параметр FMRAchieved представляет собой значение ОЛС, определяющее степень близости сопоставляемых ЗБИ.

Подпункт 8.4.8.1:

Параметры: Payload (выходной, необязательный параметр) – если с ReferenceTemplate связана полезная информация, то она возвращается в выделенную в памяти структуру BioAPI_DATA в том случае, если значение FMRAchieved соответствует политике ПБУ.

Подраздел 7.47:
#define BioAPI_PAYLOAD (0x00000080)
Если данное значение установлено, то ПБУ поддерживает перенос полезной информации (сохраняет полезную информацию при выполнении функций BioAPI_Enroll или BioAPI_CreateTemplate и возвращает ее при успешном выполнении функций BioAPI_Verify или BioAPI_VerifyMatch).

Подпункт А.4.6.2.6
Если функция предусмотрена, то ПБУ должен ввести в реестр компонентов максимальный размер поля для полезной информации. Если размер входной полезной информации превышает установленный размер, то функция должна возвратить код ошибки.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Enroll для создания шаблона с использованием предоставленной полезной информации.
4) Вызвать функцию BioSPI_Verify для верификации полученной ЗБИ на основе сохраненной ЗБИ шаблона.
5) Проверить выходной параметр Payload (полезная информация). Предполагается, что значение этого параметра совпадает со значением входного параметра Payload функции BioSPI_CreateTemplate.
6) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".
</description>

```
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для функции BioSPI_Enroll. -->
<input name="_capturetimeout"/>

<!-- Порог сопоставления для функции BioSPI_Verify. -->
<input name="_maxFMRRequested"/>

<!-- Время ожидания для функции BioSPI_Verify. -->
<input name="_verifytimeout"/>

<!-- Указание на поддержку испытуемым ПБУ полезной
информации.-->
<input name="_payloadSupported"/>

<!-- Пороговое значение (минимальное значение ОЛС),
используемое для принятия решения о выдаче полезной
информации после успешной верификации. -->
<input name="_payloadPolicy"/>

<!-- Полезная информация должна соответствовать
ограничению к размеру полезной информации, установленному
ПБУ в реестре компонента. -->
```

```
        <input name="_payload"/>
        <!--Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_Verify">
             <input name="bspUuid" var="_bspUuid"/>
             <input name="inserttimeouttime"
             var="_inserttimeout"/>
             <input name="nosourcepresentsupported"
             var="_noSourcePresentSupported"/>
             <input name="sourcepresenttimeouttime"
             var="_sourcepresenttimeout"/>
             <input name="capturetimeouttime"
             var="_capturetimeout"/>
             <input name="maxFMRRequested"
             var="_maxFMRRequested"/>
             <input name="verifytimeout" var="_verifytimeout"/>
             <input name="payloadSupported"
             var="_payloadSupported"/>
             <input name="payloadPolicy" var="_payloadPolicy"/>
             <input name="payload" var="_payload"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
   </assertion>

   <activity name="BioSPI_Verify">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
```

```
<input name="nosourcepresentsupported"/>
<input name="sourcepresenttimeouttime"/>
<input name="capturetimeouttime"/>
<input name="maxFMRRequested"/>
<input name="verifytimeout"/>
<input name="payloadSupported"/>
<input name="payloadPolicy"/>
<input name="payload"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
```

```
        <or var1="nosourcepresentsupported"
        var2="_sourcePresent"/>
    </wait_until>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Испытуемый ПБУ не поддерживает уведомление о
            событии BioAPI_NOTIFY_SOURCE_PRESENT или
            уведомление о событии получено в пределах
            указанной максимальной продолжительности.
        </description>
        <not var="eventtimeoutflag"/>
    </assert_condition>

    <!-- Вызов функции BioSPI_Enroll для создания шаблона.-->
    <invoke function="BioSPI_Enroll">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="Purpose"
        var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
        <input name="Subtype" value="0"/>
        <input name="Timeout" var="capturetimeouttime"/>
        <input name="Payload" var="payload"/>
        <output name="NewTemplate"
        setvar="template_handle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
```

```
выдается заключение о соответствии "UNDECIDED", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided" >
     <description>
          Функция BioSPI_Enroll возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_Verify для регистрации ЗБИ и ее
верификации с использованием сохраненного шаблона. -->
<invoke function="BioSPI_Verify" >
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="MaxFMRRequested"
     var="maxFMRRequested"/>
     <input name="ReferenceTemplate_Form"
     var="__BioAPI_BIR_HANDLE_INPUT"/>
     <input name="ReferenceTemplate_BIRHandle"
     var="template_handle"/>
     <input name="Subtype" value="0"/>
     <input name="no_AdaptedBIR" value="true"/>
     <input name="Timeout" var="verifytimeout"/>
     <input name="no_AuditData" value="true"/>
     <input name="no_Result" value="false"/>
     <input name="no_Payload" value="false"/>
     <output name="Result" setvar="result"/>
     <output name="AdaptedBIR"
     setvar="adaptedbir_handle"/>
     <output name="FMRAchieved" setvar="fmrAchieved"/>
     <output name="Payload" setvar="output_payload"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
```

```
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true" >
     <description>
          Функция BioSPI_Verify возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Проверка значения выходного параметра "payload"
(если ПБУ поддерживает полезную информацию). -->
<invoke activity="payloadSupport_checkPayload"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <only_if>
          <same_as var1="payloadSupported"
          value2="true"/>
     </only_if>
     <input name="inputPayload" var="payload" />
     <input name="outputPayload" var="output_payload"/>
     <input name="result" var="result"/>
     <input name="payloadPolicy" var="payloadPolicy"/>
     <input name="fmrAchieved" var="fmrAchieved"/>
</invoke>

<!-- Проверить значение выходного праметра "payload"
(если ПБУ не поддерживает полезную информацию). -->
<invoke activity="payloadNotSupport_checkPayload"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <only_if>
          <same_as var1="payloadSupported" value2="false" />
```

```
            </only_if>
            <input name="outputPayload" var="output_payload" />
        </invoke>

        <!-- Вызов функций BioSPI_BSPDetach and BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid"/>
            <input name="BSPHandle" var="_bsphandle"/>
        </invoke>
    </activity>
</package>
```

8.52 Утверждение 14c *BioSPI_Verify_AuditData*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_Verify* с предоставлением контрольных данных.

Выдержки

Подпункт 9.3.4.8

BioAPI_RETURN BioAPI BioSPI_Verify

(BioAPI_HANDLE BSPHandle,

BioAPI_FMR MaxFMRRequested,

*const BioAPI_INPUT_BIR *ReferenceTemplate,*

BioAPI_BIR_SUBTYPE Subtype,

*BioAPI_BIR_HANDLE *AdaptedBIR,*

*BioAPI_BOOL *Result,*

*BioAPI_FMR *FMRAchieved,*

*BioAPI_DATA *Payload,*

int32_t Timeout,

*BioAPI_BIR_HANDLE *AuditData).*

Подпункт 8.4.8.1

Данная функция получает биометрические данные с присоединенного устройства (модуля датчика) и сравнивает их с контрольным шаблоном ReferenceTemplate.

Приложение должно запросить максимальное значение критерия ОЛС (пороговое значение) для успешного сопоставления. Булево значение Result указывает, успешно или нет была проведена верификация, а параметр FMRAchieved представляет собой значение ОЛС, определяющее степень близости сопоставляемых ЗБИ.

AuditData (выходной, необязательный параметр) – дескриптор ЗБИ, содержащей исходные биометрические данные. Эти данные могут быть использованы для предоставления биометрических данных личности. Входное значение указателя NULL указывает, что контрольные данные не собирают. Не все ПБУ поддерживают сбор контрольных данных. ПБУ может возвратить значение дескриптора BioAPI_UNSUPPORTED_BIR_HANDLE, чтобы указать, что AuditData не поддерживается, или значение BioAPI_INVALID_BIR_HANDLE, чтобы указать, что контрольные данные недоступны.

Подраздел 7.47

#define BioAPI_RAW (0x00000001)

Если данное значение установлено, то ПБУ поддерживает возврат исходных/контрольных данных.

Подпункт А.4.6.2.1

Возврат исходныхданных

Функции, обеспечивающие получение биометрических данных от датчика, могут дополнительно поддерживать возврат исходных данных для визуализации или контроля. Если данная функция поддерживает возврат исходных данных, то выходной параметр AuditData будет содержать указатель этих данных. Если возврат исходных данных не поддерживается, то ПБУ возвращает значение "–1".

Ссылки: 9.3.4.8, 8.4.8.1, 7.47 и А.4.6.2.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_Enroll* для создания шаблона.

4) Вызвать функцию *BioSPI_Verify* для верификации полученной ЗБИ на основе сохраненного шаблона.

5) Проверить возвращаемый код. Если контрольные данные не поддерживаются, то предполагается, что дескриптор ЗБИ контрольных данных содержит значение BioAPI_UNSUPPORTED_BIR_HANDLE. Если контрольные данные недоступны, то предполагается, что дескриптор ЗБИ контрольных данных содержит значение BioAPI_INVALID_BIR_HANDLE.

6) Отсоединить и выгрузить испытуемый ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_Verify* с предоставлением контрольных данных возвращает значение BioAPI_UNSUPPORTED_BIR_HANDLE (если контрольные данные не поддерживаются) или одно из значений BioAPI_OK или BioAPI_INVALID_BIR_HANDLE (если контрольные данные поддерживаются).

Пакет языка утверждения

```
<package name="89719700-f218-11d9-b028-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение
        "BioSPI_Verify_AuditData" (см. далее элемент
        <description> утверждения).
    </description>

    <assertion name="BioSPI_Verify_AuditData" model="BSPTesting">
        <description>
```

Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции BioSPI_Verify с предоставлением контрольных данных.

Следующий текст соответствует приведенному в 8.4.8.1, A.4.6.2.1 и 7.47 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_Verify
     (BioAPI_HANDLE BSPHandle,
     BioAPI_FMR MaxFMRRequested,
     const BioAPI_INPUT_BIR *ReferenceTemplate,
     BioAPI_BIR_SUBTYPE Subtype,
     BioAPI_BIR_HANDLE *AdaptedBIR,
     BioAPI_BOOL *Result,
     BioAPI_FMR *FMRAchieved,
     BioAPI_DATA *Payload,
     int32_t Timeout,
     BioAPI_BIR_HANDLE *AuditData).
```

Данная функция получает биометрические данные с присоединенного устройства (модуля датчика) и сравнивает их с контрольным шаблоном ReferenceTemplate. Приложение должно запросить максимальное значение критерия ОЛС (пороговое значение) для успешного сопоставления. Булево значение Result указывает, успешно или нет была проведена верификация, а параметр FMRAchieved представляет собой значение ОЛС, определяющее степень близости сопоставляемых ЗБИ.

Подпункт 8.4.8.1

AuditData (выходной, необязательный параметр) - дескриптор ЗБИ, содержащей исходные биометрические данные. Эти данные могут быть использованы для предоставления биометрических данных личности. Входное значение указателя NULL указывает, что контрольные

данные не собирают. Не все ПБУ поддерживают сбор контрольных данных. ПБУ может возвратить значение дескриптора BioAPI_UNSUPPORTED_BIR_HANDLE, чтобы указать, что AuditData не поддерживается, или значение BioAPI_INVALID_BIR_HANDLE, чтобы указать, что контрольные данные недоступны.

Подраздел 7.47: BioAPI_OPTIONS_MASK
#define BioAPI_RAW (0x00000001)
Если данное значение установлено, то ПБУ поддерживает возврат исходных/контрольных данных.

Подпункт A.4.6.2.1:
Возврат исходных данных.
Функции, обеспечивающие получение биометрических данных от датчика, могут дополнительно поддерживать возврат исходных данных для визуализации или контроля. Если данная функция поддерживает возврат исходных данных, то выходной параметр AuditData будет содержать указатель этих данных. Если возврат исходных данных не поддерживается, то ПБУ возвращает значение "-1".

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_Enroll для создания шаблона.
4) Вызвать функцию BioSPI_Verify для верификации полученной ЗБИ на основе сохраненного шаблона.
5) Проверить возвращаемый код. Если контрольные данные не поддерживаются, то предполагается, что дескриптор ЗБИ контрольных данных содержит значение BioAPI_UNSUPPORTED_BIR_HANDLE. Если контрольные данные недоступны, то предполагается, что дескриптор

```
        ЗБИ контрольных данных содержит значение
        BioAPI_INVALID_BIR_HANDLE.
        6) Отсоединить и выгрузить испытуемый ПБУ.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>

    <!-- Указание на отсутствие поддержки испытуемым ПБУ
    уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
    <input name="_noSourcePresentSupported" />

    <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
    PRESENT. -->
    <input name="_sourcepresenttimeout"/>

    <!-- Время ожидания для функции BioSPI_Enroll. -->
    <input name="_capturetimeout"/>

    <!-- Порог сопоставления для функции BioSPI_Verify. -->
    <input name="_maxFMRRequested"/>

    <!-- Время ожидания для функции BioSPI_Verify. -->
    <input name="_verifytimeout" />

    <!-- Указание на поддержку испытуемым ПБУ контрольных
    данных.-->
    <input name="_supportAuditData" />
```

```
        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_Verify">
            <input name="bspUuid" var="_bspUuid"/>
            <input name="inserttimeouttime"
            var="_inserttimeout"/>
            <input name="nosourcepresentsupported"
            var="_noSourcePresentSupported" />
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="capturetimeouttime"
            var="_capturetimeout"/>
            <input name="maxFMRRequested"
            var="_maxFMRRequested"/>
            <input name="verifytimeout" var="_verifytimeout" />
            <input name="supportAuditData"
            var="_supportAuditData" />
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_Verify">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeouttime"/>
```

```
<input name="maxFMRRequested" />
<input name="verifytimeout" />
<input name="supportAuditData" />

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<set name="eventtimeoutflag" value="false"/>

<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
          setvar="eventtimeoutflag">
     <or var1="nosourcepresentsupported"
     var2="_sourcePresent" />
</wait_until>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Испытуемый ПБУ не поддерживает уведомление о
          событии BioAPI_NOTIFY_SOURCE_PRESENT или
          уведомление о событии получено в пределах
          указанной максимальной продолжительности.
     </description>
     <not var="eventtimeoutflag"/>
</assert_condition>

<!-- Вызов функции BioSPI_Enroll для создания шаблона. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Subtype" value="0"/>
     <input name="Timeout" var="capturetimeouttime"/>
     <output name="NewTemplate"
     setvar="template_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided" >
     <description>
```

```
            Функция BioSPI_Enroll возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызов функции BioSPI_Verify для получения ЗБИ и ее
верификации с сохраненным шаблоном.-->
<invoke function="BioSPI_Verify" >
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="MaxFMRRequested" var="maxFMRRequested" />
        <input name="ReferenceTemplate_Form"
        var="__BioAPI_BIR_HANDLE_INPUT" />
        <input name="ReferenceTemplate_BIRHandle"
        var="template_handle" />
        <input name="Subtype" value="0"/>
        <input name="no_AdaptedBIR" value="true" />
        <input name="Timeout" var="verifytimeout" />
        <input name="no_AuditData" value="false" />
        <input name="no_Result" value="false" />
        <input name="no_Payload" value="true" />
        <output name="Result" setvar="result" />
        <output name="AuditData" setvar="auditbir_handle" />
        <output name="FMRAchieved" setvar="fmrAchieved" />
        <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
            break_if_false="true" >
        <description>
```

```
            Функция BioSPI_Verify возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition break_if_false="true">
        <description>
            Выходной дескриптор ЗБИ контрольных данных имеет
            значение BioAPI_UNSUPPORTED_BIR_HANDLE (если
            контрольные данные не поддерживаются) или одно
            из значений BioAPI_OK или BioAPI_INVALID_BIR_
            HANDLE (если контрольные данные поддерживаются).
        </description>
        <or>
            <and>
                <same_as var1="supportAuditData"
                value2="true" />
                <or>
                    <equal_to var1="auditbir_handle"
                        var2="__BioAPI_INVALID_BIR_HANDLE" />
                    <greater_than_or_equal_to
                    var1="auditbir_handle" value2="0" />
                </or>
            </and>
            <and>
                <same_as var1="supportAuditData"
                value2="false" />
                <equal_to var1="auditbir_handle"
                    var2="__BioAPI_UNSUPPORTED_BIR_HANDLE" />
```

```
                </and>
            </or>
        </assert_condition>

        <!-- Проверка степени обработки контрольных данных ЗБИ. -->
        <invoke activity="check_audit_data_type" >
            <only_if>
                <same_as var1="supportAuditData" value2="true" />
                <greater_than_or_equal_to
                var1="auditbir_handle" value2="0" />
            </only_if>
            <input name="BSPHandle" var="_bsphandle"/>
            <input name="Handle" var="auditbir_handle"/>
        </invoke>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>

    <activity name="check_audit_data_type" >
        <input name="BSPHandle" />
        <input name="Handle" />

        <!-- Вызов функции BioSPI_GetHeaderFromHandle. -->
        <invoke function="BioSPI_GetHeaderFromHandle">
            <input name="BSPHandle" var="_bsphandle"/>
            <input name="Handle" var="Handle"/>
            <output name="ProcessedLevel"
            setvar="processedLevel"/>
            <return setvar="return"/>
        </invoke>
```

```
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
                break_if_false="true">
            <description>
                Функция BioSPI_GetHeaderFromHandle возвращает
                значение BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Степень обработки контрольных данных ЗБИ
                соответствует исходному биометрическому образцу.
            </description>
            <equal_to var1="processedLevel"
                var2="__BioAPI_BIR_DATA_TYPE_RAW"/>
        </assert_condition>
    </activity>
</package>
```

8.53 Утверждение 15a *BioSPI_DbOpen_ValidParam*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbOpen* с достоверными значениями входных параметров и проверки, является ли возвращенный код значением BioAPI_OK.

Выдержки

Подпункт 9.3.5.1

BioAPI_RETURN BioAPI BioSPI_DbOpen

(BioAPI_HANDLE BSPHandle,

*const BioAPI_UUID *DbUuid,*

BioAPI_DB_ACCESS_TYPE AccessRequest,

*BioAPI_DB_HANDLE *DbHandle,*

*BioAPI_DB_MARKER_HANDLE *MarkerHandle).*

Подпункт 8.5.1.1

Данная функция открывает базу данных ЗБИ, поддерживаемую текущим присоединенным архивом установленного вызова ПБУ, используя режим доступа, указанный в AccessRequest.

Ссылки: 9.3.5.1 и 8.5.1.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbOpen* для открытия определенной базы данных.

4) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

5) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbOpen* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="e68ff9a0-e506-11d9-a6a1-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
```

```
    Данный пакет содержит утверждение "BioSPI_DbOpen_ValidParam".
</description>

<assertion name="BioSPI_DbOpen_ValidParam" model="BSPTesting">
    <description>
        Проверку данного утверждения проводят с целью вызова
        функции BioSPI_DbOpen с достоверными значениями
        входных параметров и проверки, является ли
        возвращенный код значением BioAPI_OK.
        Следующий текст соответствует приведенному в 8.5.1.1
        спецификации БиоАПИ 2.0.
        ------------------------------------------------
        BioAPI_RETURN BioAPI BioSPI_DbOpen
            (BioAPI_HANDLE BSPHandle,
            BioAPI_DB_ACCESS_TYPE AccessRequest,
            BioAPI_DB_HANDLE *DbHandle,
            BioAPI_DB_MARKER_HANDLE *MarkerHandle).

            Данная функция открывает базу данных ЗБИ,
            поддерживаемую текущим присоединенным архивом
            установленного вызова ПБУ, используя режим
            доступа, указанный в AccessRequest.
        ------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Вызвать функцию BioSPI_DbOpen для открытия опре-
        деленной базы данных.
        4) Проверить возвращаемый код. Предполагается воз-
        вращение значения BioAPI_OK.
        5) Отсоединить и выгрузить ПБУ.
```

```
        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>

    <!-- Указание на отсутствие поддержки испытуемым ПБУ
    уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
    <input name="_noSourcePresentSupported" />

    <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
    PRESENT.-->
    <input name="_sourcepresenttimeout"/>

    <!-- УУИД базы данных, которую необходимо открыть. -->
    <input name="_dbUuid"/>

    <!-- Чтение запроса на доступ к базе данных.-->
    <input name="_readAccessRequest"/>

    <!-- Запись запроса на доступ.-->
    <input name="_writeAccessRequest"/>

    <!--Вызов основного процесса данного утверждения со зна-
    чениями входных параметров, определенными значениями па-
    раметров утверждения. -->
    <invoke activity="BioSPI_DbOpen">
        <input name="bspUuid" var="_bspUuid"/>
        <input name="inserttimeouttime"
        var="_inserttimeout"/>
```

```
            <input name="nosourcepresentsupported"
            var="_noSourcePresentSupported" />
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="dbUuid" var="_dbUuid"/>
            <input name="readAccessRequest"
            var="_readAccessRequest"/>
            <input name="writeAccessRequest"
            var="_writeAccessRequest" />
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_DbOpen">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="dbUuid"/>
        <input name="readAccessRequest"/>
        <input name="writeAccessRequest"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
```

```
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<invoke activity="PrepareDBTesting"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspHandle" var="_bsphandle" />
     <input name="dbUuid" var="dbUuid" />
     <input name="nosourcepresentsupported"
     var="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"
     var="sourcepresenttimeouttime"/>
</invoke>

<!-- Вызов функции BioSPI_DbOpen для открытия определенной
базы данных. -->
<invoke function="BioSPI_DbOpen">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="ReadAccessRequest"
     var="readAccessRequest"/>
     <input name="WriteAccessRequest"
     var="writeAccessRequest"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
```

```
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
    <description>
        Функция BioSPI_DbOpen возвращает значение
        BioAPI_OK и выходной маркер dbHandle является
        достоверным маркером базы данных.
    </description>
    <equal_to var1="return" var2="__BioAPI_OK"/>
    <not_equal_to var1="dbHandle"
    var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>

<!-- Вызов функции BioSPI_DbClose с достоверными значе-
ниями входных параметров. -->
<invoke function="BioSPI_DbClose">
    <input name="BSPHandle" var="_bsphandle"/>
    <input name="DbHandle" var="dbHandle"/>
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
        break_if_false="true">
    <description>
        Функция BioSPI_DbClose возвращает значение
        BioAPI_OK.
    </description>
```

```
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <invoke activity="CleanUpDBTesting"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                break_on_break="true">
            <input name="BSPHandle" var="_bsphandle" />
            <input name="dbUuid" var="dbUuid" />
        </invoke>

        <!-- Вызов функций BioSPI_ModuleDetach и
        BioSPI_ModuleUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.54 Утверждение 15b *BioSPI_DbOpen_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbOpen* с недостоверным дескриптором ПБУ и проверки, является ли возвращенный код значением BioAPIERR_INVALID_BSP_HANDLE.

Выдержки

Подпункт 9.3.5.1

BioAPI_RETURN BioAPI BioSPI_DbOpen
(BioAPI_HANDLE BSPHandle,
*const BioAPI_UUID *DbUuid,*
BioAPI_DB_ACCESS_TYPE AccessRequest,
*BioAPI_DB_HANDLE *DbHandle,*
*BioAPI_DB_MARKER_HANDLE *MarkerHandle).*

Подпункт 8.5.1.1

Данная функция открывает базу данных ЗБИ, поддерживаемую текущим присоединенным архивом установленного вызова ПБУ, используя режим доступа, указанный в AccessRequest.

Ссылки: 9.3.5.1 и 8.5.1.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbOpen* с недостоверным дескриптором ПБУ.

4) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_INVALID_BSP_HANDLE.

5) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbOpen* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="bfd44400-e5de-11d9-bdb9-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение "BioSPI_DbOpen_
        InvalidBSPHandle".
    </description>

    <assertion name="BioSPI_DbOpen_InvalidBSPHandle"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью вызова
            функции BioSPI_DbOpen с недостоверным дескриптором
            ПБУ и проверки, является ли возвращенный код значением
            BioAPIERR_INVALID_BSP_HANDLE.
```

```
        Приведенный ниже текст соответствует приведенному в
        8.5.1.1 спецификации БиоАПИ 2.0.
        ------------------------------------------------
        BioAPI_RETURN BioAPI BioSPI_DbOpen
             (BioAPI_HANDLE BSPHandle,
             const BioAPI_UUID *DbUuid,
             BioAPI_DB_ACCESS_TYPE AccessRequest,
             BioAPI_DB_HANDLE *DbHandle,
             BioAPI_DB_MARKER_HANDLE *MarkerHandle).

        Данная функция открывает базу данных ЗБИ, поддерживаемую
        текущим присоединенным архивом установленного вызова
        ПБУ, используя режим доступа, указанный в AccessRequest.
        ------------------------------------------------
        Подпункт 8.5.1.2:
        BSPHandle (входной параметр) - дескриптор
        присоединенного вызова ПБУ.
        ------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Вызвать функцию BioSPI_DbOpen с недостоверным
        дескриптором ПБУ.
        4) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPIERR_INVALID_BSP_HANDLE.
        5) Отсоединить и выгрузить ПБУ.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
```

```
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />
<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Имя базы данных, которую необходимо открыть. -->
<input name="_dbUuid"/>

<!-- Чтение запроса на доступ к базе данных.-->
<input name="_readAccessRequest"/>

<!-- Запись запроса на доступ.-->
<input name="_writeAccessRequest"/>

<!--Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbOpen">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="dbUuid" var="_dbUuid"/>
     <input name="readAccessRequest"
     var="_readAccessRequest"/>
     <input name="writeAccessRequest"
```

```
            var="_writeAccessRequest" />
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            function="BioSPI_EventHandler"/>
    </assertion>

    <activity name="BioSPI_DbOpen">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="dbUuid"/>
        <input name="readAccessRequest"/>
        <input name="writeAccessRequest"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
            <input name="bspVersion" value="32"/>
            <input name="unitIDOrNull" value="0"/>
```

```
    <input name="bspHandle" var="_bsphandle"/>
    <input name="eventtimeouttime"
    var="inserttimeouttime"/>
</invoke>

<invoke activity="PrepareDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
    <input name="bspHandle" var="_bsphandle" />
    <input name="dbUuid" var="dbUuid" />
    <input name="nosourcepresentsupported"
    var="nosourcepresentsupported"/>
    <input name="sourcepresenttimeouttime"
    var="sourcepresenttimeouttime"/>
</invoke>

<!-- Вызов функции BioSPI_DbOpen с недостоверным
дескриптором модуля. -->
<invoke function="BioSPI_DbOpen">
    <input name="BSPHandle" value="0"/>
    <input name="DbUuid" var="dbUuid"/>
    <input name="ReadAccessRequest"
    var="readAccessRequest"/>
    <input name="WriteAccessRequest"
    var="writeAccessRequest"/>
    <output name="DbHandle" setvar="dbHandle"/>
    <output name="MarkerHandle" setvar="markerHandle"/>
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition>
    <description>
```

```
                    Функция BioSPI_DbOpen возвращает значение
                    BioAPIERR_INVALID_BSP_HANDLE.
               </description>
               <equal_to var1="return"
               var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
               <equal_to var1="dbHandle"
               var2="__BioAPI_DB_INVALID_HANDLE" />
          </assert_condition>

          <invoke activity="CleanUpDBTesting"
                    package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                    break_on_break="true">
               <input name="BSPHandle" var="_bsphandle" />
               <input name="dbUuid" var="dbUuid" />
          </invoke>

          <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
          <invoke activity="DetachAndUnload"
                    package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
               <input name="bspUuid" var="bspUuid" />
               <input name="BSPHandle" var="_bsphandle" />
          </invoke>
     </activity>
</package>
```

8.55 Утверждение 16a *BioSPI_DbClose_ValidParam*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbClose* с достоверными значениями входных параметров и проверки, является ли возвращенный код значением BioAPI_OK.

Выдержки

Подпункт 9.3.5.2

BioAPI_RETURN BioAPI BioSPI_DbClose

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_HANDLE DbHandle).

Подпункт 8.5.2.1

Данная функция закрывает открытую базу данных ЗБИ.

Ссылки: 9.3.5.2 и 8.5.2.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Создать временную базу данных.

4) Вызвать функцию *BioSPI_DbOpen* для открытия временной базы данных.

5) Вызвать функцию *BioSPI_DbClose* с достоверными значениями входных параметров. Проверить возвращаемый код.

6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbClose* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="39aa9560-e5f1-11d9-89f3-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение
          "BioSPI_DbClose_ValidParam".
     </description>

     <assertion name="BioSPI_DbClose_ValidParam"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью вызова
               функции BioSPI_DbClose с достоверными значениями вход-
```

```
ных параметров и проверки, является ли возвращенный
код значением BioAPI_OK.
Приведенный ниже текст соответствует приведенному в
8.5.2.1 спецификации БиоАПИ 2.0.
-------------------------------------------------
BioAPI_RETURN BioAPI BioAPI_DbClose
     (BioAPI_HANDLE BSPHandle,
     BioAPI_DB_HANDLEDbHandle).

Данная функция закрывает открытую базу данных ЗБИ.
-------------------------------------------------
Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Создать временную базу данных.
4) Вызвать функцию BioSPI_DbOpen для открытия
временной базы данных.
5) Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. Проверить возвращаемый
код.
6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли
неуспешно, то выдается заключение о соответствии
"UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>
```

```
        <!-- Указание на отсутствие поддержки испытуемым ПБУ
        уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
        <input name="_noSourcePresentSupported" />

        <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
        PRESENT. -->
        <input name="_sourcepresenttimeout"/>

        <!-- Имя базы данных, которая должна быть создана. -->
        <input name="_dbUuid"/>

        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_DbClose">
             <input name="bspUuid" var="_bspUuid"/>
             <input name="inserttimeouttime"
             var="_inserttimeout"/>
             <input name="nosourcepresentsupported"
             var="_noSourcePresentSupported" />
             <input name="sourcepresenttimeouttime"
             var="_sourcepresenttimeout"/>
             <input name="dbUuid" var="_dbUuid"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_DbClose">
        <input name="bspUuid"/>
```

```
<input name="inserttimeouttime"/>
<input name="nosourcepresentsupported" />
<input name="sourcepresenttimeouttime"/>
<input name="dbUuid"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<invoke activity="PrepareDBTesting"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspHandle" var="_bsphandle" />
     <input name="dbUuid" var="dbUuid" />
     <input name="nosourcepresentsupported"
     var="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"
     var="sourcepresenttimeouttime"/>
</invoke>
```

```
<!-- Вызов функции BioSPI_DbOpen для открытия заданной БД. -->
<invoke function="BioSPI_DbOpen">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_DbOpen возвращает значение
          BioAPI_OK, и выходной дескриптор dbHandle
          является достоверным дескриптором базы данных.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
     <not_equal_to var1="dbHandle"
     var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>

<!-- Вызов функции BioSPI_DbClose с достоверными значениями
входных параметров. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <return setvar="return"/>
</invoke>
```

```
        <!-- Выдача заключения о соответствии. Если условие, оп-
        ределенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL", в противном
        случае выдается заключение о соответствии "PASS". -->
        <assert_condition>
            <description>
                Функция BioSPI_DbClose возвращает значение
                BioAPI_OK.
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>

        <invoke activity="CleanUpDBTesting"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                break_on_break="true">
            <input name="BSPHandle" var="_bsphandle" />
            <input name="dbUuid" var="dbUuid" />
        </invoke>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.56 Утверждение 16b *BioSPI_DbClose_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbClose* с недостоверным дескриптором ПБУ и проверки, является ли возвращенный код значением BioAPIERR_INVALID_BSP_HANDLE.

Выдержки

Подпункт 9.3.5.2

BioAPI_RETURN BioAPI BioSPI_DbClose

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_HANDLE DbHandle).

Подпункт 8.5.2.1

Данная функция закрывает открытую базу данных ЗБИ.

Ссылки: 9.3.5.2 и 8.5.2.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Создать временную базу данных.

4) Вызвать функцию *BioSPI_DbOpen* для открытия временной базы данных.

5) Вызвать функцию *BioSPI_DbClose* с недостоверным дескриптором ПБУ. Проверить возвращаемый код.

6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbClose* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="6e3f5c00-e5f3-11d9-b663-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_DbClose_
          InvalidBSPHandle".
     </description>

     <assertion name="BioSPI_DbClose_InvalidBSPHandle"
     model="BSPTesting">
          <description>
```

```
Проверку данного утверждения проводят с целью вызова
функции BioSPI_DbClose с недостоверным дескриптором
ПБУ и проверки, является ли возвращенный код значением
BioAPIERR_INVALID_BSP_HANDLE.
Приведенный ниже текст соответствует приведенному в
8.5.2.1 спецификации БиоАПИ 2.0.
-------------------------------------------------
BioAPI_RETURN BioAPI BioAPI_DbClose
     (BioAPI_HANDLE BSPHandle,
     BioAPI_DB_HANDLEDbHandle).

Данная функция закрывает открытую базу данных ЗБИ.
-------------------------------------------------
8.5.2.2 Параметры
BSPHandle (входной параметр) - дескриптор
присоединенного ПБУ.
-------------------------------------------------
Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Создать временную базу данных.
4) Вызвать функцию BioSPI_DbOpen для открытия
временной базы данных.
5) Вызвать функцию BioSPI_DbClose с недостоверным
дескриптором ПБУ. Проверить возвращаемый код.
6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли
неуспешно, то выдается заключение о соответствии
"UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>
```

```
<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT.-->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Имя базы данных, которая должна быть создана. -->
<input name="_dbUuid"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbClose">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="dbUuid" var="_dbUuid"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически. -->
<bind activity="EventHandler"
     package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
     function="BioSPI_EventHandler"/>
```

```
</assertion>

<activity name="BioSPI_DbClose">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported" />
     <input name="sourcepresenttimeouttime"/>
     <input name="dbUuid"/>

     <!-- Данное утверждение использует BSPHandle со значением
     "1" для всех вызовов BioSPI, которые этого требуют. -->
     <set name="_bsphandle" value="1"/>

     <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
     предоставленных испытуемым ПБУ. Входное значение параметра
     "unitIDOrNull" равно "0", поэтому утверждение будет
     тестировать модуль датчика, выбранный ПБУ. -->
     <invoke activity="LoadAndAttach"
               package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
               break_on_break="true">
          <input name="bspUuid" var="bspUuid"/>
          <input name="bspVersion" value="32"/>
          <input name="unitIDOrNull" value="0"/>
          <input name="bspHandle" var="_bsphandle"/>
          <input name="eventtimeouttime"
          var="inserttimeouttime"/>
     </invoke>

     <invoke activity="PrepareDBTesting"
               package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
               break_on_break="true">
          <input name="bspHandle" var="_bsphandle" />
          <input name="dbUuid" var="dbUuid" />
          <input name="nosourcepresentsupported"
          var="nosourcepresentsupported"/>
```

```
     <input name="sourcepresenttimeouttime"
     var="sourcepresenttimeouttime"/>
</invoke>

<!-- Вызов функции BioSPI_DbOpen для открытия указанной
базы данных. -->
<invoke function="BioSPI_DbOpen">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED", в противном
случае выдается заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
          break_if_false="true">
     <description>
          Функция BioSPI_DbOpen возвращает значение
          BioAPI_OK и выходной дескриптор dbHandle
          является достоверным дескриптором базы данных.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
     <not_equal_to var1="dbHandle"
     var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>

<!-- Вызов функции BioSPI_DbClose с недостоверным
дескриптором ПБУ. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" value="0"/>
```

```
        <input name="DbHandle" var="dbHandle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "FAIL", в противном
    случае выдается заключение о соответствии "PASS". -->
    <assert_condition>
        <description>
            Функция BioSPI_DbClose возвращает значение
            BioAPIERR_H_FRAMEWORK_INVALID_BSP_HANDLE.
        </description>
        <equal_to var1="return"
        var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
    </assert_condition>
    <!-- Вызов функции BioSPI_DbClose с достоверными значениями
    входных параметров.-->
    <invoke function="BioSPI_DbClose">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="DbHandle" var="dbHandle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие, оп-
    ределенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "FAIL", в противном
    случае выдается заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
            break_if_false="true">
        <description>
            Функция BioSPI_DbClose возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
```

```
        </assert_condition>

        <invoke activity="CleanUpDBTesting"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
                break_on_break="true">
            <input name="BSPHandle" var="_bsphandle" />
            <input name="dbUuid" var="dbUuid" />
        </invoke>

        <!-- Вызов функций BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
                package="02c59458-0c46-1085-95d7-0002a5d5fd2e">
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.57 Утверждение 17а *BioSPI_DbCreate_DbProtected*

Описание: Проверку данного утверждения проводят с целью двойного вызова функции *BioSPI_DbCreate* и проверки возвращения сообщения об ошибке BioAPIERR_DATABASE_ALREADY_EXISTS при повторномвызове.

Выдержки

Подпункт 9.3.5.2

BioAPI_RETURN BioAPI BioSPI_DbClose

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_HANDLE DbHandle).

Подпункт 8.5.3.1

Данная функция создает и открывает новую базу данных ЗБИ для присоединенного архива, идентифицированного вызовом ПБУ. Идентификатор новой базы данных определен входным параметром DbUuid, который должен быть создан биометрическим приложением и должен отличаться от всех текущих УУИД баз данных, поддерживаемых данным архивом, открытым или нет.

Вновь созданная база данных ЗБИ открывается в установленном режиме доступа.

Ссылки: 9.3.5.3 и 8.5.3.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвавать функцию *BioSPI_DbCreate* для создания базы данных.

4) Повторно вызвать функцию *BioSPI_DbCreate*.

5) Проверить возвращаемое значение вызова функции. Предполагается завершение функции с ошибкой BioAPIERR_INVALID_BSP_HANDLE.

6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Повторный вызов функции *BioSPI_DbCreate* возвращает значение BioAPIERR_DATABASE_ALREADY_EXISTS.

Пакет языка утверждения

```
<package name="7b6c2f40-e650-11d9-812f-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_DbCreate_
     DbProtected".
     </description>

     <assertion name="BioSPI_DbCreate_DbProtected"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью двойного
               вызова функции BioSPI_DbCreate и проверки возвращения
```

сообщения об ошибке BioAPIERR_DATABASE_ALREADY_EXISTS при повторном вызове.

Следующий текст соответствует приведенному в 8.5.3.1 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_DbCreate
     (BioAPI_HANDLE BSPHandle,
     const BioAPI_UUID *DbUuid,
     uint32_t NumberOfRecords,
     BioAPI_DB_ACCESS_TYPE AccessRequest,
     BioAPI_DB_HANDLE *DbHandle).
```

Данная функция создает и открывает новую базу данных ЗБИ для присоединенного архива, идентифицированного вызовом ПБУ. Идентификатор новой базы данных определен входным параметром DbUuid, который должен быть создан биометрическим приложением и должен отличаться от всех текущих УУИД баз данных, поддерживаемых данным архивом, открытым или нет. Вновь созданная база данных ЗБИ открывается в установленном режиме доступа.

Ошибки:BioAPIERR_DATABASE_ALREADY_EXISTS.

DbHandle (выходной) - дескриптор созданного и открытого хранилища данных. Значение будет установлено в BioAPI_DB_INVALID_HANDLE, если функция выйдет из строя.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвавать функцию BioSPI_DbCreate для создания базы данных.
4) Повторно вызвать функцию BioSPI_DbCreate.

```
        5) Проверить возвращаемое значение вызова функции.
        Предполагается завершение функции с ошибкой
        BioAPIERR_DATABASE_ALREADY_EXISTS.
        6) Отсоединить и выгрузить ПБУ.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- База данных Uuid будет открыта. -->
<input name="_dbUuid"/>

<!-- Запрос доступа на чтение базы данных. -->
<input name="_readAccessRequest"/>

<!-- Запрос доступа на запись. -->
<input name="_writeAccessRequest"/>
<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbCreate">
        <input name="bspUuid" var="_bspUuid"/>
        <input name="inserttimeouttime"
        var="_inserttimeout"/>
        <input name="dbUuid" var="_dbUuid"/>
        <input name="readAccessRequest"
        var="_readAccessRequest"/>
        <input name="writeAccessRequest"
        var="_writeAccessRequest" />
```

```
    </invoke>

    <!-- Процесс связан с функцией интерфейса обратного вызова
    инфраструктуры, предоставленной испытуемым компонентом
    ПБУ. При каждом входящем вызове функции связанный с ней
    процесс будет вызываться автоматически. -->
    <bind activity="EventHandler"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        function="BioSPI_EventHandler"/>
    </assertion>
<activity name="BioSPI_DbCreate">
    <input name="bspUuid"/>
    <input name="inserttimeouttime"/>
    <input name="dbUuid"/>
    <input name="readAccessRequest"/>
    <input name="writeAccessRequest"/>
    <!-- Данное утверждение использует BSPHandle со значением
    "1" для всех вызовов BioSPI, которые этого требуют. -->

    <set name="_bsphandle" value="1"/>

    <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
    предоставленных испытуемым ПБУ. Входное значение параметра
    "unitIDOrNull" равно "0", поэтому утверждение будет
    тестировать модуль датчика, выбранный ПБУ. -->
    <invoke activity="LoadAndAttach"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
    </invoke>
```

```
<!-- Удалить базу данных, если она существует. -->
<invoke function="BioSPI_DbDelete" >
     <input name="BSPHandle" var="_bsphandle" />
     <input name="DbUuid" var="dbUuid" />
     <return setvar="return"/>
</invoke>
<!-- Вызвать функцию BioSPI_DbCreate для создания
необходимой базы данных. -->
<invoke function="BioSPI_DbCreate">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="NumberOfRecords" value="1"/>
     <input name="ReadAccessRequest"
     var="readAccessRequest"/>
     <input name="WriteAccessRequest"
     var="writeAccessRequest"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
     <description>
          Функция BioSPI_DbCreate возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
```

```
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
     <description>
          Функция BioSPI_DbClose возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbCreate для повторного
создания необходимой базы данных. -->
<invoke function="BioSPI_DbCreate">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="NumberOfRecords" value="1"/>
     <input name="ReadAccessRequest"
     var="readAccessRequest"/>
     <input name="WriteAccessRequest"
     var="writeAccessRequest"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
```

```
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
     <description>
          Функция BioSPI_DbCreate возвращает значение
          BioAPIERR_DATABASE_ALREADY_EXISTS и выходной
          дескриптор DbHandle установлен в положение
          BioAPI_DB_INVALID_HANDLE.
     </description>
     <equal_to var1="return"
     var2="__BioAPIERR_BSP_DATABASE_ALREADY_EXISTS"/>
     <equal_to var1="dbHandle"
     var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbDelete с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbDelete">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
     <description>
          Функция BioSPI_DbDelete возвращает значение
          BioAPI_OK.
     </description>
     <equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
```

```
        <!-- Вызвать функцию BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
             <input name="bspUuid" var="bspUuid" />
             <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.58 Утверждение 17b *BioSPI_DbCreate_ValidParam*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_DbCreate* с достоверными значениями входных параметров.

Выдержки

Подпункт 9.3.5.3

BioAPI_RETURN BioAPI BioSPI_DbCreate

(BioAPI_HANDLE BSPHandle,

*const BioAPI_UUID *DbUuid,*

uint32_t NumberOfRecords,

BioAPI_DB_ACCESS_TYPE AccessRequest,

*BioAPI_DB_HANDLE *DbHandle).*

Подпункт 8.5.3.1

Данная функция создает и открывает новую базу данных ЗБИ для присоединенного архива, идентифицированного вызовом ПБУ. Идентификатор новой базы данных определен входным параметром DbUuid, который должен быть создан биометрическим приложением и должен отличаться от всех текущих УУИД баз данных, поддерживаемых данным архивом, открытым или нет. Вновь созданная база данных ЗБИ открывается в установленном режиме доступа.

Ссылки: 9.3.5.3 и 8.5.3.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных.

4) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

5) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbCreate* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="1421ec38-1db6-49d4-873d-03e2de17598b">
     <author>
          ISO/IEC JTC1 SC37
     </author>
     <description>
          Данный пакет содержит утверждение "BioSPI_DbCreate_ Va-
          lidParam".
     </description>

     <assertion name="BioSPI_DbCreate_ValidParam"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               определения возвращения значения BioAPI_OK при вызове
               функции BioSPI_DbCreate с достоверными значениями
               входных параметров.
               Следующий текст соответствует приведенному в 8.5.3.1
               спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioSPI_DbCreate
                    (BioAPI_HANDLE BSPHandle,
                    const BioAPI_UUID *DbUuid,
```

```
            uint32_t NumberOfRecords,
            BioAPI_DB_ACCESS_TYPE AccessRequest,
            BioAPI_DB_HANDLE *DbHandle).
        Данная функция создает и открывает новую базу данных
        ЗБИ для присоединенного архива, идентифицированного
        вызовом ПБУ. Идентификатор новой базы данных опреде-
        лен входным параметром DbUuid, который должен быть
        создан биометрическим приложением и должен отличать-
        ся от всех текущих УУИД баз данных, поддерживаемых
        данным архивом, открытым или нет. Вновь созданная
        база данных ЗБИ открывается в установленном режиме
        доступа.
        ----------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Вызвать функцию BioSPI_DbCreate для создания базы
        данных.
        4) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPI_OK.
        5) Отсоединить и выгрузить ПБУ.

        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- База данных Name будет открыта. -->
<input name="_dbUuid"/>
```

```
        <!-- Число записей. -->
        <input name="_nbrRecords"/>

        <!-- Запрос доступа на чтение базы данных. -->
        <input name="_readAccessRequest"/>

        <!-- Запрос доступа на запись. -->
        <input name="_writeAccessRequest"/>
        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_DbCreate">
             <input name="bspUuid" var="_bspUuid"/>
             <input name="inserttimeouttime"
             var="_inserttimeout"/>
             <input name="dbUuid" var="_dbUuid"/>
             <input name="nbrRecords" var="_nbrRecords"/>
             <input name="readAccessRequest"
             var="_readAccessRequest"/>
             <input name="writeAccessRequest"
             var="_writeAccessRequest" />
        </invoke>
        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
             package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
             function="BioSPI_EventHandler"/>
    </assertion>
    <activity name="BioSPI_DbCreate">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="dbSupported"/>
        <input name="dbUuid"/>
```

```
<input name="nbrRecords"/>
<input name="readAccessRequest"/>
<input name="writeAccessRequest"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
          package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
          break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<!-- Удалить базу данных,если она существует. -->
<invoke function="BioSPI_DbDelete" >
     <input name="BSPHandle" var="_bsphandle" />
     <input name="DbUuid" var="dbUuid" />
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается заклю-
чение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
```

```
break_if_false="true">
<description>
     Функция BioSPI_DbCreate возвращает значение
     BioAPI_OK.
</description>
<or>
<equal_to var1="return" var2="__BioAPI_OK"/>
<equal_to var1="return"
var2="__BioAPIERR_BSP_DATABASE_DOES_NOT_EXIST"/>
</or>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbCreate для создания базы
данных. -->
<invoke function="BioSPI_DbCreate">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="NumberOfRecords" var="nbrRecords"/>
     <input name="ReadAccessRequest"
     var="readAccessRequest"/>
     <input name="WriteAccessRequest"
     var="writeAccessRequest"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
<description>
     Функция BioSPI_DbCreate возвращает значение
     BioAPI_OK.
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
```

```
<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
<description>
     Функция BioSPI_DbClose возвращает значение BioAPI_OK
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbDelete с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbDelete">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
<description>
     Функция BioSPI_DbDelete возвращает значение
     BioAPI_OK.
```

```
            </description>
            <equal_to var1="return" var2="__BioAPI_OK"/>
            </assert_condition>
            <!-- Вызвать функции BioSPI_ModuleDetach и
            BioSPI_ModuleUnload. -->
            <invoke activity="DetachAndUnload"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
                 <input name="bspUuid" var="bspUuid" />
                 <input name="BSPHandle" var="_bsphandle" />
            </invoke>
      </activity>
</package>
```

8.59 Утверждение 17c *BioSPI_DbCreate_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbCreate* с недостоверным дескриптором ПБУ и проверки, является ли возвращенный код значением BioAPIERR_INVALID_BSP_HANDLE.

Выдержки

Подпункт 9.3.5.3

BioAPI_RETURN BioAPI BioSPI_DbCreate

(BioAPI_HANDLE BSPHandle,

*const BioAPI_UUID *DbUuid,*

uint32_t NumberOfRecords,

BioAPI_DB_ACCESS_TYPE AccessRequest,

*BioAPI_DB_HANDLE *DbHandle).*

Подпункт 8.5.3.1

Данная функция создает и открывает новую базу данных ЗБИ для присоединенного архива, идентифицированного вызовом ПБУ. Идентификатор новой базы данных определен входным параметром DbUuid, который должен быть создан биометрическим приложением и должен отличаться от всех текущих УУИД баз данных, поддерживаемых данным архивом, открытым или нет.

Вновь созданная база данных ЗБИ открывается в установленном режиме доступа.

Ссылки: 9.3.5.3 и 8.5.3.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных с недостоверным дескриптором ПБУ.

4) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_INVALID_BSP_HANDLE.

5) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbCreate* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="ef4bb862-79f6-4f01-8f5d-af5c3abf23c0">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_DbCreate_
          InvalidBSPHandle".
     </description>
     <assertion name="BioSPI_DbCreate_InvalidBSPHandle"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью вызова
               функции BioSPI_DbCreate с недостоверным дескриптором
               ПБУ и проверки, является ли возвращенный код значе-
               нием BioAPIERR_INVALID_BSP_HANDLE.
```

Приведенный ниже текст соответствует приведенному в 8.5.3.1 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_DbCreate
     (BioAPI_HANDLE BSPHandle,
     const BioAPI_UUID *DbUuid,
     uint32_t NumberOfRecords,
     BioAPI_DB_ACCESS_TYPE AccessRequest,
     BioAPI_DB_HANDLE *DbHandle).
```

Данная функция создает и открывает новую базу данных ЗБИ для присоединенного архива идентифицированного вызовом ПБУ. Идентификатор новой базы данных определен входным параметром DbUuid, который должен быть создан биометрическим приложением и должен отличаться от всех текущих УУИД баз данных, поддерживаемых данным архивом, открытым или нет. Вновь созданная база данных ЗБИ открывается в установленном режиме доступа.

8.5.3.2 Параметры

BSPHandle (входной параметр) - дескриптор присоединенного поставщика услуг БиоАПИ.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию BioSPI_DbCreate для создания определенной базы данных с недостоверным дескриптором ПБУ.

4) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_INVALID_BSP_HANDLE.

5) Отсоединить и выгрузить ПБУ.

```
      Если какие-либо промежуточные операции прошли
      неуспешно, то выдается заключение о соответствии
      "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- База данных Uuid будет открыта. -->
<input name="_dbUuid"/>

<!-- Запрос доступа на чтение базы данных. -->
<input name="_readAccessRequest"/>

<!-- Запрос доступа на запись. -->
<input name="_writeAccessRequest"/>
<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbCreate">
      <input name="bspUuid" var="_bspUuid"/>
      <input name="inserttimeouttime"
      var="_inserttimeout"/>
      <input name="dbUuid" var="_dbUuid"/>
      <input name="readAccessRequest"
      var="_readAccessRequest"/>
      <input name="writeAccessRequest"
      var="_writeAccessRequest" />
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом ПБУ.
```

```
При каждом входящем вызове функции связанный с ней процесс
будет вызываться автоматически.-->
<bindactivity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>
<activity name="BioSPI_DbCreate">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported" />
     <input name="sourcepresenttimeouttime"/>
     <input name="dbUuid"/>
     <input name="readAccessRequest"/>
     <input name="writeAccessRequest"/>
<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>
<!-- Удалить базу данных, если она существует. -->
<invoke function="BioSPI_DbDelete" >
     <input name="BSPHandle" var="_bsphandle" />
```

```
    <input name="DbUuid" var="dbUuid" />
    <return setvar="return"/>
</invoke>
<!-- Вызвать BioSPI_DbCreate для создания базы данных. -->
<invoke function="BioSPI_DbCreate">
    <input name="BSPHandle" value="0"/>
    <input name="DbUuid" var="dbUuid"/>
    <input name="NumberOfRecords" value="1"/>
    <input name="ReadAccessRequest"
    var="readAccessRequest"/>
    <input name="WriteAccessRequest"
    var="writeAccessRequest"/>
    <output name="DbHandle" setvar="dbHandle"/>
    <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
<description>
    Функция BioSPI_DbCreate возвращает значение BioAPIERR_
    INVALID_BSP_HANDLE и выходной дескриптор DbHandle
    установлен в положение BioAPI_DB_INVALID_HANDLE.
</description>
<equal_to var1="return"
var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE "/>
<equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE" />
</assert_condition>
<invoke activity="DetachAndUnload"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
    <input name="bspUuid" var="bspUuid" />
    <input name="BSPHandle" var="_bsphandle" />
</invoke>
```

```
    </activity>
</package>
```

8.60 Утверждение 18a*BioSPI_DbDelete_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbDelete* с недостоверным дескриптором ПБУ и проверки, является ли возвращенный код значением BioAPIERR_INVALID_BSP_HANDLE.

Выдержки

Подпункт 9.3.5.3

BioAPI_RETURN BioAPI BioSPI_DbCreate

(BioAPI_HANDLE BSPHandle,

*const BioAPI_UUID *DbUuid,*

uint32_t NumberOfRecords,

BioAPI_DB_ACCESS_TYPE AccessRequest,

*BioAPI_DB_HANDLE *DbHandle).*

Подпункт 8.5.4.1

Данная функция удаляет все записи из указанной базы данных ЗБИ и данные о состоянии, связанные с этой базой данных.

Ссылки: 9.3.5.4 и 8.5.4.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных.

4) Вызвать функцию *BioSPI_DbClose* для закрытия созданной базы данных.

5) Вызвать функцию *BioSPI_DbDelete* с недостоверным дескриптором модуля.

6) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_INVALID_BSP_HANDLE.

7) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbDelete* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="678e5d12-3d51-41ec-a672-13f34ea24545">
    <author>
        ISO/IEC JTC1 SC37
    </author>
    <description>
        Данный пакет содержит утверждение "BioSPI_DbDelete_
        InvalidBSPHandle".
    </description>
    <assertion name="BioSPI_DbDelete_InvalidBSPHandle"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью вызова
            функции BioSPI_DbDelete с недостоверным дескриптором
            ПБУ и проверки, является ли возвращенный код значением
            BioAPIERR_INVALID_BSP_HANDLE.
            Следующий текст соответствует приведенному в 8.5.4.1
            спецификации БиоАПИ 2.0.
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_DbDelete
                (BioAPI_HANDLE BSPHandle,
                const BioAPI_UUID *DbUuid).
            Данная функция удаляет все записи из указанной базы
            данных ЗБИ и данные о состоянии, связанные с этой
            базой данных.
            ------------------------------------------------
            8.5.4.2 Параметры
            BSPHandle (входной параметр) - дескриптор
            присоединенного поставщика услуг БиоАПИ.
            ------------------------------------------------
            Порядок действий при испытании на соответствие:
            1) Загрузить испытуемый ПБУ.
            2) Присоединить испытуемый ПБУ.
```

```
        3) Вызвать функцию BioSPI_DbCreate для создания базы
        данных.
        4) Вызвать функцию BioSPI_DbClose для закрытия
        созданной базы данных.
        5) Вызвать функцию BioSPI_DbDelete с недостоверным
        дескриптором модуля.
        6) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPIERR_INVALID_BSP_HANDLE.
        7) Отсоединить и выгрузить ПБУ.
        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>
<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>
<!-- Указывает на требование испытуемым ПБУ поддержки
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />
<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>
<!-- База данных Uuid будет открыта. -->
<input name="_dbUuid"/>
<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbDelete">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
```

```
     var="_sourcepresenttimeout"/>
     <input name="dbUuid" var="_dbUuid"/>
</invoke>
<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически. -->
<bind activity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>
<activity name="BioSPI_DbDelete">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported" />
     <input name="sourcepresenttimeouttime"/>
     <input name="dbUuid"/>
<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>
<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<invoke activity="PrepareDBTesting"
```

```
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspHandle" var="_bsphandle" />
     <input name="dbUuid" var="dbUuid" />
     <input name="nosourcepresentsupported"
     var="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"
     var="sourcepresenttimeouttime"/>
</invoke>

<!-- Вызвать функцию BioSPI_DbDelete с недостоверным
дескриптором ПБУ. -->
<invoke function="BioSPI_DbDelete">
     <input name="BSPHandle" value="0"/>
     <input name="DbUuid" var="dbUuid"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
<description>
     Функция BioSPI_DbDelete возвращает значение
     BioAPIERR_INVALID_BSP_HANDLE.
</description>
<equal_to var1="return"
var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
</assert_condition>

<invoke activity="CleanUpDBTesting"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="BSPHandle" var="_bsphandle" />
```

```
            <input name="dbUuid" var="dbUuid" />
        </invoke>
        <!-- Вызвать функцию BioSPI_BSPDetach и
        BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.61 Утверждение 18b *BioSPI_DbDelete_OpenDbProtected*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbDelete* при открытой базе данных и проверки, является ли возвращенный код значением BioAPIERR_DATABASE_IS_OPEN.

Выдержки

Подпункт 9.3.5.4

BioAPI_RETURN BioAPI BioSPI_DbDelete

(BioAPI_HANDLE BSPHandle,

*const BioAPI_UUID *DbUuid).*

Подпункт 8.5.4.1

Данная функция удаляет все записи из указанной базы данных ЗБИ и данные о состоянии, связанные с этой базой данных.

Ссылки: 9.3.5.4 и 8.5.4.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных.

4) Вызвать функцию *BioSPI_DbDelete* с достоверными параметрами.

5) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_DATABASE_IS_OPEN.

6) Вызвать функцию *BioSPI_DbClose* для закрытия базы данных.

7) Вызвать функцию *BioSPI_DbDelete* для удаления базы данных.

8) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbDelete* возвращает значение BioAPIERR_DATABASE_IS_OPEN.

Пакет языка утверждения

```
<package name="9e4d0c6d-4d59-479c-9f58-160ecde99aad">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_DbDelete_
          OpenDbProtected".
     </description>
     <assertion name="BioSPI_DbDelete_OpenDbProtected"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью вызова
               функции BioSPI_DbDelete при открытой базе данных и
               проверки, является ли возвращенный код значением
               BioAPIERR_DATABASE_IS_OPEN.
               Приведенный ниже текст соответствует приведенному в
               8.5.4.1 спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioSPI_DbDelete
                    (BioAPI_HANDLE BSPHandle,
                    const BioAPI_UUID *DbUuid).
               Данная функция удаляет все записи из определенной
               базы данных ЗБИ и данные о состоянии, связанные с
               этой базой данных.
               Ошибки: BioAPIERR_DATABASE_IS_OPEN.
```

```
      ------------------------------------------------
      Порядок действий при испытании на соответствие:

      1) Загрузить испытуемый ПБУ.
      2) Присоединить испытуемый ПБУ.
      3) Вызвать функцию BioSPI_DbCreate для создания базы
      данных.
      4) Вызвать функцию BioSPI_DbDelete с достоверными
      параметрами.
      5) Проверить возвращаемый код. Предполагается
      возвращение значения BioAPIERR_DATABASE_IS_OPEN.
      6) Вызвать функцию BioSPI_DbClose для закрытия базы
      данных.
      7) Вызвать функцию BioSPI_DbDelete для удаления базы
      данных.
      8) Отсоединить и выгрузить ПБУ.
      Если какие-либо промежуточные операции прошли
      неуспешно, то выдается заключение о соответствии
      "UNDECIDED".
</description>
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>
<!-- Указание на требование испытуемым ПБУ поддержки
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- База данных Uuid будет открыта. -->
<input name="_dbUuid"/>
```

```
        <!-- Вызов основного процесса данного утверждения со
        значениями входных параметров, определенными значениями
        параметров утверждения. -->
        <invoke activity="BioSPI_DbDelete">
             <input name="bspUuid" var="_bspUuid"/>
             <input name="inserttimeouttime"
             var="_inserttimeout"/>
             <input name="nosourcepresentsupported"
             var="_noSourcePresentSupported" />
             <input name="sourcepresenttimeouttime"
             var="_sourcepresenttimeout"/>
             <input name="dbUuid" var="_dbUuid"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически.-->
        <bind activity="EventHandler"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        function="BioSPI_EventHandler"/>
   </assertion>
   <activity name="BioSPI_DbDelete">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="dbUuid"/>
        <input name="readAccessRequest"/>
        <input name="writeAccessRequest"/>
        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>
        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
```

```
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>
<!-- Удалить базу данных, если она существует. -->
<invoke function="BioSPI_DbDelete" >
     <input name="BSPHandle" var="_bsphandle" />
     <input name="DbUuid" var="dbUuid" />
     <return setvar="return"/>
</invoke>
<!-- Вызывать функцию BioSPI_DbCreate для создания базы
данных. -->
<invoke function="BioSPI_DbCreate">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="NumberOfRecords" value="1"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
```

```
<description>
      Функция BioSPI_DbCreate возвращает значение
     BioAPI_OK.
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbDelete с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbDelete">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
<description>
     Функция BioSPI_DbDelete возвращает значение
     BioAPIERR_DATABASE_IS_OPEN.
</description>
<equal_to var1="return"
var2="__BioAPIERR_BSP_DATABASE_IS_OPEN"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
```

```
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
<description>
     Функция BioSPI_DbClose возвращает значение
     BioAPI_OK.
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbDelete с достоверными
значениями входных параметров для удаления созданной базы
данных. -->
<invoke function="BioSPI_DbDelete">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
<description>
     Функция BioSPI_DbDelete возвращает значение
     BioAPI_OK.
</description>
<equal_to var1="return" var2="__BioAPI_OK "/>
</assert_condition>
<!-- Вызвать функции BioSPI_BSPDetach и BioSPI_BSPUnload. -->
<invoke activity="DetachAndUnload"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
```

```
                <input name="bspUuid" var="bspUuid" />
                <input name="BSPHandle" var="_bsphandle" />
            </invoke>
        </activity>
</package>
```

8.62 Утверждение 18c *BioSPI_DbDelete_ValidParam*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_DbDelete* с достоверными значениями входных параметров.

Выдержки

Подпункт 9.3.5.4

BioAPI_RETURN BioAPI BioSPI_DbDelete

(BioAPI_HANDLE BSPHandle,

*const BioAPI_UUD *DbUuid).*

Подпункт 8.5.4.1

Данная функция удаляет все записи из указанной базы данных ЗБИ и данные о состоянии, связанные с этой базой данных.

Ссылки: 9.3.5.4 и 8.5.4.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных.

4) Вызвать функцию *BioSPI_DbClose* для закрытия созданной базы данных.

5) Вызвать функцию *BioSPI_DbDelete* для удаления созданной базы данных.

6) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

7) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbDelete* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="499d9cc3-4269-4671-9d69-29a31bc1a08f">

    <author>
        ISO/IEC JTC1 SC37
    </author>

    <description>
        Данный пакет содержит утверждение "BioSPI_DbDelete_ Va-
        lidParam".
    </description>
    <assertion name="BioSPI_DbDelete_ValidParam"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью
            определения возвращения значения BioAPI_OK при вызове
            функции BioSPI_DbDelete с достоверными значениями
            входных параметров.
            Приведенный ниже текст соответствует приведенному в
            8.5.4.1 спецификации БиоАПИ 2.0.
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_DbDelete
                (BioAPI_HANDLE BSPHandle,
                constBioAPI_UUID *DbUuid).
            Данная функция удаляет все записи из указанной базы
            данных ЗБИ и данные о состоянии, связанные с этой
            базой данных.
            ------------------------------------------------
            Порядок действий при испытании на соответствие:

            1) Загрузить испытуемый ПБУ.
            2) Присоединить испытуемый ПБУ.
            3) Вызвать функцию BioSPI_DbCreate для создания базы
            данных.
```

```
        4) Вызвать функцию BioSPI_DbClose для закрытия
        созданной базы данных.
        5) Вызвать функцию BioSPI_DbDelete для удаления
        созданной базы данных.
        6) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPI_OK.
        7) Отсоединить и выгрузить ПБУ.
</description>
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указывает на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- База данных Uuid будет открыта. -->
<input name="_dbUuid"/>
<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbDelete">
        <input name="bspUuid" var="_bspUuid"/>
        <input name="inserttimeouttime"
        var="_inserttimeout"/>
        <input name="nosourcepresentsupported"
        var="_noSourcePresentSupported" />
        <input name="sourcepresenttimeouttime"
        var="_sourcepresenttimeout"/>
```

```
            <input name="dbUuid" var="_dbUuid"/>
        </invoke>
        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        function="BioSPI_EventHandler"/>
    </assertion>
    <activity name="BioSPI_DbDelete">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="dbUuid"/>
        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>
        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->

        <invoke activity="LoadAndAttach"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
            <input name="bspVersion" value="32"/>
            <input name="unitIDOrNull" value="0"/>
            <input name="bspHandle" var="_bsphandle"/>
            <input name="eventtimeouttime"
            var="inserttimeouttime"/>
        </invoke>
        <invoke activity="PrepareDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
```

```
        break_on_break="true">
            <input name="bspHandle" var="_bsphandle" />
            <input name="dbUuid" var="dbUuid" />
            <input name="nosourcepresentsupported"
            var="nosourcepresentsupported"/>
            <input name="sourcepresenttimeouttime"
            var="sourcepresenttimeouttime"/>
        </invoke>
        <!-- Вызвать функцию BioSPI_DbDelete с достоверными
        значениями входных параметров. -->
        <invoke function="BioSPI_DbDelete">
            <input name="BSPHandle" var="_bsphandle"/>
            <input name="DbUuid" var="dbUuid"/>
            <return setvar="return"/>
        </invoke>
        <!-- Выдача заключения о соответствии. Если условие, оп-
        ределенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition>
        <description>
            Функция BioSPI_DbDelete возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <!-- Вызвать функции BioSPI_ModuleDetach и
        BioSPI_ModuleUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
```

```
</package>
```

8.63 Утверждение19а *BioSPI_DbSetMarker_ValidParam*

Описание: Проверку данного утверждения проводят с целью определения возвращения значения BioAPI_OK при вызове функции *BioSPI_DbSetMarker* с достоверными значениями входных параметров.

Выдержки

Подпункт 9.3.5.4

BioAPI_RETURN BioAPI BioSPI_DbDelete

(BioAPI_HANDLE BSPHandle,

*const BioAPI_UUID *DbUuid).*

Подпункт 8.5.5.1

Маркеру, идентифицированному параметром MarkerHandle, присваивается указатель на запись, представленную параметром KeyValue в базе данных ЗБИ, идентифицированной параметром DbHandle. Значение NULL установит маркер на первую запись в базе данных.

Ссылки: 9.3.5.5 и 8.5.5.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbOpen* для открытия базы данных.

4) Вызвать функцию *BioSPI_DbSetMarker* с достоверными значениями входных параметров.

5) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbSetMarker* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="94271080-e723-11d9-898c-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_DbSetMarker_
          ValidParam".
     </description>
     <assertion name="BioSPI_DbSetMarker_ValidParam"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью
               определения возвращения значения BioAPI_OK при
               вызове функции BioSPI_DbSetMarker с достоверными
               значениями входных параметров.
               Приведенный ниже текст соответствует приведенному в
               8.5.5.1 спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioAPI_DbSetMarker
                    (BioAPI_HANDLE BSPHandle,
                    BioAPI_DB_HANDLE DbHandle,
                    const BioAPI_UUID *KeyValue,
                    BioAPI_DB_MARKER_HANDLE MarkerHandle).
               Маркеру, идентифицированному параметром MarkerHandle,
               присваивается указатель на запись, представленную
               параметром KeyValue в базе данных ЗБИ, идентифициро-
               ванной параметром DbHandle. Значение NULL установит
               маркер на первую запись в базе данных.
               ------------------------------------------------
               Порядок действий при испытании на соответствие:
               1) Загрузить испытуемый ПБУ.
               2) Присоединить испытуемый ПБУ.
```

```
        3) Вызвать функцию BioSPI_DbOpen для открытия базы
        данных.
        4) Вызвать функцию BioSPI_DbSetMarker с достоверными
        значениями входных параметров.
        5) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPI_OK.
        6) Отсоединить и выгрузить ПБУ.
        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- База данных Uuid будет открыта. -->
<input name="_dbUuid"/>
<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbSetMarker">
        <input name="bspUuid" var="_bspUuid"/>
        <input name="inserttimeouttime"
        var="_inserttimeout"/>
        <input name="nosourcepresentsupported"
```

```
            var="_noSourcePresentSupported" />
            <input name="sourcepresenttimeouttime"
            var="_sourcepresenttimeout"/>
            <input name="dbUuid" var="_dbUuid"/>
        </invoke>

        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        function="BioSPI_EventHandler"/>

    </assertion>
    <activity name="BioSPI_DbSetMarker">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="dbUuid"/>
        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>
        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
            <input name="bspVersion" value="32"/>
            <input name="unitIDOrNull" value="0"/>
            <input name="bspHandle" var="_bsphandle"/>
```

```
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
</invoke>
<invoke activity="PrepareDBTesting"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
        <input name="bspHandle" var="_bsphandle" />
        <input name="dbUuid" var="dbUuid" />
        <input name="nosourcepresentsupported"
        var="nosourcepresentsupported"/>
        <input name="sourcepresenttimeouttime"
        var="sourcepresenttimeouttime"/>
        <output name="biruuid" setvar="biruuid"/>
</invoke>
<!-- Вызвать функцию BioSPI_DbOpen для открытия базы
данных. -->
<invoke function="BioSPI_DbOpen">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="DbUuid" var="dbUuid"/>
        <input name="ReadAccessRequest" value="true"/>
        <input name="WriteAccessRequest" value="true"/>
        <output name="DbHandle" setvar="dbHandle"/>
        <output name="MarkerHandle" setvar="markerHandle"/>
        <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >
<description>
        Функция BioSPI_DbOpen возвращает значение BioAPI_OK
        и выходной дескриптор dbHandle является достоверным
        дескриптором DB.
```

```
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbSetMarker с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbSetMarker" >
     <input name="BSPHandle" var="_bsphandle" />
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="biruuid"/>
     <input name="MarkerHandle" var="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
<description>
     Функция BioSPI_DbSetMarker возвращает значение
     BioAPI_OK.
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
```

```
        выдается заключение о соответствии "UNDECIDED" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
        break_if_false="true" >
        <description>
             Функция BioSPI_DbClose возвращает значение
             BioAPI_OK.
        </description>

        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <invoke activity="CleanUpDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
             <input name="BSPHandle" var="_bsphandle" />
             <input name="dbUuid" var="dbUuid" />
        </invoke>
        <!-- Вызвать функции BioSPI_ModuleDetach и
        BioSPI_ModuleUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
             <input name="bspUuid" var="bspUuid" />
             <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.64 Утверждение 19b *BioSPI_DbSetMarker_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbSetMarker* с недостоверным дескриптором ПБУ и проверки, является ли возвращенный код значением BioAPIERR_INVALID_BSP_HANDLE.

Выдержки

Подпункт 9.3.5.5

BioAPI_RETURN BioAPI BioSPI_DbSetMarker

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_HANDLE DbHandle,

*const BioAPI_UUID *KeyValue,*

BioAPI_DB_MARKER_HANDLE MarkerHandle).

Подпункт 8.5.5.1

Маркеру, идентифицированному параметром MarkerHandle, присваивается указатель на запись, представленную параметром KeyValue в базе данных ЗБИ, идентифицированной параметром DbHandle. Значение NULL установит маркер на первую запись в базе данных.

Ссылки: 9.3.5.5 и 8.5.5.1

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbOpen* для открытия базы данных.

4) Вызвать функцию *BioSPI_DbSetMarker* с недостоверным дескриптором ПБУ.

5) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_INVALID_BSP_HANDLE.

6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbSetMarker* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="69f7ce20-e72e-11d9-b4e0-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>
    <description>
        Данный пакет содержит утверждение "BioSPI_DbSetMarker_
        InvalidBSPHandle".
```

```
</description>
<assertion name="BioSPI_DbSetMarker_InvalidBSPHandle"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью вызова
          функции BioSPI_DbSetMarker с недостоверным дескриптором
          ПБУ и проверки, является ли возвращенный код
          значением BioAPIERR_INVALID_BSP_HANDLE.
          Приведенный ниже текст соответствует приведенному в
          8.5.5.1 спецификации БиоАПИ 2.0.
          ---------------------------------------------------
          BioAPI_RETURN BioAPI BioAPI_DbSetMarker
               (BioAPI_HANDLE BSPHandle,
               BioAPI_DB_HANDLE DbHandle,
               const BioAPI_UUID *KeyValue,
               BioAPI_DB_MARKER_HANDLE MarkerHandle).
          Маркеру, идентифицированному параметром MarkerHandle,
          присваивается указатель на запись, представленную
          параметром KeyValue в базе данных ЗБИ, идентифициро-
          ванной параметром DbHandle. Значение NULL установит
          маркер на первую запись в базе данных.
          ---------------------------------------------------
          8.5.5.2 Параметры
          BSPHandle (входной параметр) - дескриптор
          присоединенного поставщика услуг БиоАПИ.
          ---------------------------------------------------
          Порядок действий при испытании на соответствие:
          1) Загрузить испытуемый ПБУ.
          2) Присоединить испытуемый ПБУ.
          3) Вызвать функцию BioSPI_DbOpen для открытия базы
          данных.
          4) Вызвать функцию BioSPI_DbSetMarker с недостоверным
          дескриптором ПБУ.
          5) Проверить возвращаемый код. Предполагается
          возвращение значения BioAPIERR_INVALID_BSP_HANDLE.
          6) Отсоединить и выгрузить ПБУ.
```

```
        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>

    <!-- Указание на отсутствие поддержки испытуемым ПБУ
    уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
    <input name="_noSourcePresentSupported" />

    <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
    PRESENT. -->
    <input name="_sourcepresenttimeout"/>

    <!-- База данных Uuid будет открыта. -->
    <input name="_dbUuid"/>
    <!--Вызов основного процесса данного утверждения со
    значениями входных параметров, определенными значениями
    параметров утверждения. -->
    <invoke activity="BioSPI_DbSetMarker">
        <input name="bspUuid" var="_bspUuid"/>
        <input name="inserttimeouttime"
        var="_inserttimeout"/>
        <input name="nosourcepresentsupported"
        var="_noSourcePresentSupported" />
        <input name="sourcepresenttimeouttime"
        var="_sourcepresenttimeout"/>
        <input name="dbUuid" var="_dbUuid"/>
    </invoke>
```

```
        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        function="BioSPI_EventHandler"/>
    </assertion>
    <activity name="BioSPI_DbSetMarker">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="dbUuid"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
            <input name="bspVersion" value="32"/>
            <input name="unitIDOrNull" value="0"/>
            <input name="bspHandle" var="_bsphandle"/>
            <input name="eventtimeouttime"
            var="inserttimeouttime"/>
        </invoke>
        <invoke activity="PrepareDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
```

```
        <input name="bspHandle" var="_bsphandle" />
        <input name="dbUuid" var="dbUuid" />
        <input name="nosourcepresentsupported"
        var="nosourcepresentsupported"/>
        <input name="sourcepresenttimeouttime"
        var="sourcepresenttimeouttime"/>
        <output name="biruuid" setvar="biruuid"/>
</invoke>

<!-- Вызвать функцию BioSPI_DbOpen для открытия базы
данных. -->
<invoke function="BioSPI_DbOpen">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="DbUuid" var="dbUuid"/>
        <input name="ReadAccessRequest" value="true"/>
        <input name="WriteAccessRequest" value="true"/>
        <output name="DbHandle" setvar="dbHandle"/>
        <output name="MarkerHandle" setvar="markerHandle"/>
        <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >

<description>
        Функция BioSPI_DbOpen возвращает значение BioAPI_OK
        и выходной дескриптор dbHandle явялется достоверным
        дескриптором DB.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
```

```
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbSetMarker с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbSetMarker" >
     <input name="BSPHandle" value="0" />
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="biruuid"/>
     <input name="MarkerHandle" var="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition >

<description>
     Функция BioSPI_DbSetMarker возвращает значение
     BioAPIERR_INVALID_BSP_HANDLE.
</description>
<equal_to var1="return"
var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE "/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <return setvar="return"/>
</invoke>
```

```
        <!-- Выдача заключения о соответствии. Если условие, оп-
        ределенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
        break_if_false="true" >

        <description>
             Функция BioSPI_DbClose возвращает значение
             BioAPI_OK.
        </description>

        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <invoke activity="CleanUpDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
             <input name="BSPHandle" var="_bsphandle" />
             <input name="dbUuid" var="dbUuid" />
        </invoke>
        <!-- Вызвать функции BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
             <input name="bspUuid" var="bspUuid" />
             <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.65 Утверждение19c *BioSPI_DbSetMarker_RecordNotFound*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbSetMarker* с ключевым значением, не существующим в базе данных, и проверки, является ли возвращенный код значением BioAPIERR_RECORD_NOT_FOUND.

Выдержки

Подпункт 9.3.5.5

BioAPI_RETURN BioAPI BioSPI_DbSetMarker

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_HANDLE DbHandle,

*const BioAPI_UUID *KeyValue,*

BioAPI_DB_MARKER_HANDLE MarkerHandle).

Подпункт 8.5.5.1

Маркеру, идентифицированному параметром MarkerHandle, присваивается указатель на запись, представленную параметром KeyValue в базе данных ЗБИ, идентифицированной параметром DbHandle. Значение NULL установит маркер на первую запись в базе данных.

Ссылки: 9.3.5.5 и 8.5.5.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbOpen* для открытия базы данных.

4) Вызвать функцию *BioSPI_DbSetMarker* с помощью ключевого значения, которого не существует в базе данных.

5) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_RECORD_NOT_FOUND.

6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbSetMarker* возвращает значение BioAPIERR_RECORD_NOT_FOUND.

Пакет языка утверждения

```
<package name="b9c90f40-e7ec-11d9-a435-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>
```

```
<description>
     Данный пакет содержит утверждение "BioSPI_DbSetMarker_
     RecordNotFound".
</description>
<assertion name="BioSPI_DbSetMarker_RecordNotFound"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью вызова
          BioSPI_DbSetMarker с ключевым значением, не существующим
          в базе данных, и проверки, является ли возвращенный
          код значением BioAPIERR_RECORD_NOT_FOUND.
          Приведенный ниже текст соответствует приведенному в
          8.5.5.1 спецификации БиоАПИ 2.0.
          ---------------------------------------------------
          BioAPI_RETURN BioAPI BioAPI_DbSetMarker
               (BioAPI_HANDLE BSPHandle,
               BioAPI_DB_HANDLE DbHandle,
               const BioAPI_UUID *KeyValue,
               BioAPI_DB_MARKER_HANDLE MarkerHandle).
          Маркеру, идентифицированному параметром MarkerHandle,
          присваивается указатель на запись, представленную
          параметром KeyValue в базе данных ЗБИ, идентифициро-
          ванной параметром DbHandle. Значение NULL установит
          маркер на первую запись в базе данных.
          ---------------------------------------------------
          Порядок действий при испытании на соответствие:

          1) Загрузить испытуемый ПБУ.
          2) Присоединить испытуемый ПБУ.
          3) Вызвать функцию BioSPI_DbOpen для открытия базы
          данных.
          4) Вызвать функцию BioSPI_DbSetMarker с помощью
          ключевого значения, которого не существует в базе
          данных.
```

```
     5) Проверить возвращаемый код. Предполагается
     возвращение значения BioAPIERR_RECORD_NOT_FOUND.
     6) Отсоединить и выгрузить ПБУ.
     Если какие-либо промежуточные операции прошли
     неуспешно, то выдается заключение о соответствии
     "UNDECIDED".
</description>
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- База данных Name, которая будет открыта. -->
<input name="_dbUuid"/>
<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbSetMarker">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="dbUuid" var="_dbUuid"/>
</invoke>
```

```
        <!-- Процесс связан с функцией интерфейса обратного вызова
        инфраструктуры, предоставленной испытуемым компонентом
        ПБУ. При каждом входящем вызове функции связанный с ней
        процесс будет вызываться автоматически. -->
        <bind activity="EventHandler"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        function="BioSPI_EventHandler"/>
    </assertion>
    <activity name="BioSPI_DbSetMarker">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="dbUuid"/>

        <!-- Данное утверждение использует BSPHandle со значением
        "1" для всех вызовов BioSPI, которые этого требуют. -->
        <set name="_bsphandle" value="1"/>

        <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
        предоставленных испытуемым ПБУ. Входное значение параметра
        "unitIDOrNull" равно "0", поэтому утверждение будет
        тестировать модуль датчика, выбранный ПБУ. -->
        <invoke activity="LoadAndAttach"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
            <input name="bspUuid" var="bspUuid"/>
            <input name="bspVersion" value="32"/>
            <input name="unitIDOrNull" value="0"/>
            <input name="bspHandle" var="_bsphandle"/>
            <input name="eventtimeouttime"
            var="inserttimeouttime"/>
        </invoke>
        <invoke activity="PrepareDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
```

```
break_on_break="true">
     <input name="bspHandle" var="_bsphandle" />
     <input name="dbUuid" var="dbUuid" />
     <input name="nosourcepresentsupported"
     var="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"
     var="sourcepresenttimeouttime"/>
     <output name="biruuid" setvar="biruuid"/>
</invoke>
<!-- Вызвать функцию BioSPI_DbOpen для открытия базы
данных. -->
<invoke function="BioSPI_DbOpen">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >

<description>
     Функция BioSPI_DbOpen возвращает значение BioAPI_OK
     и выходной дескриптор dbHandle является достоверным
     дескриптором базы данных.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
```

```
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>
<set name="nonExistingKey"
value="ffffffff-ffff-ffff-ffff-ffffffffffff">
     <only_if>
          <not_same_as var1="biruuid"
          value2="ffffffff-ffff-ffff-ffff-ffffffffffff"/>
     </only_if>
</set>

<set name="nonExistingKey"
value="00000000-0000-0000-0000-000000000001">
     <only_if>
          <same_as var1="biruuid"
          value2="ffffffff-ffff-ffff-ffff-ffffffffffff"/>
     </only_if>
</set>

<!-- Вызвать функцию BioSPI_DbSetMarker с достоверными
входными праметрами. -->
<invoke function="BioSPI_DbSetMarker" >
     <input name="BSPHandle" var="_bsphandle" />
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="nonExistingKey"/>
     <input name="MarkerHandle" var="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition >
```

```
<description>
     Функция BioSPI_DbSetMarker возвращает значение
     BioAPIERR_RECORD_NOT_FOUND.
</description>
<equal_to var1="return"
var2="__BioAPIERR_BSP_RECORD_NOT_FOUND "/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >

<description>
     Функция BioSPI_DbClose возвращает значение
     BioAPI_OK.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<invoke activity="CleanUpDBTesting"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="BSPHandle" var="_bsphandle" />
     <input name="dbUuid" var="dbUuid" />
</invoke>
```

```
        <!-- Вызвать функцию BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.66 Утверждение 20a *BioSPI_DbFreeMarker_ValidParam*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbFreeMarker* с достоверными значениями входных параметров и проверки, является ли возвращенный код значением BioAPI_OK.

Выдержки

Подпункт 9.3.5.5

BioAPI_RETURN BioAPI BioSPI_DbSetMarker

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_HANDLE DbHandle,

*const BioAPI_UUID *KeyValue,*

BioAPI_DB_MARKER_HANDLE MarkerHandle).

Подпункт 8.5.6.1

Освобождает память и ресурсы, связанные с определенным маркером, и объявляет недостоверным MarkerHandle.

Ссылки: 9.3.5.6 и 8.5.6.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbOpen* для открытия базы данных.

4) Вызвать функцию *BioSPI_DbSetMarker* с достоверными значениями входных параметров.

5) Вызвать функцию *BioSPI_DbFreeMarker* с достоверными значениями входных параметров.

6) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

7) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbFreeMarker* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="2e1c9520-e7f1-11d9-a011-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_DbFreeMarker_
          ValidParam".
     </description>
     <assertion name="BioSPI_DbFreeMarker_ValidParam"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью вызова
               функции BioSPI_DbFreeMarker с достоверными значениями
               входных параметров и проверки, является ли возвращенный
               код значением BioAPI_OK.
               Приведенный ниже текст соответствует приведенному в
               8.5.6.1 спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioAPI_DbFreeMarker
                    (BioAPI_HANDLE BSPHandle,
                    BioAPI_DB_HANDLE DbHandle,
                    const BioAPI_UUID *KeyValue,
                    BioAPI_DB_MARKER_HANDLE MarkerHandle).
```

```
        Освобождает память и ресурсы, связанные с определенным
        маркером, и объявляет недостоверным MarkerHandle.
        ---------------------------------------------------
        Порядок действий при испытании на соответствие:

        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Вызвать функцию BioSPI_DbOpen для открытия базы
        данных.
        4) Вызвать функцию BioSPI_DbSetMarker с достоверными
        значениями входных параметров.
        5) Вызвать функцию BioSPI_DbFreeMarker с достоверными
        значениями входных параметров.
        6) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPI_OK.
        7) Отсоединить и выгрузить ПБУ.
        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>

    <!-- Указание на отсутствие поддержки испытуемым ПБУ
    уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
    <input name="_noSourcePresentSupported" />

    <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
    PRESENT. -->
    <input name="_sourcepresenttimeout"/>
```

```
<!-- База данных Name, которая будет открыта. -->
<input name="_dbUuid"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbFreeMarker">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="dbUuid" var="_dbUuid"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически.-->
<bind activity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>
<activity name="BioSPI_DbFreeMarker">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported" />
     <input name="sourcepresenttimeouttime"/>
     <input name="dbUuid"/>
<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>
```

```
<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>
<invoke activity="PrepareDBTesting"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspHandle" var="_bsphandle" />
     <input name="dbUuid" var="dbUuid" />
     <input name="nosourcepresentsupported"
     var="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"
     var="sourcepresenttimeouttime"/>
     <output name="biruuid" setvar="biruuid"/>
</invoke>

<!-- Вызвать функцию BioSPI_DbOpen для открытия базы
данных. -->
<invoke function="BioSPI_DbOpen">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
```

```
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >

<description>
    Функция BioSPI_DbOpen возвращает значение BioAPI_OK
    и выходной дескриптор dbHandle является достоверным
    дескриптором базы данных.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbSetMarker с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbSetMarker" >
    <input name="BSPHandle" var="_bsphandle" />
    <input name="DbHandle" var="dbHandle"/>
    <input name="KeyValue" var="biruuid"/>
    <input name="MarkerHandle" var="markerHandle"/>
    <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается заклю-
чение о соответствии "PASS". -->
```

```
<assert_condition response_if_false="undecided"
break_if_false="true">

<description>
     Функция BioSPI_DbSetMarker возвращает значение
     BioAPI_OK.
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbFreeMarker с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbFreeMarker">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="MarkerHandle" var="markerHandle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition >

<description>
     Функция BioSPI_DbFreeMarker возвращает значение
     BioAPI_OK.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных парметров. -->
<invoke function="BioSPI_DbClose">
```

```
            <input name="BSPHandle" var="_bsphandle"/>
            <input name="DbHandle" var="dbHandle"/>
            <return setvar="return"/>
        </invoke>
        <!-- Выдача заключения о соответствии. Если условие, оп-
        ределенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
        break_if_false="true" >

        <description>
            Функция BioSPI_DbClose возвращает значение
            BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <invoke activity="CleanUpDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
            <input name="BSPHandle" var="_bsphandle" />
            <input name="dbUuid" var="dbUuid" />
        </invoke>
        <!-- Вызвать функции BioSPI_ModuleDetach и
        BioSPI_ModuleUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.67 Утверждение 20b *BioSPI_DbFreeMarker_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbFreeMarker* с недостоверным дескриптором ПБУ и проверки, является ли возвращенный код значением BioAPIERR_INVALID_BSP_HANDLE.

Выдержки

Подпункт 9.3.5.6

BioAPI_RETURN BioAPI BioSPI_DbFreeMarker

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_MARKER_HANDLE MarkerHandle).

Подпункт 8.5.6.1

Освобождает память и ресурсы, связанные с определенным маркером и объявляет недостоверным MarkerHandle.

Ссылки: 9.3.5.6 и 8.5.6.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbOpen* для открытия базы данных.

4) Вызвать функцию *BioSPI_DbSetMarker* с достоверными значениями входных параметров.

5) Вызвать функцию *BioSPI_DbFreeMarker* с недостоверным дескриптором ПБУ.

6) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_INVALID_BSP_HANDLE.

7) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbFreeMarker* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="0f735140-e800-11d9-8a8a-0002a5d5c51b">
    <author>
```

```
     ISO/IEC JTC1 SC37
</author>

<description>
     Данный пакет содержит утверждение "BioSPI_DbFreeMarker_
     InvalidBSPHandle".
</description>
<assertion name="BioSPI_DbFreeMarker_InvalidBSPHandle"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью вызова
          функции BioSPI_DbFreeMarker с недостоверным дескриптором
          ПБУ и проверки, является ли возвращенный код
          значением BioAPIERR_INVALID_BSP_HANDLE.
          Следующий текст соответствует приведенному в 8.5.6.1
          спецификации БиоАПИ 2.0.
          ------------------------------------------------------
          BioAPI_RETURN BioAPI BioAPI_DbFreeMarker
               (BioAPI_HANDLE BSPHandle,
               BioAPI_DB_MARKER_HANDLE MarkerHandle).
          Освобождает память и ресурсы, связанные с определенным
          маркером и объявляет недостоверным MarkerHandle.
          ------------------------------------------------------
          8.5.6.2 Параметры
          BSPHandle (входной параметр) - дескриптор присоеди-
          ненного ПБУ.
          ------------------------------------------------------
          Порядок действий при испытании на соответствие:

          1) Загрузить испытуемый ПБУ.
          2) Присоединить испытуемый ПБУ.
          3) Вызвать функцию BioSPI_DbOpen для открытия базы
          данных.
          4) Вызвать функцию BioSPI_DbSetMarker с достоверными
          значениями входных параметров.
```

```
        5) Вызвать функцию BioSPI_DbFreeMarker с недостоверным
        дескриптором ПБУ.
        6) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPIERR_INVALID_BSP_HANDLE.
        7) Отсоединить и выгрузить ПБУ.
        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- База данных Name, которая будет открыта. -->
<input name="_dbUuid"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbFreeMarker">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
```

```
        <input name="sourcepresenttimeouttime"
        var="_sourcepresenttimeout"/>
        <input name="dbUuid" var="_dbUuid"/>
</invoke>
<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически. -->
<bind activity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>
<activity name="BioSPI_DbFreeMarker">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="dbUuid"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
```

```
</invoke>
<invoke activity="PrepareDBTesting"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspHandle" var="_bsphandle" />
     <input name="dbUuid" var="dbUuid" />
     <input name="nosourcepresentsupported"
     var="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"
     var="sourcepresenttimeouttime"/>
     <output name="biruuid" setvar="biruuid"/>
</invoke>
<!--Вызвать функцию BioSPI_DbOpen для открытия базы
данных. -->
<invoke function="BioSPI_DbOpen">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие, оп-
ределенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >

<description>
     Функция BioSPI_DbOpen возвращает значение BioAPI_OK и
     выходной дескриптор dbHandle является достоверным
     дескриптором БД.
```

```
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbSetMarker с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbSetMarker" >
     <input name="BSPHandle" var="_bsphandle" />
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="biruuid"/>
     <input name="MarkerHandle" var="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">

<description>
     Функция BioSPI_DbSetMarker возвращает значение
     BioAPI_OK.
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbFreeMarker с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbFreeMarker">
     <input name="BSPHandle" value="0"/>
     <input name="MarkerHandle" var="markerHandle"/>
     <return setvar="return"/>
```

```
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS'. -->
<assert_condition >

<description>
     Функция BioSPI_DbFreeMarker возвращает значение
     BioAPIERR_INVALID_BSP_HANDLE.
</description>

<equal_to var1="return"
var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >

<description>
```

```
            Функция BioSPI_DbClose возвращает значение
            BioAPI_OK.
        </description>

        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <invoke activity="CleanUpDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
            <input name="BSPHandle" var="_bsphandle" />
            <input name="dbUuid" var="dbUuid" />
        </invoke>

        <!-- Вызвать функции BioSPI_ModuleDetach и BioS-
        PI_ModuleUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.68 Утверждение 20c*BioSPI_DbFreeMarker_InvalidMarker*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbFreeMarker* с недостоверным дескриптором маркера и проверки, является ли возвращенный код значением BioAPIERR_MARKER_HANDLE_IS_INVALID.

Выдержки

Подпункт 9.3.5.6

BioAPI_RETURN BioAPI BioSPI_DbFreeMarker

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_MARKER_HANDLE MarkerHandle).

Подпункт 8.5.6.1

Освобождает память и ресурсы, связанные с определенным маркером и объявляет недостоверным MarkerHandle.

Ссылки: 9.3.5.6 и 8.5.6.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbOpen* для открытия определенной базы данных.

4) Вызвать функцию *BioSPI_DbFreeMarker* с недостоверным значением дескриптора маркера.

5) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_MARKER_HANDLE_IS_INVALID.

6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbFreeMarker* возвращает значение BioAPIERR_MARKER_HANDLE_IS_INVALID.

Пакет языка утверждения

```
<package name="a9007b60-e802-11d9-8a5d-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>

     <description>
          Данный пакет содержит утверждение "BioSPI_DbFreeMarker_
          InvalidMarker"
     </description>

     <assertion name="BioSPI_DbFreeMarker_InvalidMarker"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью вызова
               функции BioSPI_DbFreeMarker с недостоверным дескриптором
```

маркера и проверки, является ли возвращенный код значением BioAPIERR_MARKER_HANDLE_IS_INVALID. Следующий текст соответствует приведенному в 8.5.6.1 спецификации БиоАПИ 2.0.

BioAPI_RETURN BioAPI BioAPI_DbFreeMarker
(BioAPI_HANDLE BSPHandle,
BioAPI_DB_MARKER_HANDLE MarkerHandle).

Освобождает память и ресурсы, связанные с определенным маркером и объявляет недостоверным MarkerHandle.

Ошибки: BioAPIERR_MARKER_HANDLE_IS_INVALID.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию BioSPI_DbOpen для открытия определенной базы данных.

4) Вызвать функцию BioSPI_DbFreeMarker с недостоверным значением дескриптора маркера.

5) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_MARKER_HANDLE_IS_INVALID.

6) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

```
</description>
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />
```

```
<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- База данных Name, которая будет открыта. -->
<input name="_dbUuid"/>

<!--Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbFreeMarker">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="dbUuid" var="_dbUuid"/>
</invoke>
<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически.-->
<bind activity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>
<activity name="BioSPI_DbFreeMarker">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported" />
     <input name="sourcepresenttimeouttime"/>
     <input name="dbUuid"/>
```

```
<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>
<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<invoke activity="PrepareDBTesting"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspHandle" var="_bsphandle" />
     <input name="dbUuid" var="dbUuid" />
     <input name="nosourcepresentsupported"
     var="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"
     var="sourcepresenttimeouttime"/>
</invoke>
<!-- Вызвать функцию BioSPI_DbOpen для открытия базы
данных. -->
<invoke function="BioSPI_DbOpen">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
```

```
        <output name="DbHandle" setvar="dbHandle"/>
        <output name="MarkerHandle" setvar="markerHandle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
    break_if_false="true" >

    <description>
        Функция BioSPI_DbOpen возвращает значение BioAPI_OK
        и выходной дескриптор dbHandle является достоверным
        дескриптором базы данных.
    </description>

    <equal_to var1="return" var2="__BioAPI_OK"/>
    <not_equal_to var1="dbHandle"
    var2="__BioAPI_DB_INVALID_HANDLE"/>
    </assert_condition>
    <!-- Конструктив недостоверного маркера дескриптора,
    использующего достоверный маркер. -->
    <set name="invalidMarkerHandle" var="markerHandle"/>
    <add name="invalidMarkerHandle" value="1"/>

    <!-- Вызвать функцию BioSPI_DbFreeMarker с достоверным
    маркером дескриптора. -->
    <invoke function="BioSPI_DbFreeMarker">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="MarkerHandle" val-
        ue="invalidMarkerHandle"/>
        <return setvar="return"/>
    </invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition >
<description>
     Функция BioSPI_DbFreeMarker возвращает значение Bio-
     APIERR_MARKER_HANDLE_IS_INVALID.
</description>

<equal_to var1="return"
var2="__BioAPIERR_BSP_MARKER_HANDLE_IS_INVALID"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >

<description>
     Функция BioSPI_DbClose возвращает значение
     BioAPI_OK.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
```

```
            <invoke activity="CleanUpDBTesting"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
            break_on_break="true">
                 <input name="BSPHandle" var="_bsphandle" />
                 <input name="dbUuid" var="dbUuid" />
            </invoke>
            <!-- Вызвать функции BioSPI_ModuleDetach и BioS-
            PI_ModuleUnload. -->
            <invoke activity="DetachAndUnload"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
                 <input name="bspUuid" var="bspUuid" />
                 <input name="BSPHandle" var="_bsphandle" />
            </invoke>
      </activity>
</package>
```

8.69 Утверждение 21a*BioSPI_DbStoreBIR_ValidParam*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbStoreBIR* с достоверными значениями входных параметров и проверки, является ли возвращенный код значением BioAPI_OK, а также принимает ли выходной УУИД значение NULL.

Выдержки

Подпункт 9.3.5.6

BioAPI_RETURN BioAPI BioSPI_DbFreeMarker

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_MARKER_HANDLE MarkerHandle).

Подпункт 8.5.7.1

ЗБИ, идентифицированная параметром BIRToStore, сохраняется в открытой базе данных ЗБИ, идентифицированной параметром DbHandle. Если BIRToStore идентифицирован дескриптором ЗБИ, то входной дескриптор ЗБИ освобождается. Если BIRToStore идентифицирован ключевым значением базы данных, то ЗБИ восстанавливается и сохраняется путем копирования в открытой

базе данных. Новой ЗБИ в базе данных присваивается новый УУИД, который может быть использован в качестве ключевого значения для входа в ЗБИ в дальнейшем.

Ссылки: 9.3.5.7 и 8.5.7.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных.

4) Вызвать функцию *BioSPI_Enroll* для создания ЗБИ.

5) Вызвать функцию *BioSPI_DbStoreBIR* для сохранения зарегистрированной ЗБИ в базе данных.

6) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

7) Вызвать функцию *BioSPI_DbDelete* для удаления базы данных.

8) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbStoreBIR* возвращает значение BioAPI_OK и производит УУИД для сохраненной ЗБИ.

Пакет языка утверждения

```
<package name="da953680-e806-11d9-90d3-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>
     <description>
          Данный пакет содержит утверждение "BioSPI_DbStoreBIR_ Va-
          lidParam".
     </description>
     <assertion name="BioSPI_DbStoreBIR_ValidParam"
     model="BSPTesting">
          <description>
```

Проверку данного утверждения проводят с целью вызова функции BioSPI_DbStoreBIR с достоверными значениями входных параметров и проверки, является ли возвращенный код значением BioAPI_OK, а также принимает ли выходной УУИД значение NULL. Следующий текст соответствует приведенному в 8.5.7.1 спецификации БиоАПИ 2.0.

```
BioAPI_RETURN BioAPI BioSPI_DbStoreBIR
     (BioAPI_HANDLE BSPHandle,
     const BioAPI_INPUT_BIR *BIRToStore,
     BioAPI_DB_HANDLE DbHandle,
     BioAPI_UUID *BirUuid).
```

ЗБИ, идентифицированная параметром BIRToStore, сохраняется в открытой базе данных ЗБИ, идентифицированной параметром DbHandle. Если BIRToStore идентифицирован дескриптором ЗБИ, то входной дескриптор ЗБИ высвобождается. Если BIRToStore идентифицирован ключевым значением базы данных, то ЗБИ восстанавливается и сохраняется путем копирования в открытой базе данных. Новой ЗБИ в базе данных присваивается новый УУИД, и этот УУИД может быть использован в качестве ключевого значения для входа в ЗБИ в дальнейшем.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_DbCreate для создания базы данных.
4) Вызвать функцию BioSPI_Enroll для создания ЗБИ.
5) Вызвать функцию BioSPI_DbStoreBIR для сохранения зарегистрированной ЗБИ в базе данных.

```
      6) Проверить возвращаемый код. Предполагается
      возвращение значения BioAPI_OK.
      7) Вызвать функцию BioSPI_DbDelete для удаления базы
      данных.
      8) Отсоединить и выгрузить ПБУ.
      Если какие-либо промежуточные операции прошли
      неуспешно, то выдается заключение о соответствии
      "UNDECIDED".
</description>
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для захвата данных.-->
<input name="_capturetimeout"/>

<!-- База данных Name, которая будет открыта. -->
<input name="_dbUuid"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbStoreBIR">
      <input name="bspUuid" var="_bspUuid"/>
      <input name="inserttimeouttime"
      var="_inserttimeout"/>
```

```
        <input name="nosourcepresentsupported"
        var="_noSourcePresentSupported" />
        <input name="sourcepresenttimeouttime"
        var="_sourcepresenttimeout"/>
        <input name="capturetimeout" var="_capturetimeout"/>
        <input name="dbUuid" var="_dbUuid"/>
    </invoke>

    <!-- Процесс связан с функцией интерфейса обратного вызова
    инфраструктуры, предоставленной испытуемым компонентом
    ПБУ. При каждом входящем вызове функции связанный с ней
    процесс будет вызываться автоматически.-->
    <bind activity="EventHandler"
    package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
    function="BioSPI_EventHandler"/>
    </assertion>
    <activity name="BioSPI_DbStoreBIR">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeout"/>
        <input name="dbUuid"/>

    <!-- Данное утверждение использует BSPHandle со значением
    "1" для всех вызовов BioSPI, которые этого требуют. -->
    <set name="_bsphandle" value="1"/>

    <!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
    предоставленных испытуемым ПБУ. Входное значение параметра
    "unitIDOrNull" равно "0", поэтому утверждение будет тес-
    тировать модуль датчика, выбранный ПБУ. -->
    <invoke activity="LoadAndAttach"
    package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
    break_on_break="true">
```

```
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
</invoke>
<!-- Вызвать функцию BioSPI_DbDelete перед созданием базы
данных. -->
<invoke function="BioSPI_DbDelete" >
        <input name="BSPHandle" var="_bsphandle" />
        <input name="DbUuid" var="dbUuid" />
        <return setvar="return"/>
</invoke>

<!-- Вызвать функцию BioSPI_DbCreate для создания базы
данных. -->
<invoke function="BioSPI_DbCreate">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="DbUuid" var="dbUuid"/>
        <input name="NumberOfRecords" value="1"/>
        <input name="ReadAccessRequest" value="true"/>
        <input name="WriteAccessRequest" value="true"/>
        <output name="DbHandle" setvar="dbHandle"/>
        <return setvar="return"/>
</invoke>
<!-- Открыть базу данных. -->
<invoke function="BioSPI_DbOpen">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="DbUuid" var="dbUuid"/>
        <input name="ReadAccessRequest" value="true"/>
        <input name="WriteAccessRequest" value="true"/>
        <output name="DbHandle" setvar="dbHandle"/>
        <output name="MarkerHandle" setvar="markerHandle"/>
        <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
```

```
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">

<description>
     Если создание базы данных поддерживается ПБУ, то
     функция BioSPI_DbCreate возвращает значение BioAPI_OK
     и выходной dbHandle является достоверным дескриптором
     базы данных. Если создание базы данных не поддерживается
     ПБУ, то функция BioSPI_DbOpen возвращает значение
     BioAPI_OK и выходной dbHandle является достоверным
     дескриптором базы данных
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>
<set name="eventtimeoutflag" value="false"/>
<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
setvar="eventtimeoutflag">
<or var1="nosourcepresentsupported" var2="_sourcePresent" />
</wait_until>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполне-
ние процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
```

```
break_if_false="true">

<description>
     Испытуемый ПБУ не поддерживает уведомление о событии
     BioAPI_NOTIFY_SOURCE_PRESENT или уведомление о событии
     получено в пределах указанной максимальной
     продолжительности.
</description>

<not var="eventtimeoutflag"/>
</assert_condition>
<!-- ПБУ готов к получению ЗБИ. Вызвать функцию
BioSPI_Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Timeout" value="15000"/>
     <output name="NewTemplate" set-
     var="newtemplate_handle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">

<description>
     Функция BioSPI_Enroll возвращает значение BioAPI_OK.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
```

```
</assert_condition>
<!-- Вызвать функцию BioSPI_DbStoreBIR для сохранения
зарегистрированной ЗБИ в базе данных. -->
<invoke function="BioSPI_DbStoreBIR">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="BIRToStore_Form"
     var="__BioAPI_BIR_HANDLE_INPUT"/>
     <input name="BIRToStore_BIRHandle"
     var="newtemplate_handle"/>
     <input name="DbHandle" var="dbHandle"/>
     <input name="no_BirUuid" value="false"/>
     <output name="BirUuid" setvar="biruuid"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition break_if_false="true">

<description>
     Функция BioSPI_DbStoreBIR возвращает значение
     BioAPI_OK.
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbGetBIR с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbGetBIR">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="biruuid"/>
     <output name="RetrievedBIR" setvar="retrievedBir"/>
```

```
    <output name="MarkerHandle" setvar="markerHandle"/>
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>

<description>
    Функция BioSPI_DbGetBIR возвращает значение
    BioAPI_OK.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbClose">
    <input name="BSPHandle" var="_bsphandle"/>
    <input name="DbHandle" var="dbHandle"/>
    <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">

<description>
```

```
            Функция BioSPI_DbClose возвращает значение
            BioAPI_OK.
        </description>

        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <invoke activity="CleanUpDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
            <input name="BSPHandle" var="_bsphandle" />
            <input name="dbUuid" var="dbUuid" />
        </invoke>
        <!-- Вызвать функции BioSPI_ModuleDetach и BioS-
        PI_ModuleUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.70 Утверждение 21b *BioSPI_DbStoreBIR_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbStoreBIR* с недостоверным дескриптором ПБУ и проверки, является ли возвращенный код значением BioAPIERR_INVALID_BSP_HANDLE.

Выдержки

Подпункт 9.3.5.7

BioAPI_RETURN BioAPI BioSPI_DbStoreBIR

(BioAPI_HANDLE BSPHandle,

*const BioAPI_INPUT_BIR *BIRToStore,*

BioAPI_DB_HANDLE DbHandle,

*BioAPI_UUID *BirUuid.*

Подпункт 8.5.7.1

ЗБИ, идентифицированная параметром BIRToStore, сохраняется в открытой базе данных ЗБИ, идентифицированной параметром DbHandle. Если BIRToStore идентифицирован дескриптором ЗБИ, то входной дескриптор ЗБИ освобождается. Если параметр BIRToStore идентифицирован ключевым значением базы данных, то ЗБИ восстанавливается и сохраняется путем копирования в открытой базе данных. Новой ЗБИ в базе данных присваивается новый УУИД, который может быть использован в качестве ключевого значения для входа в ЗБИ в дальнейшем.

Ссылки: 9.3.5.7 и 8.5.7.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных.

4) Вызвать функцию *BioSPI_Enroll* для создания ЗБИ.

5) Вызвать функцию *BioSPI_DbStoreBIR* для сохранения зарегистрированной ЗБИ в базе данных с недостоверным дескриптором ПБУ.

6) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_INVALID_BSP_HANDLE.

7) Вызвать функцию *BioSPI_DbDelete* для удаления базы данных.

8) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbStoreBIR* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="e39027e0-e80b-11d9-85eb-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>
     <description>
```

```
        Данный пакет содержит утверждение "BioSPI_DbStoreBIR_
        InvalidBSPHandle".
    </description>
    <assertion name="BioSPI_DbStoreBIR_InvalidBSPHandle"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью вызова
            функции BioSPI_DbStoreBIR с недостоверным дескриптором
            ПБУ и проверки, является ли возвращенный код
            значением BioAPIERR_INVALID_BSP_HANDLE.
            Следующий текст соответствует приведенному в 8.5.7.1
            спецификации БиоАПИ 2.0.
            ---------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_DbStoreBIR
                (BioAPI_HANDLE BSPHandle,
                const BioAPI_INPUT_BIR *BIRToStore,
                BioAPI_DB_HANDLE DbHandle,
                BioAPI_UUID *BirUuid).
            ЗБИ, идентифицированная параметром BIRToStore,
            сохраняется в открытой базе данных ЗБИ, идентифици-
            рованной параметром DbHandle. Если BIRToStore иден-
            тифицирован дескриптором ЗБИ, то входной дескриптор
            ЗБИ высвобождается. Если BIRToStore идентифицирован
            ключевым значением базы данных, то ЗБИ восстанавли-
            вается и сохраняется путем копирования в открытой
            базе данных. Новой ЗБИ в базе данных присваивается
            новый УУИД, который может быть использован в качестве
            ключевого значения для входа в ЗБИ в дальнейшем.
            ---------------------------------------------------
            8.5.7.2 Параметры
            BSPHandle (входной параметр) - дескриптор присоеди-
            ненного ПБУ.
            ---------------------------------------------------=
            Порядок действий при испытании на соответствие:
```

```
        1) Загрузить испытуемый ПБУ.
        2) Присоединить испытуемый ПБУ.
        3) Вызвать функцию BioSPI_DbCreate для создания базы
        данных.
        4) Вызвать функцию BioSPI_Enroll для создания ЗБИ.
        5) Вызвать функцию BioSPI_DbStoreBIR для сохранения
        зарегистрированной ЗБИ в базе данных с недостоверным
        дескриптором ПБУ.
        6) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPIERR_INVALID_BSP_HANDLE.
        7) Вызвать функцию BioSPI_DbDelete для удаления базы
        данных.
        8) Отсоединить и выгрузить ПБУ.
        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для захвата данных. -->
<input name="_capturetimeout"/>

<!-- База данных Uuid будет открыта. -->
```

```
<input name="_dbUuid"/>

<!-- Вызов главного действия данного утверждения со
значениями входных параметров, определенных параметрами
утверждения. -->
<invoke activity="BioSPI_DbStoreBIR">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeout" var="_capturetimeout"/>
     <input name="dbUuid" var="_dbUuid"/>
</invoke>
<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически.-->
<bind activity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>

<activity name="BioSPI_DbStoreBIR">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported" />
     <input name="sourcepresenttimeouttime"/>
     <input name="capturetimeout"/>
     <input name="dbUuid"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
```

```
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<invoke function="BioSPI_DbDelete" >
     <input name="BSPHandle" var="_bsphandle" />
     <input name="DbUuid" var="dbUuid" />
     <return setvar="return"/>
</invoke>

<invoke function="BioSPI_DbCreate">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="NumberOfRecords" value="1"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
```

```
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
<description>
     Функция BioSPI_DbCreate возвращает значение
     BioAPI_OK.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<set name="eventtimeoutflag" value="false"/>
<!-- Если испытуемый ПБУ поддерживает уведомление о событии
BioAPI_NOTIFY_SOURCE_PRESENT, следует ждать получения
уведомления о событии в пределах указанной максимальной
продолжительности. -->
<wait_until timeout_var="sourcepresenttimeouttime"
setvar="eventtimeoutflag">
<or var1="nosourcepresentsupported" var2="_sourcePresent" />
</wait_until>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
<description>
     Испытуемый ПБУ не поддерживает уведомление о событии
     BioAPI_NOTIFY_SOURCE_PRESENT или уведомление о
     событии получено в пределах указанной максимальной
     продолжительности.
</description>
<not var="eventtimeoutflag"/>
</assert_condition>
```

```
<!-- ПБУ готов к получению ЗБИ. Вызвать функцию BioSPI_
Enroll для регистрации. -->
<invoke function="BioSPI_Enroll">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="Purpose"
     var="__BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY"/>
     <input name="Timeout" value="15000"/>
     <output name="NewTemplate" set-
     var="newtemplate_handle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">

<description>
     Функция BioSPI_Enroll возвращает значение BioAPI_OK.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbStoreBIR с целью сохранения
полученной ЗБИ в базе данных.-->
<invoke function="BioSPI_DbStoreBIR">
     <input name="BSPHandle" value="0"/>
     <input name="BIRToStore_Form"
     var="__BioAPI_BIR_HANDLE_INPUT"/>
     <input name="BIRToStore_BIRHandle"
     var="newtemplate_handle"/>
     <input name="DbHandle" var="dbHandle"/>
     <input name="no_BirUuid" value="false"/>
     <output name="BirUuid" setvar="uuid"/>
```

```
        <return setvar="return"/>
    </invoke>
    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "FAIL" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition>

    <description>
        Функция BioSPI_DbStoreBIR возвращает значение Bio-
        APIERR_INVALID_BSP_HANDLE.
    </description>

    <equal_to var1="return"
    var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE "/>
    </assert_condition>
    <!-- Вызвать функцию BioSPI_DbClose с достоверными
    значениями входных параметров. -->
    <invoke function="BioSPI_DbClose">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="DbHandle" var="dbHandle"/>
        <return setvar="return"/>
    </invoke>
    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
    break_if_false="true">
    <description>
        Функция BioSPI_DbClose возвращает значение
        BioAPI_OK.
    </description>
```

```
        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <invoke activity="CleanUpDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
             <input name="BSPHandle" var="_bsphandle" />
             <input name="dbUuid" var="dbUuid" />
        </invoke>
        <!-- Вызвать функции BioSPI_ModuleDetach и BioS-
        PI_ModuleUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
             <input name="bspUuid" var="bspUuid" />
             <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.71 Утверждение 22a *BioSPI_DbGetBIR_ValidParam*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbGetBIR* с достоверными значениями входных параметров и проверки, является ли возвращенный код значением BioAPI_OK.

Выдержки

Подпункт 9.3.5.8

BioAPI_RETURN BioAPI BioSPI_DbGetBIR

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_HANDLE DbHandle,

*const BioAPI_UUID *KeyValue,*

*BioAPI_BIR_HANDLE *RetrievedBIR,*

*BioAPI_DB_MARKER_HANDLE *MarkerHandle).*

Подпункт 8.5.8.1

Восстанавливается ЗБИ, идентифицированная параметром KeyValue в открытой базе данных ЗБИ, идентифицированной параметром DbHandle. ЗБИ копируется

в память ПБУ, и к ней возвращается дескриптор. Маркеру, идентифицированному параметром MarkerHandle, присваивается указатель на запись, представленную параметром KeyValue в базе данных ЗБИ, идентифицированной параметром DbHandle. Значение NULL установит маркер на первую запись в базе данных.

Ссылки: 9.3.5.8 и 8.5.8.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных.

4) Вызвать функцию *BioSPI_Enroll* для создания ЗБИ.

5) Вызвать функцию *BioSPI_DbStoreBIR* для сохранения зарегистрированной ЗБИ в базе данных.

6) Вызвать функцию *BioSPI_DbGetBIR* для восстановления сохраненной ЗБИ.

7) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK.

8) Вызвать функцию *BioSPI_DbDelete* для удаления базы данных.

9) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbGetBIR* возвращает значение BioAPI_OK.

Пакет языка утверждения

```
<package name="67512700-e811-11d9-a5e0-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>
     <description>
          Данный пакет содержит утверждение "BioSPI_DbGetBIR_ Va-
          lidParam".
     </description>
```

```
<assertion name="BioSPI_DbGetBIR_ValidParam"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью вызова
          функции BioSPI_DbGetBIR с достоверными значениями
          входных параметров и проверки, является ли
          возвращенный код значением BioAPI_OK.
          Приведенный ниже текст соответствует приведенному в
          8.5.8.1 спецификации БиоАПИ 2.0.
          ------------------------------------------------------
          BioAPI_RETURN BioAPI BioSPI_DbGetBIR
               (BioAPI_HANDLE BSPHandle,
               BioAPI_DB_HANDLE DbHandle,
               const BioAPI_UUID *KeyValue,
               BioAPI_BIR_HANDLE *RetrievedBIR,
               BioAPI_DB_MARKER_HANDLE *MarkerHandle).
          Восстанавливается ЗБИ, идентифицированная параметром
          KeyValue в открытой базе данных ЗБИ, идентифицированной
          параметром DbHandle. ЗБИ копируется в память ПБУ, и
          к ней возвращается дескриптор. Маркеру, идентифици-
          рованному параметром MarkerHandle, присваивается
          указатель на запись, представленную параметром
          KeyValue в базе данных ЗБИ, идентифицированной
          параметром DbHandle. Значение NULL установит маркер
          на первую запись в базе данных.
          ------------------------------------------------------
          Порядок действий при испытании на соответствие:
          1) Загрузить испытуемый ПБУ.
          2) Присоединить испытуемый ПБУ.
          3) Вызвать функцию BioSPI_DbCreate для создания базы
          данных.
          4) Вызвать функцию BioSPI_Enroll для создания ЗБИ.
          5) Вызвать функцию BioSPI_DbStoreBIR для сохранения
          зарегистрированной ЗБИ в базе данных.
          6) Вызвать функцию BioSPI_DbGetBIR для восстановления
          сохраненной ЗБИ.
```

```
        7) Проверить возвращаемый код. Предполагается
        возвращение значения BioAPI_OK.
        8) Вызвать функцию BioSPI_DbDelete для удаления базы
        данных.
        9) Отсоединить и выгрузить ПБУ.
        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>

    <!-- Указание на отсутствие поддержки испытуемым ПБУ
    уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
    <input name="_noSourcePresentSupported" />

    <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
    PRESENT. -->
    <input name="_sourcepresenttimeout"/>

    <!-- Время ожидания для захвата данных. -->
    <input name="_capturetimeout"/>

    <!-- База данных Uuid будет открыта. -->
    <input name="_dbUuid"/>

    <!-- Вызов основного процесса данного утверждения со
    значениями входных параметров, определенными значениями
    параметров утверждения. -->
    <invoke activity="BioSPI_DbGetBIR">
        <input name="bspUuid" var="_bspUuid"/>
```

```
        <input name="inserttimeouttime"
        var="_inserttimeout"/>
        <input name="nosourcepresentsupported"
        var="_noSourcePresentSupported" />
        <input name="sourcepresenttimeouttime"
        var="_sourcepresenttimeout"/>
        <input name="capturetimeout" var="_capturetimeout"/>
        <input name="dbUuid" var="_dbUuid"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически. -->
<bind activity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>
<activity name="BioSPI_DbGetBIR">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeout"/>
        <input name="dbUuid"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
```

```
break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<invoke activity="PrepareDBTesting"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspHandle" var="_bsphandle" />
     <input name="dbUuid" var="dbUuid" />
     <input name="nosourcepresentsupported"
     var="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"
     var="sourcepresenttimeouttime"/>
     <output name="biruuid" setvar="uuid"/>
</invoke>

<!-- Вызвать функцию BioSPI_DbOpen для открытия базы
данных. -->
<invoke function="BioSPI_DbOpen">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
```

```
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >

<description>
     Функция BioSPI_DbOpen возвращает значение BioAPI_OK
     и выходной дескриптор dbHandle является достоверным
     дескриптором DB.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbGetBIR с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbGetBIR">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="uuid"/>
     <output name="RetrievedBIR" set-
     var="retrievedBir_handle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>

<description>
```

```
        Функция BioSPI_DbGetBIR возвращает значение
        BioAPI_OK.
    </description>

    <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>
    <!-- Вызвать функцию BioSPI_DbClose с достоверными
    значениями входных параметров. -->
    <invoke function="BioSPI_DbClose">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="DbHandle" var="dbHandle"/>
        <return setvar="return"/>
    </invoke>
    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии'PASS". -->
    <assert_condition response_if_false="undecided"
    break_if_false="true">

    <description>
        Функция BioSPI_DbClose возвращает значение
        BioAPI_OK.
    </description>

    <equal_to var1="return" var2="__BioAPI_OK"/>
    </assert_condition>
    <invoke activity="CleanUpDBTesting"
    package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
    break_on_break="true">
        <input name="BSPHandle" var="_bsphandle" />
        <input name="dbUuid" var="dbUuid" />
    </invoke>
    <!-- Вызвать функции BioSPI_BSPDetach и BioSPI_BSPUnload. -->
```

```
            <invoke activity="DetachAndUnload"
            package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
                  <input name="bspUuid" var="bspUuid" />
                  <input name="BSPHandle" var="_bsphandle" />
            </invoke>
      </activity>
</package>
```

8.72 Утверждение 22b *BioSPI_DbGetBIR_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbGetBIR* с недостоверным дескриптором ПБУ и проверки, является ли возвращенный код значением BioAPIERR_INVALID_BSP_HANDLE.

Выдержки

Подпункт 9.3.5.8

BioAPI_RETURN BioAPI BioSPI_DbGetBIR

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_HANDLE DbHandle,

*const BioAPI_UUID *KeyValue,*

*BioAPI_BIR_HANDLE *RetrievedBIR,*

*BioAPI_DB_MARKER_HANDLE *MarkerHandle).*

Подпункт 8.5.8.1

Восстанавливается ЗБИ, идентифицированная параметром KeyValue в открытой базе данных ЗБИ, идентифицированной параметром DbHandle. ЗБИ копируется в память ПБУ, и к ней возвращается дескриптор. Маркеру, идентифицированному параметром MarkerHandle, присваивается указатель на запись, представленную параметром KeyValue в базе данных ЗБИ, идентифицированной параметром DbHandle. Значение NULL установит маркер на первую запись в базе данных.

Ссылки: 9.3.5.8 и 8.5.8.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных.

4) Вызвать функцию *BioSPI_Enroll* для создания ЗБИ.

5) Вызвать функцию *BioSPI_DbStoreBIR* для сохранения зарегистрированной ЗБИ в базе данных.

6) Вызвать функцию *BioSPI_DbGetBIR* с недостоверным дескриптором ПБУ.

7) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_INVALID_BSP_HANDLE.

8) Вызвать функцию *BioSPI_DbDelete* для удаления базы данных.

9) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbGetBIR* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="f62947e0-e8b7-11d9-9dad-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>
    <description>
        Данный пакет содержит утверждение "BioSPI_DbGetBIR_ Inva-
        lidBSPHandle".
    </description>
    <assertion name="BioSPI_DbGetBIR_InvalidBSPHandle"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью вызова
            функции BioSPI_DbGetBIR с недостоверным дескриптором
            ПБУ и проверки, является ли возвращенный код значе-
            нием BioAPIERR_INVALID_BSP_HANDLE.
            Следующий текст соответствует приведенному в 8.5.8.1
            спецификации БиоАПИ 2.0.
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioSPI_DbGetBIR
```

```
    (BioAPI_HANDLE BSPHandle,
    BioAPI_DB_HANDLE DbHandle,
    const BioAPI_UUID *KeyValue,
    BioAPI_BIR_HANDLE *RetrievedBIR,
    BioAPI_DB_MARKER_HANDLE *MarkerHandle).
```

Восстанавливается ЗБИ, идентифицированная параметром KeyValue в открытой базе данных ЗБИ, идентифицированной параметром DbHandle. ЗБИ копируется в память ПБУ, и к ней возвращается дескриптор. Маркеру, идентифицированному параметром MarkerHandle, присваивается указатель на запись, представленную параметром KeyValue в базе данных ЗБИ, идентифицированной параметром DbHandle. Значение NULL установит маркер на первую запись в базе данных.

8.5.8.2 Параметр BSPHandle (входной параметр) - дескриптор присоединенного ПБУ.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию BioSPI_DbCreate для создания базы данных.

4) Вызвать функцию BioSPI_Enroll для создания ЗБИ.

5) Вызвать функцию BioSPI_DbStoreBIR для сохранения зарегистрированной ЗБИ в базе данных.

6) Вызвать функцию BioSPI_DbGetBIR с недостоверным дескриптором ПБУ.

7) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_INVALID_BSP_HANDLE.

8) Вызвать функцию BioSPI_DbDelete для удаления базы данных.

9) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

```
</description>
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для захвата данных. -->
<input name="_capturetimeout"/>

<!-- База данных Uuid будет открыта. -->
<input name="_dbUuid"/>

<!--Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbGetBIR">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeout" var="_capturetimeout"/>
     <input name="dbUuid" var="_dbUuid"/>
</invoke>
```

```
<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически.-->
<bind activity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>
<activity name="BioSPI_DbGetBIR">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported" />
     <input name="sourcepresenttimeouttime"/>
     <input name="capturetimeout"/>
     <input name="dbUuid"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>
```

```
<invoke activity="PrepareDBTesting"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspHandle" var="_bsphandle" />
     <input name="dbUuid" var="dbUuid" />
     <input name="nosourcepresentsupported"
     var="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"
     var="sourcepresenttimeouttime"/>
<output name="biruuid" setvar="uuid"/>
</invoke>
<!-- Вызвать функцию BioSPI_DbOpen для открытия базы
данных. -->
<invoke function="BioSPI_DbOpen">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >
<description>
     Функция BioSPI_DbOpen возвращает значение BioAPI_OK
     и выходной параметр dbHandle является достоверным
     дескриптором DB.
</description>
```

```
<equal_to var1="return" var2="__BioAPI_OK"/>
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbGetBIR с достоверными
значениями входных парметров. -->
<invoke function="BioSPI_DbGetBIR">
     <input name="BSPHandle" value="0"/>
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="uuid"/>
     <output name="RetrievedBIR"
     setvar="retrievedBir_handle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>

<description>
     Функция BioSPI_DbGetBIR возвращает значение
     BioAPIERR_INVALID_BSP_HANDLE.
</description>
<equal_to var1="return"
var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
```

```
            <return setvar="return"/>
        </invoke>
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
        break_if_false="true">

        <description>
            Функция BioSPI_DbClose возвращает значение
            BioAPI_OK.
        </description>

        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <invoke activity="CleanUpDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
            <input name="BSPHandle" var="_bsphandle" />
            <input name="dbUuid" var="dbUuid" />
        </invoke>
        <!-- Вызвать функции BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.73 Утверждение22c*BioSPI_DbGetBIR_RecordNotFound*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbGetBIR* с ключевым значением, не существующим в базе данных и проверки, является ли возвращенный код значением BioAPIERR_BSP_RECORD_NOT_FOUND.

Выдержки

Подпункт 9.3.5.8

BioAPI_RETURN BioAPI BioSPI_DbGetBIR
(BioAPI_HANDLE BSPHandle,
BioAPI_DB_HANDLE DbHandle,
*const BioAPI_UUID *KeyValue,*
*BioAPI_BIR_HANDLE *RetrievedBIR,*
*BioAPI_DB_MARKER_HANDLE *MarkerHandle).*

Подпункт 8.5.8.1

Восстанавливается ЗБИ, идентифицированная параметром KeyValue в открытой базе данных ЗБИ, идентифицированной параметром DbHandle. ЗБИ копируется в память ПБУ, и к ней возвращается дескриптор. Создается маркер и устанавливается на указатель на запись за восстановленной ЗБИ в базе данных ЗБИ (или на первую запись в базе данных, если восстановленная ЗБИ является последней), и к этому маркеру возвращается дескриптор.

Ссылки: 9.3.5.8 и 8.5.8.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных.

4) Вызвать функцию *BioSPI_Enroll* для создания ЗБИ.

5) Вызвать функцию *BioSPI_DbStoreBIR* для сохранения зарегистрированной ЗБИ в базе данных.

6) Вызвать функцию *BioSPI_DbGetBIR* с ключевым значением, не существующим в базе данных.

7) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_RECORD_NOT_FOUND.

8) Вызвать функцию *BioSPI_DbDelete* для удаления базы данных.

9) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbGetBIR* возвращает значение BioAPIERR_BSP_RECORD_NOT_FOUND.

Пакет языка утверждения

```
<package name="37457440-e8ba-11d9-87da-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>
     <description>
          Данный пакет содержит утверждение "BioSPI_DbGetBIR_ Re-
          cordNotFound"
</description>
<assertion name="BioSPI_DbGetBIR_RecordNotFound"
model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью вызова
               функции BioSPI_DbGetBIR с ключевым значением, не су-
               ществующим в базе данных и проверки, является ли
               возвращенный код значением BioAPIERR_BSP_RECORD_
               NOT_FOUND.
               Следующий текст соответствует приведенному в 8.5.8.1
               спецификации БиоАПИ 2.0.
               ------------------------------------------------
               BioAPI_RETURN BioAPI BioSPI_DbGetBIR
                    (BioAPI_HANDLE BSPHandle,
                    BioAPI_DB_HANDLE DbHandle,
                    const BioAPI_UUID *KeyValue,
                    BioAPI_BIR_HANDLE *RetrievedBIR,
                    BioAPI_DB_MARKER_HANDLE *MarkerHandle).
               Восстанавливается ЗБИ, идентифицированная параметром
               KeyValue в открытой базе данных ЗБИ, идентифициро-
               ванной параметром DbHandle. ЗБИ копируется в память
               ПБУ и к ней возвращается дескриптор. Создается
```

```
маркер и устанавливается на указатель на запись за
восстановленной ЗБИ в базе данных ЗБИ (или на первую
запись в базе данных, если восстановленная ЗБИ
является последней), и к этому маркеру возвращается
дескриптор.
Ошибки: BioAPIERR_RECORD_NOT_FOUND.
-----------------------------------------------------
Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.
2) Присоединить испытуемый ПБУ.
3) Вызвать функцию BioSPI_DbCreate для создания базы
данных.
4) Вызвать функцию BioSPI_Enroll для создания ЗБИ.
5) Вызвать функцию BioSPI_DbStoreBIR для сохранения
зарегистрированной ЗБИ в базе данных.
6) Вызвать функцию BioSPI_DbGetBIR с ключевым
значением, несуществующим в базе данных.
7) Проверить возвращаемый код. Предполагается
возвращение значения BioAPIERR_RECORD_NOT_FOUND.
8) Вызвать функцию BioSPI_DbDelete для удаления базы
данных.
9) Отсоединить и выгрузить ПБУ.
Если какие-либо промежуточные операции прошли
неуспешно, то выдается заключение о соответствии
"UNDECIDED".
</description>

<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>
<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />
```

```
<!-- Время ожидания для события
BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для захвата данных. -->
<input name="_capturetimeout"/>

<!-- База данных Uuid будет открыта. -->
<input name="_dbUuid"/>

<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbGetBIR">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeout" var="_capturetimeout"/>
     <input name="dbUuid" var="_dbUuid"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически.-->
<bind activity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>
<activity name="BioSPI_DbGetBIR">
```

```
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeout"/>
        <input name="dbUuid"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
</invoke>

<invoke activity="PrepareDBTesting"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
        <input name="bspHandle" var="_bsphandle" />
        <input name="dbUuid" var="dbUuid" />
        <input name="nosourcepresentsupported"
        var="nosourcepresentsupported"/>
        <input name="sourcepresenttimeouttime"
        var="sourcepresenttimeouttime"/>
        <output name="biruuid" setvar="uuid"/>
```

```
</invoke>
<!-- Вызвать функцию BioSPI_DbOpen для открытия базы
данных. -->
<invoke function="BioSPI_DbOpen">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >

<description>
     Функция BioSPI_DbOpen возвращает значение BioAPI_OK
     и выходной дескриптор dbHandle является достоверным
     дескриптором DB.
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>
<set name="nonExistingKey"
value="ffffffff-ffff-ffff-ffff-ffffffffffff">
<only_if>
     <not_same_as var1="uuid"
     value2="ffffffff-ffff-ffff-ffff-ffffffffffff"/>
</only_if>
</set>
```

```
<set name="nonExistingKey"
value="00000000-0000-0000-0000-000000000001">
<only_if>
     <same_as var1="uuid"
     value2="ffffffff-ffff-ffff-ffff-ffffffffffff"/>
</only_if>
</set>
<!-- Вызвать функцию BioSPI_DbGetBIR. -->
<invoke function="BioSPI_DbGetBIR">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="nonExistingKey"/>
     <output name="RetrievedBIR"
     setvar="retrievedBir_handle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>

<description>
     Функция BioSPI_DbGetBIR возвращает значение
     BioAPIERR_RECORD_NOT_FOUND.
</description>
<equal_to var1="return"
var2="__BioAPIERR_BSP_RECORD_NOT_FOUND "/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
```

```
            <input name="DbHandle" var="dbHandle"/>
            <return setvar="return"/>
        </invoke>
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
        break_if_false="true">

        <description>
            Функция BioSPI_DbClose возвращает значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <invoke activity="CleanUpDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
            <input name="BSPHandle" var="_bsphandle" />
            <input name="dbUuid" var="dbUuid" />
        </invoke>
        <!-- Вызвать функции BioSPI_ModuleDetach и BioS-
        PI_ModuleUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.74 Утверждение 23а *BioSPI_DbGetNextBIR_ValidParam*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbGetNextBIR* с достоверными значениями входных параметров и проверки, является возвращенный код значением BioAPI_OK или значением BioAPIERR_END_OF_DATABASE.

Выдержки

Подпункт 9.3.5.9

BioAPI_RETURN BioAPI BioSPI_DbGetNextBIR

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_HANDLE DbHandle,

BioAPI_DB_MARKER_HANDLE MarkerHandle,

*BioAPI_BIR_HANDLE *RetrievedBIR,*

*BioAPI_UUID *BirUuid).*

Подпункт 8.5.9.1

Восстанавливается ЗБИ, идентифицированная параметром MarkerHandle. ЗБИ копируется в память ПБУ и к ней возвращается дескриптор, а также указатель на УУИД, который уникально идентифицирует ЗБИ в базе данных. Маркер устанавливается на следующую запись в базе данных.

Ссылки: 9.3.5.9 и 8.5.9.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать процесс PrepareDBTesting для создания БД и сохранения в ней ЗБИ.

4) Вызвать функцию *BioSPI_Enroll* для создания ЗБИ.

5) Вызвать функцию *BioSPI_DbStoreBIR* для сохранения зарегистрированной ЗБИ в базе данных.

6) Вызвать функцию *BioSPI_DbSetMarker* для установки маркера на первую запись.

7) Вызвать функцию *BioSPI_DbGetNextBIR* для восстановления сохраненной ЗБИ.

8) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK или BioAPIERR_END_OF_DATABASE.

9) Вызвать функцию *BioSPI_DbGetNextBIR* для восстановления следующей ЗБИ; предполагается возвращение значения BioAPI_OK или BioAPIERR_END_OF_DATABASE.

10) Вызвать процесс CleanUpDBTesting для удаления базы данных.

11) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbGetNextBIR* возвращает значение BioAPI_OK или BioAPIERR_END_OF_DATABASE.

Пакет языка утверждения

```
<package name="e3396400-e8c9-11d9-990f-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>
     <description>
          Данный пакет содержит утверждение "BioSPI_DbGetNextBIR_
          ValidParam".
     </description>
     <assertion name="BioSPI_DbGetNextBIR_ValidParam"
     model="BSPTesting">
          <description>
               Проверку данного утверждения проводят с целью вызова
               функции BioSPI_DbGetNextBIR с достоверными значениями
               входных параметров и проверки, является возвращенный
               код значением BioAPI_OK или значением BioAPIERR_END_
               OF_DATABASE.
               Следующий текст соответствует приведенному в 8.5.9.1
               спецификации БиоАПИ 2.0.
               ------------------------------------------------
```

```
BioAPI_RETURN BioAPI BioSPI_DbGetNextBIR
     (BioAPI_HANDLE BSPHandle,
     BioAPI_DB_HANDLE DbHandle,
     BioAPI_DB_MARKER_HANDLE MarkerHandle,
     BioAPI_BIR_HANDLE *RetrievedBIR,
     BioAPI_UUID *BirUuid).
```

Восстанавливается ЗБИ, идентифицированная параметром MarkerHandle. ЗБИ копируется в память ПБУ, и к ней возвращается дескриптор, а также указатель на УУИД, который уникально идентифицирует ЗБИ в базе данных. Маркер устанавливается на следующую запись в базе данных.

Порядок действий при испытании на соответствие:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать процесс PrepareDBTesting для создания базы данных и сохранения в ней ЗБИ.

4) Вызвать функцию BioSPI_Enroll для создания ЗБИ.

5) Вызвать функцию BioSPI_DbStoreBIR для сохранения зарегистрированной ЗБИ в базе данных.

6) Вызвать функцию BioSPI_DbSetMarker для установки маркера на первую запись.

7) Вызвать функцию BioSPI_DbGetNextBIR для восстановления сохраненной ЗБИ.

8) Проверить возвращаемый код. Предполагается возвращение значения BioAPI_OK или BioAPIERR_END_OF_DATABASE.

9) Вызвать функцию BioSPI_DbGetNextBIR для восстановления следующей ЗБИ; предполагается возвращение значения BioAPI_OK или BioAPIERR_END_OF_DATABASE.

10) Вызвать процесс CleanUpDBTesting для удаления базы данных.

11) Отсоединить и выгрузить ПБУ.

```
        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>
    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>
    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>
    <!-- Указание на отсутствие поддержки испытуемым ПБУ
    уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
    <input name="_noSourcePresentSupported" />
    <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
    PRESENT. -->
    <input name="_sourcepresenttimeout"/>
    <!-- Время ожидания для захвата данных. -->
    <input name="_capturetimeout"/>
    <!-- База данных Uuid будет открыта. -->
    <input name="_dbUuid"/>
    <!--Вызов основного процесса данного утверждения со
    значениями входных параметров, определенными значениями
    параметров утверждения. -->
    <invoke activity="BioSPI_DbGetNextBIR">
        <input name="bspUuid" var="_bspUuid"/>
        <input name="inserttimeouttime"
        var="_inserttimeout"/>
        <input name="nosourcepresentsupported"
        var="_noSourcePresentSupported" />
        <input name="sourcepresenttimeouttime"
        var="_sourcepresenttimeout"/>
        <input name="capturetimeout" var="_capturetimeout"/>
        <input name="dbUuid" var="_dbUuid"/>
    </invoke>

    <!-- Процесс связан с функцией интерфейса обратного вызова
    инфраструктуры, предоставленной испытуемым компонентом ПБУ.
```

```
При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически.-->
<bind activity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>
<activity name="BioSPI_DbGetNextBIR">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported" />
     <input name="sourcepresenttimeouttime"/>
     <input name="capturetimeout"/>
     <input name="dbUuid"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<invoke activity="PrepareDBTesting"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
```

```
        <input name="bspHandle" var="_bsphandle" />
        <input name="dbUuid" var="dbUuid" />
        <input name="nosourcepresentsupported"
        var="nosourcepresentsupported"/>
        <input name="sourcepresenttimeouttime"
        var="sourcepresenttimeouttime"/>
        <output name="biruuid" setvar="biruuid"/>
</invoke>
<!-- Вызвать функцию BioSPI_DbOpen для открытия базы
данных. -->
<invoke function="BioSPI_DbOpen">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="DbUuid" var="dbUuid"/>
        <input name="ReadAccessRequest" value="true"/>
        <input name="WriteAccessRequest" value="true"/>
        <output name="DbHandle" setvar="dbHandle"/>
        <output name="MarkerHandle" setvar="markerHandle"/>
        <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS".  -->
<assert_condition response_if_false="undecided"
break_if_false="true" >

<description>
        Функция BioSPI_DbOpen возвращает значение BioAPI_OK
        и выходной дескриптор dbHandle является достоверным
        дескриптором DB.
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
```

```
</assert_condition>
<!-- Вызвать функцию BioSPI_DbSetMarker с достоверными
значениями входных пареметров. -->
<invoke function="BioSPI_DbSetMarker" >
     <input name="BSPHandle" var="_bsphandle" />
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="biruuid"/>
     <input name="MarkerHandle" var="markerHandle"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">

<description>
     Функция BioSPI_DbSetMarker возвращает значение
     BioAPI_OK.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbGetNextBIR с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbGetNextBIR">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="MarkerHandle" var="markerHandle"/>
     <output name="RetrievedBIR"
     setvar="retrievedBir_handle"/>
     <output name="BirUuid" setvar="biruuid1"/>
     <return setvar="return"/>
</invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
<description>
     Функция BioSPI_DbGetNextBIR возвращает значение
     BioAPI_OK или BioAPIERR_END_OF_DATABASE в зависимо-
     сти от команды, в которой сохранены ЗБИ и выходной
     УУИД, соответствующий указанному в шаблоне.
</description>
<and>
     <or>
          <equal_to var1="return" var2="__BioAPI_OK"/>
          <equal_to var1="return"
          var2="__BioAPIERR_BSP_END_OF_DATABASE"/>
     </or>
<same_as var1="biruuid" var2="biruuid1" />
</and>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
```

```
        <description>
            Функция BioSPI_DbClose возвращает значение BioAPI_OK.
        </description>

        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <invoke activity="CleanUpDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
            <input name="BSPHandle" var="_bsphandle" />
            <input name="dbUuid" var="dbUuid" />
        </invoke>
        <!-- Вызвать функции BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.75 Утверждение 23б *BioSPI_DbGetNextBIR_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbGetNextBIR* с недостоверным дескриптором ПБУ и проверки, является возвращенный код значением BioAPIERR_INVALID_BSP_HANDLE.

Выдержки

Подпункт 9.3.5.9

BioAPI_RETURN BioAPI BioSPI_DbGetNextBIR

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_HANDLE DbHandle,

BioAPI_DB_MARKER_HANDLE MarkerHandle,

*BioAPI_BIR_HANDLE *RetrievedBIR,*

*BioAPI_UUID *BirUuid).*

Подпункт 8.5.9.1

Восстанавливается ЗБИ, идентифицированная параметром MarkerHandle. ЗБИ копируется в память ПБУ и к ней возвращается дескриптор, а также указатель на УУИД, который уникально идентифицирует ЗБИ в базе данных. Маркер устанавливается на следующую запись в базе данных.

Ссылки: 9.3.5.9 и 8.5.9.1

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных.

4) Вызвать функцию *BioSPI_Enroll* для создания ЗБИ.

5) Вызвать функцию *BioSPI_DbStoreBIR* для сохранения зарегистрированной ЗБИ в базе данных.

6) Вызвать функцию *BioSPI_DbSetMarker* для установки маркера на первую запись.

7) Вызвать функцию *BioSPI_DbGetNextBIR* с недостоверным дескриптором ПБУ.

8) Проверить возвращаемый код. Предполагается возвращение значения BioAPIERR_INVALID_BSP_HANDLE.

9) Вызвать функцию *BioSPI_DbDelete* для удаления базы данных.

10) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbGetNextBIR* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="f0ef5320-e8d3-11d9-bbc1-0002a5d5c51b">
    <author>
```

```
     ISO/IEC JTC1 SC37
</author>
<description>
     Данный пакет содержит утверждение
     "BioSPI_DbGetNextBIR_InvalidBSPHandle".
</description>
<assertion name="BioSPI_DbGetNextBIR_InvalidBSPHandle"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью вызова
          функции BioSPI_DbGetNextBIR с недостоверным дескрип-
          тором ПБУ и проверки, является возвращенный код зна-
          чением BioAPIERR_INVALID_BSP_HANDLE.
          Приведенный ниже текст соответствует приведенному в
          8.5.9.1 спецификации БиоАПИ 2.0.
          ------------------------------------------------
          BioAPI_RETURN BioAPI BioSPI_DbGetNextBIR
               (BioAPI_HANDLE BSPHandle,
               BioAPI_DB_HANDLE DbHandle,
               BioAPI_DB_MARKER_HANDLE MarkerHandle,
               BioAPI_BIR_HANDLE *RetrievedBIR,
               BioAPI_UUID *BirUuid).
          Восстанавливается ЗБИ, идентифицированная параметром
          MarkerHandle. ЗБИ копируется в память ПБУ, и к ней
          возвращается дескриптор, а также указатель на УУИД,
          который уникально идентифицирует ЗБИ в базе данных.
          Маркер устанавливается на следующую запись в базе
          данных.
          ------------------------------------------------
          8.5.9.2 Параметры BSPHandle (входной параметр) - де-
          скриптор присоединенного ПБУ.
          ------------------------------------------------
          Порядок действий при испытании на соответствие:

          1) Загрузить испытуемый ПБУ.
          2) Присоединить испытуемый ПБУ.
```

```
            3) Вызвать функцию BioSPI_DbCreate для создания базы
            данных.
            4) Вызвать функцию BioSPI_Enroll для создания ЗБИ.
            5) Вызвать функцию BioSPI_DbStoreBIR для сохранения
            зарегистрированной ЗБИ в базе данных.
            6) Вызвать функцию BioSPI_DbSetMarker для установки
            маркера на первую запись.
            7) Вызвать функцию BioSPI_DbGetNextBIR с недостоверным
            дескриптором ПБУ.
            8) Проверить возвращаемый код. Предполагается
            возвращение значения BioAPIERR_INVALID_BSP_HANDLE.
            9) Вызвать функцию BioSPI_DbDelete для удаления базы
            данных.
            10) Отсоединить и выгрузить ПБУ.
        </description>
        <!-- УУИД испытуемого ПБУ. -->
        <input name="_bspUuid"/>

        <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
        <input name="_inserttimeout"/>

        <!-- Указание на отсутствие поддержки испытуемым ПБУ
        уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
        <input name="_noSourcePresentSupported" />

        <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
        PRESENT. -->
        <input name="_sourcepresenttimeout"/>
        <!-- Время ожидания для захвата данных -->
        <input name="_capturetimeout"/>

        <!-- База данных Uuid будет открыта. -->
        <input name="_dbUuid"/>
```

```
<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbGetNextBIR">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeout" var="_capturetimeout"/>
     <input name="dbUuid" var="_dbUuid"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически.-->
<bind activity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>
<activity name="BioSPI_DbGetNextBIR">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported" />
     <input name="sourcepresenttimeouttime"/>
     <input name="capturetimeout"/>
     <input name="dbUuid"/>
<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
```

```
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>

<invoke activity="PrepareDBTesting"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspHandle" var="_bsphandle" />
     <input name="dbUuid" var="dbUuid" />
     <input name="nosourcepresentsupported"
     var="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"
     var="sourcepresenttimeouttime"/>
     <output name="biruuid" setvar="uuid"/>
</invoke>

<!-- Вызвать функцию BioSPI_DbOpen для открытия базы
данных. -->
<invoke function="BioSPI_DbOpen">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
```

```
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >

<description>
     Функция BioSPI_DbOpenвозвращает значение BioAPI_OK и
     выходной дескриптор dbHandle, является достоверным
     дескриптором DB.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbSetMarker с достоверными
значениями входных параметров -->
<invoke function="BioSPI_DbSetMarker" >
     <input name="BSPHandle" var="_bsphandle" />
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="uuid"/>
     <input name="MarkerHandle" var="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии'PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
```

```
<description>
     Функция BioSPI_DbSetMarker возвращает значение
     BioAPI_OK.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbGetNextBIR с достоверными
значениями входных парметров. -->
<invoke function="BioSPI_DbGetNextBIR">
     <input name="BSPHandle" value="0"/>
     <input name="MarkerHandle" var="markerHandle"/>
     <output name="RetrievedBIR" set-
     var="retrievedBir_handle"/>
     <output name="BirUuid" setvar="biruuid"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
<description>
     Функция BioSPI_DbGetNextBIR возвращает значение Bio-
     APIERR_INVALID_BSP_HANDLE.
</description>
<equal_to var1="return"
var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE "/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
```

```
            <input name="DbHandle" var="dbHandle"/>
            <return setvar="return"/>
        </invoke>

        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
        break_if_false="true">
        <description>
            Функция BioSPI_DbClose возвращает значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <invoke activity="CleanUpDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
            <input name="BSPHandle" var="_bsphandle" />
            <input name="dbUuid" var="dbUuid" />
        </invoke>
        <!-- Вызвать функции BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
            <input name="bspUuid" var="bspUuid" />
            <input name="BSPHandle" var="_bsphandle" />
        </invoke>
    </activity>
</package>
```

8.76 Утверждение 24а *BioSPI_DbDeleteBIR_ValidParam*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbGetNextBIR* с достоверными значениями параметров и проверки, является возвращенный код значением BioAPI_OK.

Выдержки

Подпункт 9.3.5.10

BioAPI_RETURN BioAPI BioSPI_DbDeleteBIR

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_HANDLE DbHandle,

*const BioAPI_UUID *KeyValue).*

Подпункт 8.5.10.1

ЗБИ, идентифицированная параметром KeyValue в открытой базе данных ЗБИ, идентифицированной параметром DbHandle, удаляется из базы данных.

Ссылки: 9.3.5.10 и 8.5.10.1.

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных.

4) Вызвать функцию *BioSPI_Enroll* для создания ЗБИ.

5) Вызвать функцию *BioSPI_DbStoreBIR* для сохранения полученной ЗБИ в базе данных.

6) Вызвать функцию *BioSPI_DbDeleteBIR* для удаления сохраненной ЗБИ.

7) Вызвать функцию *BioSPI_DbGetBIR* для восстановления удаленной ЗБИ. Предполагается возвращение значения BioAPIERR_RECORD_NOT_FOUND.

8) Вызвать функцию *BioSPI_DbDelete* для удаления базы данных.

9) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции BioSPI_DbGetBIR возвращает значение BioAPIERR_RECORD_NOT_FOUND.

Пакет языка утверждения

```
<package name="ed4afbe0-e8d6-11d9-9ed0-0002a5d5c51b">
    <author>
        ISO/IEC JTC1 SC37
    </author>
    <description>
        Данный пакет содержит утверждение "BioSPI_DbDeleteBIR_
        ValidParam".
    </description>
    <assertion name="BioSPI_DbDeleteBIR_ValidParam"
    model="BSPTesting">
        <description>
            Проверку данного утверждения проводят с целью вызова
            функции BioSPI_DbGetNextBIR с недостоверным дескрип-
            тором ПБУ и проверки, является возвращенный код зна-
            чением BioAPIERR_INVALID_BSP_HANDLE.
            Приведенный ниже текст соответствует приведенному в
            8.5.10.1 спецификации БиоАПИ 2.0.
            ------------------------------------------------
            BioAPI_RETURN BioAPI BioAPI_DbDeleteBIR
            (BioAPI_HANDLE BSPHandle,
            BioAPI_DB_HANDLE DbHandle,
            constBioAPI_UUID *KeyValue).

            ЗБИ, идентифицированная параметром KeyValue в открытой
            базе данных ЗБИ, идентифицированной параметром
            DbHandle, удаляется из базы данных.
            ------------------------------------------------
            Порядок действий при испытании на соответствие:
            1) Загрузить испытуемый ПБУ.
            2) Присоединить испытуемый ПБУ.
            3) Вызвать функцию BioSPI_DbCreate для создания базы
            данных.
            4) Вызвать функцию BioSPI_Enroll для создания ЗБИ.
```

```
        5) Вызвать функцию BioSPI_DbStoreBIR для сохранения
        полученной ЗБИ в базе данных.
        6) Вызвать функцию BioSPI_DbDeleteBIR для удаления
        сохраненной ЗБИ.
        7) Вызвть функцию BioSPI_DbGetBIR для восстановления
        удаленной ЗБИ. Предполагается возвращение значения
        BioAPIERR_RECORD_NOT_FOUND.
        8) Вызвать функцию BioSPI_DbDelete для удаления базы
        данных.
        9) Отсоединить и выгрузить ПБУ.
        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
</description>
<!-- УУИД испытуемого ПБУ. -->
<input name="_bspUuid"/>

<!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
<input name="_inserttimeout"/>

<!-- Указание на отсутствие поддержки испытуемым ПБУ
уведомления о событии BioAPI_NOTIFY_SOURCE_PRESENT. -->
<input name="_noSourcePresentSupported" />

<!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
PRESENT. -->
<input name="_sourcepresenttimeout"/>

<!-- Время ожидания для захвата данных. -->
<input name="_capturetimeout"/>

<!—- Открытие базы данных УУИД. -->
<input name="_dbUuid"/>
```

```
<!-- Вызов основного процесса данного утверждения со
значениями входных параметров, определенными значениями
параметров утверждения. -->
<invoke activity="BioSPI_DbDeleteBIR">
     <input name="bspUuid" var="_bspUuid"/>
     <input name="inserttimeouttime"
     var="_inserttimeout"/>
     <input name="nosourcepresentsupported"
     var="_noSourcePresentSupported" />
     <input name="sourcepresenttimeouttime"
     var="_sourcepresenttimeout"/>
     <input name="capturetimeout" var="_capturetimeout"/>
     <input name="dbUuid" var="_dbUuid"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически.-->
<bind activity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>
<activity name="BioSPI_DbDeleteBIR">
     <input name="bspUuid"/>
     <input name="inserttimeouttime"/>
     <input name="nosourcepresentsupported" />
     <input name="sourcepresenttimeouttime"/>
     <input name="capturetimeout"/>
     <input name="dbUuid"/>

<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>
<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
```

```
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspUuid" var="bspUuid"/>
     <input name="bspVersion" value="32"/>
     <input name="unitIDOrNull" value="0"/>
     <input name="bspHandle" var="_bsphandle"/>
     <input name="eventtimeouttime"
     var="inserttimeouttime"/>
</invoke>
<invoke activity="PrepareDBTesting"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
     <input name="bspHandle" var="_bsphandle" />
     <input name="dbUuid" var="dbUuid" />
     <input name="nosourcepresentsupported"
     var="nosourcepresentsupported"/>
     <input name="sourcepresenttimeouttime"
     var="sourcepresenttimeouttime"/>
     <output name="biruuid" setvar="biruuid"/>
</invoke>
<!-- Вызвать функцию BioSPI_DbOpen для открытия базы
данных. -->
<invoke function="BioSPI_DbOpen">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbUuid" var="dbUuid"/>
     <input name="ReadAccessRequest" value="true"/>
     <input name="WriteAccessRequest" value="true"/>
     <output name="DbHandle" setvar="dbHandle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
</invoke>
```

```
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается заклю-
чение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true" >
<description>
     Функция BioSPI_DbOpen возвращает значение BioAPI_OK
     и выходной дескриптор dbHandle является достоверным
     дескриптором базы данных.
</description>
<equal_to var1="return" var2="__BioAPI_OK"/>
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>

<!-- Вызвать функцию BioSPI_DbDeleteBIR с достоверными
значениями входных параметров. -->
     <invoke function="BioSPI_DbDeleteBIR">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="biruuid"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>
<description>
     Функция BioSPI_DbDeleteBIR возвращает значение BioAPI_OK.
</description>
```

```
<equal_to var1="return" var2="__BioAPI_OK"/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbGetBIR с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbGetBIR">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="biruuid"/>
     <output name="RetrievedBIR" set-
     var="retrievedBir_handle"/>
     <output name="MarkerHandle" setvar="markerHandle"/>
     <return setvar="return"/>
</invoke>
<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "UNDECIDED" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition response_if_false="undecided"
break_if_false="true">
<description>
     Функция BioSPI_DbGetBIR возвращает значение Bio-
     APIERR_RECORD_NOT_FOUND.
</description>
<equal_to var1="return"
var2="__BioAPIERR_BSP_RECORD_NOT_FOUND "/>
</assert_condition>
<!-- Вызвать функцию BioSPI_DbClose с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbClose">
     <input name="BSPHandle" var="_bsphandle"/>
     <input name="DbHandle" var="dbHandle"/>
     <return setvar="return"/>
</invoke>
```

```
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "UNDECIDED" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
        break_if_false="true">
        <description>
             Функция BioSPI_DbClose возвращает значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <invoke activity="CleanUpDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
             <input name="BSPHandle" var="_bsphandle" />
             <input name="dbUuid" var="dbUuid" />
             </invoke>
        <!-- Вызвать функции BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
             <input name="bspUuid" var="bspUuid" />
             <input name="BSPHandle" var="_bsphandle" />
        </invoke>
</activity>
</package>
```

8.77 Утверждение 24b *BioSPI_DbDeleteBIR_InvalidBSPHandle*

Описание: Проверку данного утверждения проводят с целью вызова функции *BioSPI_DbDeleteBIR* с недостоверным дескриптором ПБУ и проверки, является ли возвращенный код значением BioAPIERR_INVALID_BSP_HANDLE.

Выдержки

Подпункт 9.3.5.10

BioAPI_RETURN BioAPI BioSPI_DbDeleteBIR

(BioAPI_HANDLE BSPHandle,

BioAPI_DB_HANDLE DbHandle,

*const BioAPI_UUID *KeyValue).*

Подпункт 8.5.10.1

ЗБИ, идентифицированная параметром KeyValue в открытой базе данных ЗБИ, идентифицированной параметром DbHandle, удаляется из базы данных.

Ссылки: 9.3.5.10 и 8.5.10.1

Порядок действий:

1) Загрузить испытуемый ПБУ.

2) Присоединить испытуемый ПБУ.

3) Вызвать функцию *BioSPI_DbCreate* для создания базы данных.

4) Вызвать функцию *BioSPI_Enroll* для создания ЗБИ.

5) Вызвать функцию *BioSPI_DbStoreBIR* для сохранения полученной ЗБИ в базе данных.

6) Вызвать функцию *BioSPI_DbDeleteBIR* для удаления сохраненной ЗБИ.

7) Вызвать функцию *BioSPI_DbDelete* для удаления базы данных.

8) Отсоединить и выгрузить ПБУ.

Если какие-либо промежуточные операции прошли неуспешно, то выдается заключение о соответствии "UNDECIDED".

Ожидаемые результаты: Вызов функции *BioSPI_DbDeleteBIR* возвращает значение BioAPIERR_INVALID_BSP_HANDLE.

Пакет языка утверждения

```
<package name="72eca940-e8d9-11d9-aa0d-0002a5d5c51b">
     <author>
          ISO/IEC JTC1 SC37
     </author>
     <description>
          Данный пакет содержит утверждение "BioSPI_DbDeleteBIR_
          InvalidBSPHandle" (см. далее элемент <description>
          утверждения).
```

```
</description>
<assertion name="BioSPI_DbDeleteBIR_InvalidBSPHandle"
model="BSPTesting">
     <description>
          Проверку данного утверждения проводят с целью вызова
          функции BioSPI_DbDeleteBIR с недостоверным дескриптором
          ПБУ и проверки, является ли возвращенный код значением
          BioAPIERR_INVALID_BSP_HANDLE.
          Приведенный ниже текст соответствует приведенному в
          8.5.10.1 спецификации БиоАПИ 2.0.
          ---------------------------------------------------
          BioAPI_RETURN BioAPI BioAPI_DbDeleteBIR
          (BioAPI_HANDLE BSPHandle,
          BioAPI_DB_HANDLE DbHandle,
          constBioAPI_UUID *KeyValue).
          ЗБИ, идентифицированная параметром KeyValue в открытой
          базе данных ЗБИ, идентифицированной параметром
          DbHandle, удаляется из базы данных.
          ---------------------------------------------------
          8.5.10.2 Параметр BSPHandle (входной параметр) -
          дескриптор присоединенного ПБУ.
          ---------------------------------------------------
          Порядок действий при испытании на соответствие:
          1) Загрузить испытуемый ПБУ.
          2) Присоединить испытуемый ПБУ.
          3) Вызвать функцию BioSPI_DbCreate для создания базы
          данных.
          4) Вызвать функцию BioSPI_Enroll для создания ЗБИ.
          5) Вызвать функцию BioSPI_DbStoreBIR для сохранения
          полученной ЗБИ в базе данных.
          6) Вызвать функцию BioSPI_DbDeleteBIR для удаления
          сохраненной ЗБИ.
          7) Вызвать функцию BioSPI_DbDelete для удаления базы
          данных.
          8) Отсоединить и выгрузить ПБУ.
```

```
        Если какие-либо промежуточные операции прошли
        неуспешно, то выдается заключение о соответствии
        "UNDECIDED".
    </description>

    <!-- УУИД испытуемого ПБУ. -->
    <input name="_bspUuid"/>

    <!-- Время ожидания для события BioAPI_NOTIFY_INSERT. -->
    <input name="_inserttimeout"/>

    <!-- Вызов основного процесса данного утверждения со
    значениями входных параметров, определенными значениями
    параметров утверждения. -->
    <input name="_noSourcePresentSupported" />

    <!-- Время ожидания для события BioAPI_NOTIFY_SOURCE_
    PRESENT. -->
    <input name="_sourcepresenttimeout"/>

    <!-- Время ожидания для захвата данных. -->
    <input name="_capturetimeout"/>

    <!-- Открыть базу данных УУИД. -->
    <input name="_dbUuid"/>

    <!-- Вызов главного действия данного утверждения со
    значениями входных параметров, определенных параметрами
    утверждения. -->
    <invoke activity="BioSPI_DbDeleteBIR">
        <input name="bspUuid" var="_bspUuid"/>
        <input name="inserttimeouttime"
        var="_inserttimeout"/>
        <input name="nosourcepresentsupported"
        var="_noSourcePresentSupported" />
```

```
        <input name="sourcepresenttimeouttime"
        var="_sourcepresenttimeout"/>
        <input name="capturetimeout" var="_capturetimeout"/>
        <input name="dbUuid" var="_dbUuid"/>
</invoke>

<!-- Процесс связан с функцией интерфейса обратного вызова
инфраструктуры, предоставленной испытуемым компонентом
ПБУ. При каждом входящем вызове функции связанный с ней
процесс будет вызываться автоматически. -->
<bind activity="EventHandler"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
function="BioSPI_EventHandler"/>
</assertion>
<activity name="BioSPI_DbDeleteBIR">
        <input name="bspUuid"/>
        <input name="inserttimeouttime"/>
        <input name="nosourcepresentsupported" />
        <input name="sourcepresenttimeouttime"/>
        <input name="capturetimeout"/>
        <input name="dbUuid"/>
<!-- Данное утверждение использует BSPHandle со значением
"1" для всех вызовов BioSPI, которые этого требуют. -->
<set name="_bsphandle" value="1"/>

<!-- Вызов функций BioSPI_BSPLoad и BioSPI_BSPAttach,
предоставленных испытуемым ПБУ. Входное значение параметра
"unitIDOrNull" равно "0", поэтому утверждение будет
тестировать модуль датчика, выбранный ПБУ. -->
<invoke activity="LoadAndAttach"
package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
break_on_break="true">
        <input name="bspUuid" var="bspUuid"/>
        <input name="bspVersion" value="32"/>
        <input name="unitIDOrNull" value="0"/>
        <input name="bspHandle" var="_bsphandle"/>
```

```
        <input name="eventtimeouttime"
        var="inserttimeouttime"/>
    </invoke>

    <invoke activity="PrepareDBTesting"
    package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
    break_on_break="true">
        <input name="bspHandle" var="_bsphandle" />
        <input name="dbUuid" var="dbUuid" />
        <input name="nosourcepresentsupported"
        var="nosourcepresentsupported"/>
        <input name="sourcepresenttimeouttime"
        var="sourcepresenttimeouttime"/>
        <output name="biruuid" setvar="biruuid"/>
    </invoke>

    <!-- Вызов функции BioSPI_DbOpen для открытия базы
    данных. -->
    <invoke function="BioSPI_DbOpen">
        <input name="BSPHandle" var="_bsphandle"/>
        <input name="DbUuid" var="dbUuid"/>
        <input name="ReadAccessRequest" value="true"/>
        <input name="WriteAccessRequest" value="true"/>
        <output name="DbHandle" setvar="dbHandle"/>
        <output name="MarkerHandle" setvar="markerHandle"/>
        <return setvar="return"/>
    </invoke>

    <!-- Выдача заключения о соответствии. Если условие,
    определенное далее в элементе <description>, ложное,
    выдается заключение о соответствии "UNDECIDED" и выполнение
    процесса прерывается, в противном случае выдается
    заключение о соответствии "PASS". -->
    <assert_condition response_if_false="undecided"
    break_if_false="true" >
```

```
<description>
     Функция BioSPI_DbOpen возвращает значение BioAPI_OK
     и выходной дескриптор dbHandle является достоверным
     дескриптором DB.
</description>

<equal_to var1="return" var2="__BioAPI_OK"/>
<not_equal_to var1="dbHandle"
var2="__BioAPI_DB_INVALID_HANDLE"/>
</assert_condition>

<!-- Вызов функции BioSPI_DbDeleteBIR с достоверными
значениями входных параметров. -->
<invoke function="BioSPI_DbDeleteBIR">
     <input name="BSPHandle" value="0"/>
     <input name="DbHandle" var="dbHandle"/>
     <input name="KeyValue" var="biruuid"/>
     <return setvar="return"/>
</invoke>

<!-- Выдача заключения о соответствии. Если условие,
определенное далее в элементе <description>, ложное,
выдается заключение о соответствии "FAIL" и выполнение
процесса прерывается, в противном случае выдается
заключение о соответствии "PASS". -->
<assert_condition>

<description>
     Функция BioSPI_DbDeleteBIR возвращает значение
     BioAPIERR_INVALID_BSP_HANDLE.
</description>

<equal_to var1="return"
var2="__BioAPIERR_BSP_INVALID_BSP_HANDLE "/>
</assert_condition>
```

```
        <!-- Вызов функции BioSPI_DbClose с достоверными
        значениями входных параметров. -->
        <invoke function="BioSPI_DbClose">
             <input name="BSPHandle" var="_bsphandle"/>
             <input name="DbHandle" var="dbHandle"/>
             <return setvar="return"/>
             </invoke>
        <!-- Выдача заключения о соответствии. Если условие,
        определенное далее в элементе <description>, ложное,
        выдается заключение о соответствии "FAIL" и выполнение
        процесса прерывается, в противном случае выдается
        заключение о соответствии "PASS". -->
        <assert_condition response_if_false="undecided"
        break_if_false="true">
        <description>
             Функция BioSPI_DbClose возвращает значение BioAPI_OK.
        </description>
        <equal_to var1="return" var2="__BioAPI_OK"/>
        </assert_condition>
        <invoke activity="CleanUpDBTesting"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e"
        break_on_break="true">
             <input name="BSPHandle" var="_bsphandle" />
             <input name="dbUuid" var="dbUuid" />
        </invoke>
        <!-- Вызвать функции BioSPI_BSPDetach и BioSPI_BSPUnload. -->
        <invoke activity="DetachAndUnload"
        package="02c59458-0c46-1085-95d7-0002a5d5fd2e" >
             <input name="bspUuid" var="bspUuid" />
             <input name="BSPHandle" var="_bsphandle" />
        </invoke>
</activity>
</package>
```

Приложение ДА

(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации

Сведения о состоянии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации приведены в таблице ДА.1.

Таблица ДА.1.

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего межгосударственного стандарта
ИСО/МЭК 19784-1:2006	IDT	ГОСТ Р ИСО/МЭК 19784-1–2007 «Автоматическая идентификация. Идентификация биометрическая. Биометрический программный интерфейс. Часть 1. Спецификация биометрического программного интерфейса»
ИСО/МЭК 24709-1:2007	IDT	ГОСТ Р 24709-1–2009 «Автоматическая идентификация. Идентификация биометрическая. Испытания на соответствие биометрическому программному интерфейсу (БиоАПИ). Часть 1. Методы и процедуры»

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

- IDT – идентичный стандарт.

УДК 004.93'1:006.89:006.354 ОКС 35.040 П85

Ключевые слова: автоматическая идентификация, биометрическая идентификация, испытания на соответствие биометрическому программному интерфейсу, тестовые утверждения

Подписано в печать 30.04.2014. Формат 60x84 1/8.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

ФГУП «СТАНДАРТИНФОРМ»

123995 Москва, Гранатный пер., 4.

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

gost.e-ecolog.ru