ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 10373-6— 2015

Карты идентификационные

МЕТОДЫ ИСПЫТАНИЙ

Часть 6

Карты близкого действия

ISO/IEC 10373-6:2011
Identification cards — Test methods — Part 6: Proximity cards
(IDT)

Издание официальное

Москва
Стандартинформ
2015

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Федеральным государственным унитарным предприятием «Всероссийский научно-исследовательский институт стандартизации и сертификации в машиностроении» (ВНИИНМАШ) и Техническим комитетом по стандартизации ТК 22 «Информационные технологии» на основе собственного аутентичного перевода на русский язык стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 22 «Информационные технологии»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 22 июня 2015 г. № 773-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 10373-6:2011 «Карты идентификационные. Методы испытаний. Часть 6. Карты близкого действия» (ISO/IEC10373-6:2011 Identification cards — Test methods — Part 6: Proximity cards), включая изменения A1:2012, A2:2012, A3:2012, A4:2012 и поправку Cor.1:2013.

Изменения и техническая поправка к указанному международному стандарту, принятые после его официальной публикации, внесены в текст настоящего стандарта и выделены двойной вертикальной линией, расположенной на полях напротив соответствующего текста, а обозначение и год принятия изменения (технической поправки) приведены в скобках после соответствующего текста.

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

Некоторые положения международного стандарта, указанного в пункте 4, могут являться объектом патентных прав. Международная организация по стандартизации (ИСО) и Международная электротехническая комиссия (МЭК) не несут ответственности за идентификацию подобных патентных прав

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в ГОСТ 1.0—2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.gost.ru)

© Стандартинформ, 2015

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

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины, определения, обозначения и сокращения . . . 2
3.1 Термины и определения . . . 2
3.2 Обозначения и сокращения . . . 2
4 Нормальные условия испытаний . . . 5
4.1 Нормальные климатические условия . . . 5
4.2 Предварительное кондиционирование . . . 5
4.3 Допускаемые отклонения . . . 5
4.4 Паразитная индуктивность . . . 5
4.5 Суммарная погрешность измерений . . . 5
5 Оборудование и схемы для испытаний по параметрам, установленным в ИСО/МЭК 14443-1 и ИСО/МЭК 14443-2 . . . 6
5.1 Минимальные требования к средствам измерений . . . 6
5.2 Калибровочная катушка . . . 6
5.3 Испытательная установка PCD . . . 7
5.4 Эталонные PICC . . . 9
5.5 Испытательная установка EMD . . . 12
6 Испытания по параметрам, установленным в ИСО/МЭК 14443-1 . . . 13
6.1 Испытания PCD . . . 13
6.2 Испытания PICC . . . 14
7 Испытания по параметрам, установленным в ИСО/МЭК 14443-2 . . . 16
7.1 Испытания PCD . . . 16
7.2 Испытания PICC . . . 21
8 Испытание по параметрам, установленным в ИСО/МЭК 14443-3 и ИСО/МЭК 14443-4 . . . 25
8.1 Испытания PCD . . . 25
8.2 Испытания PICC . . . 25
Приложение A (обязательное) Испытательная антенна PCD 1 . . . 26
Приложение B (справочное) Настройка испытательной антенны PCD . . . 34
Приложение C (обязательное) Катушки считывания . . . 36
Приложение D (обязательное) Эталонные PICC . . . 39
Приложение E (обязательное) Инструментарий для анализа формы сигнала и определения коэффициента модуляции . . . 45
Приложение F (справочное) Программа для вычисления спектра . . . 93
Приложение G (обязательное) Дополнительные методы испытаний PICC . . . 99
Приложение H (обязательное) Дополнительные методы испытаний PCD . . . 151
Приложение I (обязательное) Методы испытания PCD для выбора высокой скорости передачи . . . 180
Приложение J (справочное) Программа для измерения уровня EMD . . . 193
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации и межгосударственным стандартам . . . 202
Библиография . . . 203

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Карты идентификационные

МЕТОДЫ ИСПЫТАНИЙ

Часть 6

Карты близкого действия

Identification cards. Test methods. Part 6. Proximity cards

Дата введения — 2016 —07—01

1 Область применения

Стандарты серии ИСО/МЭК 10373 устанавливают методы испытаний для определения характеристик идентификационных карт, соответствующих определению, приведенному в ИСО/МЭК 7810 (далее — карты). В каждом методе испытания имеется ссылка на один или несколько базовых стандартов, которыми могут быть ИСО/МЭК 7810 либо один или несколько дополнительных стандартов, устанавливающих требования к технологиям хранения информации, применяемым в картах.

Примечание 1 — Критерии оценки результатов испытаний не включены в настоящий стандарт; они установлены в базовых стандартах.

Примечание 2 — Испытания, определенные в настоящем стандарте, следует проводить независимо одно от другого. Любую конкретную карту близкого действия или терминальное оборудование близкого действия не следует подвергать последовательно всем испытаниям.

Настоящий стандарт устанавливает методы испытаний для карт и объектов близкого действия и соответствующего терминального оборудования, требования к которым установлены в ИСО/МЭК 14443-1:2008, ИСО/МЭК 14443-2:2010, ИСО/МЭК 14443-3:2011 и ИСО/МЭК 14443-4:2008. ИСО/МЭК 10373-1 определяет методы испытаний, являющиеся общими для одной или нескольких технологий хранения информации в картах на интегральных схемах, а остальные стандарты серии ИСО/МЭК 10373 устанавливают другие методы испытаний, предназначенные для других технологий.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие международные стандарты. Для датированных ссылок следует использовать только указанное издание, для недатированных ссылок следует использовать последнее издание указанного документа, включая все поправки:

ИСО/МЭК 7810:2003 Карты идентификационные. Физические характеристики (ISO/IEC 7810:2003 Identification cards — Physical characteristics)

ИСО/МЭК 14443-1:2008 Карты идентификационные. Карты на интегральных схемах бесконтактные. Карты близкого действия. Часть 1. Физические характеристики (ISO/IEC 14443-1:2008 Identification cards — Contactless integrated circuit cards — Proximity cards — Part 1: Physical characteristics)

ИСО/МЭК 14443-2:2010 Карты идентификационные. Карты на интегральных схемах бесконтактные. Карты близкого действия. Часть 2. Радиочастотный энергетический и сигнальный интерфейс (ISO/IEC 14443-2:2010 Identification cards — Contactless integrated circuit cards — Proximity cards — Part 2: Radio frequency power and signal interface)

ИСО/МЭК 14443-3:2011 Карты идентификационные. Карты на интегральных схемах бесконтактные. Карты близкого действия. Часть 3. Инициализация и антиколлизия (ISO/IEC 14443-3:2011 Identification cards — Contactless integrated circuit cards — Proximity cards — Part 3: Initialization and anticollision)

Издание официальное

ИСО/МЭК 14443-4:2008 Карты идентификационные. Карты на интегральных схемах бесконтактные. Карты близкого действия. Часть 4. Протокол передачи (ISO/IEC 14443-4:2008 Identification cards — Contactless integrated circuit cards — Proximity cards — Part 4: Transmission protocol)

МЭК 61000-4-2:2008 Электромагнитная совместимость. Часть 4-2. Методики испытаний и измерений. Испытание на невосприимчивость к электростатическому разряду [IEC 61000-4-2:2008 Electromagnetic compatibility (EMC) — Part 4-2: Testing and measurement techniques — Electrostatic discharge immunity test]

3 Термины, определения, обозначения и сокращения

В настоящем стандарте применены термины, определения, сокращения и обозначения по ИСО/МЭК 14443-1:2008, ИСО/МЭК 14443-2:2010, ИСО/МЭК 14443-3:2011, ИСО/МЭК 14443-4:2008, а также следующие термины с соответствующими определениями.

Примечание — Элементы в жирных скобках **[ ]** являются дополнительными определениями.

3.1 Термины и определения

3.1.1 **базовый стандарт** (base standard): Стандарт, для проверки соответствия требованиям которого используют данный метод испытания.

3.1.2 **CascadeLevels**: Число каскадных уровней PICC.

3.1.3 **PICC «Класс 1»** («Class 1» PICC): PICC, у которой антенна расположена согласно ИСО/МЭК 14443-1:2008 и которая выдерживает испытание на максимальный эффект нагрузки по 7.2.5 настоящего стандарта в условиях, соответствующих PICC «Класс 1».

3.1.4 **набор команд** (command set): Набор, описывающий команды, посылаемые PICC во время инициализации и антиколлизии.

Примечание — См. ИСО/МЭК 14443-3:2011 (подраздел 6.4) для PICC типа A и ИСО/МЭК 14443-3:2011 (подраздел 7.5) для PICC типа B.

3.1.5 **эффект нагрузки** (loading effect): Изменение тока антенны PCD, вызванное присутствием PICC в поле, в результате взаимного изменения резонанса и добротности антенны PCD.

(Введен дополнительно. Изм. А1:2012.)

3.1.6[1] **отключение** (mute): Отсутствие ответа в течение указанного времени ожидания, например, по истечению FWT.

3.1.7 **состояния PICC** (PICC states): Различные состояния PICC во время инициализации и антиколлизии.

Примечание — См. ИСО/МЭК 14443-3:2011 (подраздел 6.3) для PICC типа A и ИСО/МЭК 14443-3:2011 (подраздел 7.4) для PICC типа B.

3.1.8 **сценарий** (scenario): Определенный типовой протокол и коммуникация, специфичная для приложения, которые используются вместе с методами испытаний, определенными в настоящем стандарте.

3.1.9 **исходное состояние испытания**; TIS (Test Initial State; TIS): Одно из состояний PICC до выполнения PICC определенной команды из набора команд.

3.1.10 **метод испытания** (test method): Метод проверки характеристик идентификационных карт с целью подтверждения их соответствия требованиям стандартов.

3.1.11 **целевое состояние испытания**; TTS (Test Target State; TTS): Одно из состояний PICC после выполнения PICC определенной команды из набора команд.

3.2 Обозначения и сокращения

(xxxxx)b	Обозначение бита информации
'XY'	Шестнадцатеричная система счисления, равно XY по основанию 16
ATA(cid)	Ответ на ATTRIB, т. е. (mbli+cid CRC_B), с произвольным шестнадцатеричным значением mbli [см. ИСО/МЭК 14443-3:2011 (подраздел 7.11)]

[1] Отсюда и до конца подраздела 3.1 нумерация пунктов последовательно изменена из-за включения в текст стандарта дополнительного пункта 3.1.5 (Изм. А1:2012).

ATTRIB(cid, fsdi)	Команда ATTRIB по умолчанию с PUPI из ATQB, CID = cid и значением кода максимального размера кадра = fsdi, т. е. ('1D' PUPI cid fsdi '01 00' CRC_B)
DUT	Испытуемое устройство (Device under test)
ESD	Электростатический разряд (Electrostatic Discharge)
$I(c)_n$(inf **[**,CID = cid**]** **[**,NAD = nad**]** **[**,~CRC**]**)	I-блок по ИСО/МЭК 14443-4 со сцеплением бит c∈{1,0}, числом блоков n∈{1,0} и информационным полем inf. По умолчанию не будет передаваться ни CID, ни NAD. Если определен CID = cid∈{0...15}, то он будет передан в качестве второго параметра. Если определен NAD = nad∈{0...'FF'}, то он будет передан как третий (или второй параметр, если CID не был передан). Если литерал '~CRC' не определен, то действительный CRC, соответствующий типу PICC, будет передан по умолчанию (т. е. CRC_A или CRC_B)
IUT	Тестируемая реализация (Implementation Under Test) (ИСО/МЭК 9646); в пределах области применения настоящего стандарта IUT представляет собой испытуемое PCD
LT	Нижний тестер (Lower Tester) (ИСО/МЭК 9646), эмуляционная часть PICC в оборудовании для испытаний PCD
m	Коэффициент модуляции (Modulation index)
Mute	Отсутствие ответа в течение указанного времени ожидания
N/A	Не применимо
PPS(cid, dri, dsi)	Запрос PPS по умолчанию с CID = cid, DRI = dri и DSI = dsi, т. е. ('D' + cid '11' dsi × 4 + dri CRC_A)
R(ACK **[**,CID = cid**]** **[**,~CRC**]**)$_n$	Блок R(ACK) по ИСО/МЭК 14443-4 с числом блоков n. Определение дополнительных CID и ~CRC символов аналогично описанию блока I(c)n выше
R(NAK **[**,CID = cid**]** **[**,~CRC**]**)$_n$	Блок R(NAK) по ИСО/МЭК 14443-4 с числом блоков n. Определение дополнительных CID и ~CRC символов аналогично описанию блока $I(c)_n$ выше
RATS(cid, fsdi)	Команда по умолчанию RATS с CID = cid и значением FSDI = fsdi, т. е. ('E0' fsdi x 16+cid CRC_A)
READY(I)	Состояние READY на каскадном уровне I, I∈ {1, 2, 3}; например, READY(2) — это каскадный уровень 2 для PICC
READY*(I)	Состояние READY* на каскадном уровне I, I∈ {1, 2, 3}; например, READY*(2) — это каскадный уровень 2 для PICC
REQB(N)	Команда REQB с N, как определено в ИСО/МЭК 14443-3:2011 (пункт 7.7.4)
S(WTX)(WTXM [,CID = cid][,~CRC])	Блок S(WTX) по ИСО/МЭК 14443-4 с параметром WTXM. Определение дополнительных CID и ~CRC символов аналогично описанию блока I(c)n выше
S(DESELECT **[**,CID = cid**]** **[**,~CRC**]**)	Блок S(DESELECT) по ИСО/МЭК 14443-4. Определение дополнительных CID и ~CRC символов аналогично описанию блока $I(c)_n$ выше
SAK(cascade)	Ответ на команду SELECT(I) с каскадным битом (бит 3), установленным на (1)b
SAK(complete)	Ответ на команду SELECT(I) с каскадным битом (бит 3), установленным на (0)b
SEL(c)	Код выборки уровня c (т. е. SEL(1) = '93', SEL(2) = '95', SEL(3) = '97')

SELECT(I)	Команда SELECT на каскадном уровне I, т. е. SELECT(1) = ('93 70' $UIDTX_1$ BCC CRC_A) SELECT(2) = ('95 70' $UIDTX_2$ BCC CRC_A) SELECT(3) = ('97 70' $UIDTX_3$ BCC CRC_A)
SLOTMARKER(n)	Команда Slot-MARKER с номером слота n, т. е. (16 × (n – 1) + 5 CRC_B)
TB-PDU	Блок Данных Протокола Передачи (Transmission Block Protocol Data Unit), который состоит либо из I-блока, либо из R-блока, либо из S-блока
TEST_COMMAND1(1)	Тестовая команда по умолчанию, состоящая из одного несцепленного I-блока Примечание — Эта команда зависит от согласованного значения максимального размера кадра PICC.
TEST_COMMAND1(n), n > 1	Тестовая команда по умолчанию, состоящая из n сцепленных I-блоков (сцепление PCD) Примечание — Эта команда зависит от согласованного значения максимального размера кадра PICC.
$TEST_COMMAND1(n)_k$	Поле INF k I-блока цепочки TEST_COMMAND1(n) Примечание — Эта команда зависит от согласованного значения максимального размера кадра PICC.
TEST_COMMAND2(n), n > 1	Тестовая команда по умолчанию, которая ожидает ответ, состоящий из n сцепленных I-блоков Примечание — Эта команда зависит от согласованного значения максимального размера кадра PCD.
TEST_COMMAND3	Тестовая команда по умолчанию, состоящая из одного I-блока, которой необходимо время на выполнение больше, чем FWT
TEST_RESPONSE1(n)	Поле INF ответа на TEST_COMMAND1(n) Примечание — Предполагается, что этот ответ всегда не сцеплен.
TEST_RESPONSE2(n)	Ответ на TEST_COMMAND2(n) Примечание — Этот ответ зависит от согласованного максимального значения размера кадра PCD.
$TEST_RESPONSE2(n)_k$	Поле INF k I-блока цепочки TEST_RESPONSE2(n) Примечание — Этот ответ зависит от согласованного максимального значения размера кадра PCD.
TEST_RESPONSE3	Ответный I-блок на TEST_COMMAND3 Примечание — Предполагается, что этот ответ всегда не сцеплен.
TM-PDU	Блок Данных Протокола Управления Тестированием (Test Management Protocol Data Unit) (ИСО/МЭК 9646-1, PDU)
!START	Старт передачи PICC (Start of PICC transmission) **(Введено дополнительно. Изм. А2:2012.)**
UIDTXI	Передаваемый UID из 32 бит данных на каскадном уровене I (см. таблицу 1)
UT	Верхний тестер (Upper Tester) (ИСО/МЭК 9646), мастер-компонента оборудования для испытаний PCD
UT_APDU	Блок Данных Прикладного Протокола Верхнего Тестера (Upper Tester Application Protocol Data Unit): пакет данных, которые должны быть отправлены PCD в LT через радиочастотный интерфейс

V_{load}	Напряжение постоянного тока. Измеренное на соединителе CON3 на эталонной PICC **(Введено дополнительно. Изм. А1:2012.)**
WUPB(N)	Команда WUPB с N, как определено в ИСО/МЭК 14443-3:2011 (пункт 7.7.4)
~X	Последовательность битов, состоящая из инвертированных битов последовательности битов X или другой последовательности битов, отличной от X
X[[a..b]]	Подпоследовательность битов из последовательности битов X, состоящая из битов между позициями a и b включительно. Если a > b, то последовательность пустая
X[[n]]	Бит в позиции n из последовательности битов X. Первый бит — в позиции 1
X[n]	Байт в позиции n из последовательности битов X. Первый байт — в позиции 1 (т. е. X[n] = X[[(n – 1) × 8 + 1..n × 8]])

Таблица 1 — Преобразование из UID в UIDTX

Уровень каскада	Одинарный UID PICC	Двойной UID PICC	Тройной UID PICC
$UIDTX_1$	UID0 UID1 UID2 UID3	'88' UID0 UID1 UID2	'88' UID0 UID1 UID2
$UIDTX_2$	—	UID3 UID4 UID5 UID6	'88' UID3 UID4 UID5
$UIDTX_3$	—	—	UID6 UID7 UID8 UID9

4 Нормальные условия испытаний

4.1 Нормальные климатические условия

Испытания проводят при температуре окружающей среды (23 ± 3) °C и относительной влажности воздуха от 40 до 60 %, если не оговорены иные климатические условия.

4.2 Предварительное кондиционирование

В настоящем стандарте методы испытаний не требуют проведения предварительного кондиционирования PICC или PCD.

4.3 Допускаемые отклонения

Отклонения значений характеристик испытательного оборудования (например, линейных размеров) и параметров испытательных режимов (например, параметров настройки испытательного оборудования) от указанных в стандарте значений не должны быть более ± 5 %, если не оговорены другие допускаемые отклонения.

4.4 Паразитная индуктивность

Резисторы и конденсаторы должны иметь пренебрежимо малую индукцию.

4.5 Суммарная погрешность измерений

Суммарная погрешность измерений по каждой величине, определяемой при испытаниях, должна быть указана в протоколе испытаний.

Основная информация дана в руководстве ИСО/МЭК 98-3:2008.

5 Оборудование и схемы для испытаний по параметрам, установленным в ИСО/МЭК 14443-1 и ИСО/МЭК 14443-2

В настоящем разделе определены испытательное оборудование и испытательные схемы для проверки работы PICC или PCD на соответствие требованиям ИСО/МЭК 14443-1:2008 и ИСО/МЭК 14443-2:2010. Испытательное оборудование включает в себя:

- средства измерений (см. 5.1);
- калибровочную катушку (см. 5.2);
- испытательную установку PCD (см. 5.3);
- эталонную PICC (см. 5.4).

Они описаны в следующих разделах.

5.1 Минимальные требования к средствам измерений

5.1.1 Осциллограф

Цифровой импульсный осциллограф должен быть способен осуществлять дискретизацию с частотой не менее 500 млн отсчетов в секунду с разрешением не менее 8 бит при оптимальном масштабировании и иметь общую минимальную полосу пропускания 250 МГц. Осциллограф должен иметь возможность выводить дискретные данные в виде текстового файла для облегчения математических и других операций, таких как кадрирование дискретных данных с использованием внешних программ (см. приложения E и F).

Примечание — Суммарная полоса пропускания представляет собой объединенную полосу пропускания осциллографа и измерительной установки.

5.2 Калибровочная катушка

В данном подразделе определены размер, толщина и характеристики калибровочных катушек 1 и 2.

Калибровочную катушку 1 используют только в испытательной установке PCD 1, а калибровочную катушку 2 — только в испытательной установке PCD 2.

5.2.1 Размер карты с калибровочной катушкой

Карта с калибровочной катушкой заключена в область высотой и шириной формата ID-1 в соответствии с ИСО/МЭК 7810:2003, которая содержит один виток катушки, концентричный с контуром карты (см. рисунок 1).

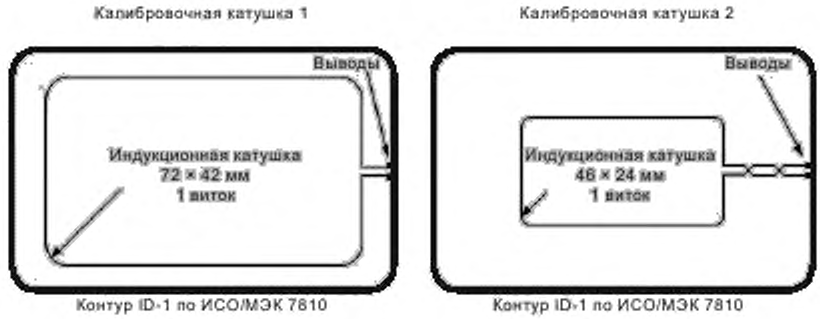


Рисунок 1 — Калибровочные катушки 1 и 2

5.2.2 Толщина и материал карты с калибровочной катушкой

Толщина карты с калибровочной катушкой должна быть меньше толщины карты ID-1. Она должна быть сделана из подходящего изоляционного материала.

5.2.3 Характеристики катушки индуктивности

Калибровочная катушка на карте должна иметь один виток. Отклонения размеров витка должны быть в пределах ± 2 %.

Наружные размеры витка калибровочной катушки 1 должны составлять 72 × 42 мм с радиусом закругления углов контура 5 мм.

Примечание 1 — Площадь, в пределах которой сосредоточено электромагнитное поле, составляет приблизительно 3000 мм^2.

Примечание 2 — На частоте 13,56 МГц индуктивность катушки составляет приблизительно 250 нГн, а сопротивление — приблизительно 0,4 Ом.

Калибровочный коэффициент разомкнутой цепи для калибровочной катушки 1 составляет 0,318 В (среднеквадратичное значение) на ампер на метр (среднеквадратичное значение) [эквивалентно 900 мВ (двойная амплитуда) на ампер на метр (среднеквадратичное значение)].

Наружные размеры витка калибровочной катушки 2 должны составлять 47 × 24 мм с радиусом закругления углов контура 2 мм.

Примечание 3 — Площадь, в пределах которой сосредоточено электромагнитное поле, составляет приблизительно 1100 мм^2.

Примечание 4 — На частоте 13,56 МГц индуктивность катушки составляет приблизительно140 нГн, а сопротивление — приблизительно 0,3 Ом.

Калибровочный коэффициент разомкнутой цепи для калибровочной катушки 2 составляет 0,118 В (среднеквадратичное значение) на ампер на метр (среднеквадратичное значение) [эквивалентно 333 мВ (двойная амплитуда) на ампер на метр (среднеквадратичное значение)].

Обмотка должна быть выполнена в виде печатной катушки на плате (PCB) с медным покрытием толщиной 35 мкм. Ширина витка должна быть 500 мкм с допускаемым отклонением ± 20 %. Размер контактных площадок должен составлять 1,5 × 1,5 мм.

Пробник осциллографа должен иметь высокий импеданс и входную проводимость, эквивалентную параллельной емкости C_p < 14 пФ и параллельному сопротивлению R_p > 9 кОм при 13,56 МГц, чтобы измерять напряжение (разомкнутой цепи), наведенное в катушке индуктивности.

Примечание 5 — Провод заземления пробника осциллографа с высоким импедансом должен быть как можно короче, менее 20 мм, или коаксиально соединен.

5.3 Испытательная установка PCD

Определены две испытательные установки PCD:

- испытательная установка PCD 1 для PICC классов 1, 2 и 3 и для PICC, которые не требуют соответствия классу;
- испытательная установка PCD 2 для PICC классов 4, 5 и 6.

Каждая испытательная установка PCD должна состоять из круглой испытательной антенны PCD и двух параллельных катушек считывания: катушки считывания *a* и катушки считывания *b*. Принципиальная схема испытательной установки показана на рисунке 2. Катушки считывания должны быть соединены таким образом, чтобы сигнал одной катушки был в противофазе с сигналом второй катушки. Потенциометр P1 на 10 Ом служит для точной настройки точки равновесия, когда катушки считывания не нагружены PICC или какой-либо цепью магнитного сопряжения. Емкостная нагрузка пробника, включая его паразитную емкость, должна быть менее 14 пФ.

Примечание 1 — Емкость соединений и пробника осциллографа должна быть сведена к минимуму для воспроизводимости результатов измерений.

Примечание 2 — Для того чтобы избежать непреднамеренного смещения в случае несимметричной настройки диапазона перестройки, потенциометр P1 настроен только на 10 Ом. Если настройка не может быть компенсирована потенциометром P1 в 10 Ом, то должна быть проверена общая симметрия установки.

Примечание 3 — Провод заземления пробника осциллографа с высоким импедансом должен быть как можно короче, менее 20 мм, или коаксиально соединен.

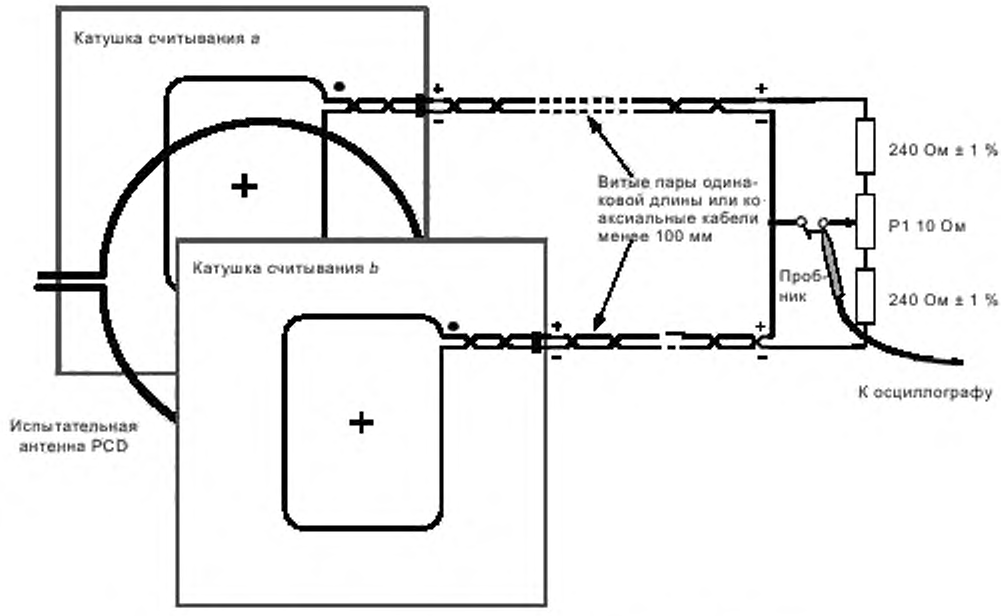


Рисунок 2 — Принципиальная схема испытательной установки

5.3.1 Испытательная антенна PCD

Испытательная антенна PCD в испытательной установке PCD 1 должна иметь диаметр 150 мм.

Испытательная антенна PCD в испытательной установке PCD 2 должна иметь диаметр 100 мм.

Конструкция каждой испытательной антенны PCD должна соответствовать рисункам, приведенным в приложении A.

Согласование каждой испытательной антенны PCD достигается путем использования анализатора импеданса, или сетевого анализатора, или измерителя LCR. Если нет ни анализатора импеданса, ни сетевого анализатора, ни измерителя LCR, то согласование может быть выполнено с помощью процедур, изложенных в приложении B.

5.3.2 Катушки считывания

Размер катушки считывания 1 в испытательной установке PCD 1 должен быть равен 100 × 70 мм с радиусом закругления 10 мм.

Размер катушки считывания 2 в испытательной установке PCD 2 должен быть равен 60 × 47 мм с радиусом закругления 10 мм.

Конструкция каждой катушки считывания должна соответствовать рисункам, приведенным в приложении C.

5.3.3 Монтаж испытательного PCD

Катушки считывания 1 и испытательная антенна PCD 1 должны быть установлены параллельно, при этом катушка считывания и антенная катушка должны иметь общую ось и расстояние между активными проводниками должно составлять 37,5 мм, как показано на рисунке 3.

Катушки считывания 2 и испытательная антенна PCD 2 должны быть установлены параллельно, при этом катушка считывания и антенная катушка должны иметь общую ось и расстояние между активными проводниками должно составлять 23 мм, как показано на рисунке 3.

Отклонения размеров должны быть менее ± 0,5 мм. Расстояния от катушки в DUT и от калибровочной катушки до катушки в испытательной антенне PCD должны быть одинаковыми.

Примечание — Эти расстояния выбирают так, чтобы создать сильное и однородное магнитное поле в положении DUT.

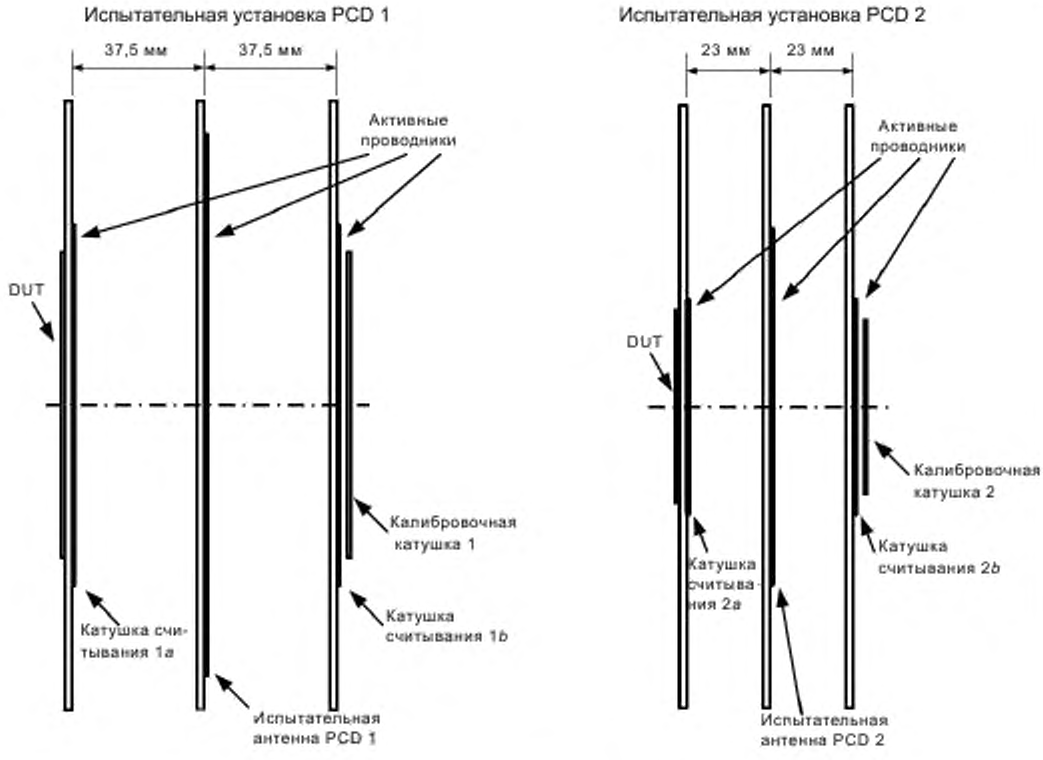


Рисунок 3 — Испытательная установка PCD 1 и испытательная установка PCD 2

(Измененная редакция. Изм. А1:2012.)

5.4 Эталонные PICC

Эталонные PICC предназначены для проверки способности PCD:
- генерировать поле напряженностью не менее H_{min} и не более H_{max};
- передавать мощность на PICC;
- передавать модулированный сигнал на PICC;
- получать модулированный сигнал нагрузки от PICC

в пределах своей рабочей области.

5.4.1 Размеры эталонной PICC

Эталонная PICC должна состоять из области высотой и шириной формата ID-1 по ИСО/МЭК 7810:2003, содержащей катушки индуктивности. Другая область, внешняя по отношению к первой, содержащая микросхему, эмулирующую требуемые функции PICC, должна быть присоединена таким образом, чтобы обеспечивать ее размещение в испытательных установках без создания помех испытаниям. Размеры должны быть такими, как показано на рисунке 4.



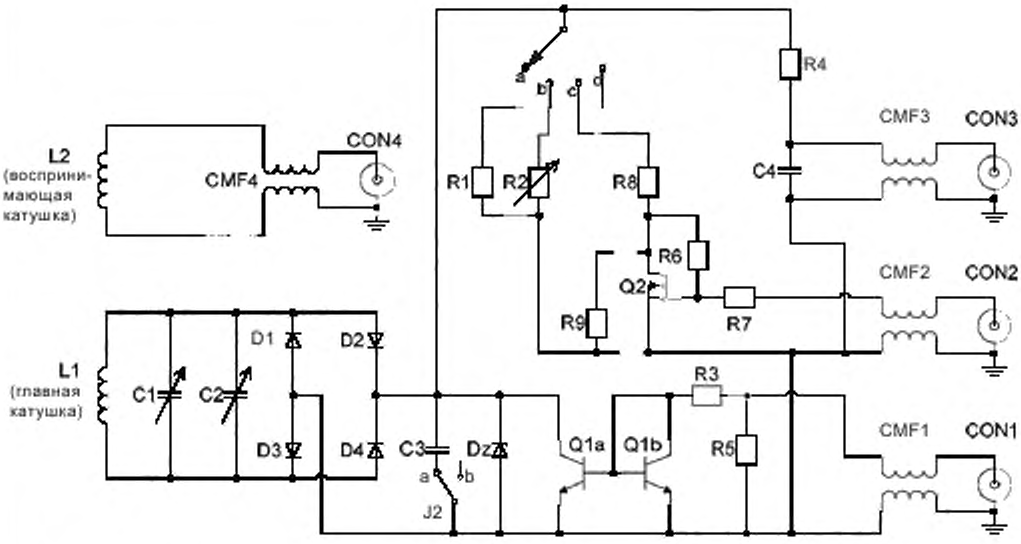
Рисунок 4 — Размеры эталонной PICC

5.4.2 Конструкция эталонной PICC

Топология катушек эталонных PICC определена в приложении D. Если используют выводы между катушками и микросхемой, то они должны оказывать минимальное воздействие (если имеется) на радиочастотные измерения.

(Измененная редакция. Изм. А1:2012.)

Эталонная PICC должна иметь электрическую схему, как определено на рисунке 5, и значения параметров компонентов схемы, как определено в таблице 2.



Примечание — Положение d перемычки — J1 — RFU[1].

Рисунок 5 — Электрическая схема эталонной PICC

(Измененная редакция. Изм. А4:2012.)

[1] RFU — зарезервировано для использования в будущем.

Таблица 2 — Список компонентов эталонной PICC

Компонент	Значение	Компонент	Значение
L1	См. приложение D	C1	От 7 до 50 пФ[b]
L2	См. приложение D	C2	От 3 до 10 пФ[b]
R1	1,8 кОм	C3	27 пФ
R2	От 0 до 2 кОм[a]	C4	1 нФ
R3	220 Ом	D1, D2, D3, D4	BAR43S или аналог[c]
R4	51 кОм	Dz	BZX84, 15 В, или аналог[c]
R5	51 Ом	Q1a, Q1b	BCV61A или аналог
R6	500 кОм	Q2	BSS83 или аналог
R7	110 кОм	CMF1, CMF2, CMF3, CMF4	ACM3225-102-2P или аналог
R8	51 Ом	CON1, CON2, CON3, CON4	Радиочастотный соединитель
R9	1,5 кОм		

[a] Необходимо использовать многовитковый потенциометр (число витков ≥ 10).
[b] Коэффициент добротности должен быть выше 100 при 13,56 МГц.
[c] Следует проявлять осторожность в отношении параметров Cj (емкость перехода), Cp (емкость монтажа), Ls (последовательная индуктивность) и Rs (последовательное сопротивление) эквивалентных диодов. Эти значения могут быть не представлены в листе с параметрами.

На CON1 должен быть подан сигнал нагрузочной модуляции. Нагрузочная модуляция может быть определена в испытательной установке PCD. Если генератор сигнала нагрузочной модуляции не используют, то он должен быть отключен или установлен на 0 В.

Нагрузку эталонной PICC можно настраивать с помощью напряжения на CON2 до тех пор, пока на CON3 не появится требуемое напряжение постоянного тока.

Напряжение постоянного тока эталонной PICC может быть измерено на CON3 с помощью вольтметра с высоким импедансом, при этом соединительные провода должны быть скручены или быть коаксиальными.

Параметры формы сигнала PCD подбирают на CON4, используя пробник осциллографа с высоким импедансом. Провод заземления пробника осциллографа с высоким импедансом должен быть как можно короче, менее 20 мм, или коаксиально соединен.

Положение a перемычки J2 используют для испытаний при скоростях передачи *fc*/128, *fc*/64, *fc*/32 и *fc*/16 бит/с.

Положение b перемычки J2 используют для испытаний при скоростях передачи *fc*/8, *fc*/4 и *fc*/2 бит/с.

(Введены дополнительно. Изм. А4:2012.)

5.4.3 Настройка резонансной частоты эталонной PICC

Резонансная частота эталонной PICC должна быть откалибрована с использованием следующей процедуры.

a) Установить перемычку J1 в положение a.

b) Подключить калибровочную катушку непосредственно к генератору сигналов, а соединитель CON3 эталонной PICC — к вольтметру с высоким импедансом. Подключить все остальные соединители к тем же устройствам, что и при испытании.

c) Расположить эталонную PICC на расстоянии d = 10 мм над калибровочной катушкой таким образом, чтобы оси двух катушек (калибровочной катушки и главной катушки эталонной PICC) совпали (см. рисунок 6).

d) Возбудить калибровочную катушку с набором синусоидальных колебаний на нужной резонансной частоте.

e) Настроить емкости C1 и C2 эталонной PICC для достижения максимального постоянного тока на CON3.

f) Настроить уровень возбуждения генератора сигналов, чтобы получить на CON3 напряжение постоянного тока V_{load}.

g) Повторять этапы e) и f), пока не будет достигнуто максимальное напряжение V_{load} после этапа e).

(Измененная редакция. Изм. А1:2012.)

h) Откалибровать испытательную установку PCD, чтобы создать рабочий режим с H_{min} на калибровочной катушке.

i) Поместить эталонную PICC в положение DUT в испытательной установке PCD. Переключить перемычку J1 в положение b и настроить R2 для получения напряжения постоянного тока V_{load}. Измеряемого на соединителе CON3. Режим рабочего поля проверить, контролируя напряжение на калибровочной катушке, и при необходимости настроить.

(Измененная редакция. Изм. А1:2012.)

j) Повторить этапы от b) до g) с полученным значением R2.

Примечание — Вместо генератора сигналов может быть использован векторный анализатор цепей, если обеспечивается достаточная мощность для создания V_{load} на CON3 при достижении максимальной резистивной части измеренного комплексного сопротивления калибровочной катушки.

(Измененная редакция. Изм. А1:2012.)

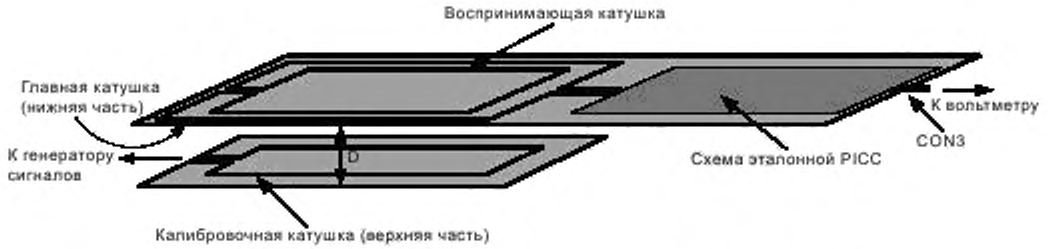


Рисунок 6 — Принципиальная схема настройки частоты эталонной PICC

5.5 Испытательная установка EMD

5.5.1 Общее описание

Испытательная установка EMD[1] состоит из:

- генератора сигналов с низким уровнем фазового шума, который используют для синтезирования тестового образца EMD и тестовых команд PCD, посылаемых испытуемой PICC;
- испытательной установки PCD;
- устройства анализа амплитуды сигнала:

либо устройства, получающего сигнал (например, осциллограф), и соответствующего программного обеспечения для вычислений,

либо анализатора спектра (см. дополнительные условия в 5.5.2).

Устройство анализа амплитуды сигнала должно быть способно осуществлять измерения зависимости мощности от времени с фиксированной частотой, фиксированной полосой пропускания, расширенным динамическим диапазоном, низкой погрешностью и высокой временной разрешающей способностью.

Примечание — EMD-испытания PICC могут быть выполнены с использованием радиочастотного выходного сигнала серийного PCD. При EMD-испытании PICC можно использовать эмулятор PICC для генерирования тестового образца EMD.

5.5.2 Вычисление зависимости мощности от времени

Начальная точка зарегистрированного сигнала должна быть обработана посредством окна Барлетта по ровно двум циклам поднесущей. Преобразование Фурье этих отсчетов, обработанных методом окна, дает одно значение мощности. Путем перемещения окна Барлетта с шагом 1/*fc* от начала

[1] EMD — электромагнитная помеха (electromagnetic disturbance).

до конца зарегистрированного сигнала искомый результат зависимости мощности от времени будет окончательно вычислен.

Примечание — Полученная полоса пропускания в 3 дБ описанного выше окна составляет 531 кГц, а его шум равен полосе пропускания величиной 843 кГц.

Вычисление зависимости мощности от времени выполняют при *fc* + *fs* и *fc* – *fs*, используя масштабирование таким образом, чтобы чистый синусоидальный сигнал был получен по всей амплитуде. Пример вычисления приведен в приложении J.

В случае использования анализатора спектра он должен иметь как минимум аналогичную полосу пропускания для анализа. Он должен пройти предварительное испытание на шумовой порог по 5.5.3, при этом иметь некоторый дополнительный запас 10/*fc* по требованию к $t_{E,PICC}$ и не должен создавать выбросов выше предела EMD.

5.5.3 Предварительное испытание на шумовой порог

Для того чтобы обеспечить высокий динамический диапазон и достаточную чувствительность, должен быть измерен шумовой порог испытательной установки EMD и получены удовлетворительные результаты. Цель этого предварительного испытания — проверить, что испытательное оборудование, используемое для измерения уровня EMD, удовлетворяет требованию к минимальному уровню шума.

Испытание на шумовой порог считают удовлетворительным, если стандартное отклонение шума как минимум в три раза меньше, чем предел EMD $V_{E,PICC}$ при измерениях по 5.5.3.1.

Стандартное отклонение шума определяют путем вычисления среднеквадратичного значения результатов преобразования Фурье (см. 5.5.2).

Примечание — Шумовой порог может быть получен либо с помощью 14-битового дискретизатора при частоте дискретизации в 100 млн отсчетов в секунду, либо с помощью 8-битового цифрового осциллографа при частоте дискретизации в 1000 млн отсчетов в секунду.

5.5.3.1 Процедура испытания

Выполняют следующие этапы, чтобы оценить уровень шума как минимум при H_{min} и H_{max}.

a) Настроить эталонную PICC на 13,56 МГц.

b) Отрегулировать радиочастотную мощность генератора сигналов, передаваемую в испытательную в антенну PCD для создания необходимой напряженности поля. Измеряемой с помощью калибровочной катушки.

c) Поместить эталонную PICC в положение DUT в испытательной установке PCD, переключить перемычку J1 в положение b и настроить R2 таким образом, чтобы получить напряжение постоянного тока 6 В на CON3. В качестве альтернативы перемычку J1 можно установить в положение c и подаваемое на CON2 напряжение настраивать до получения напряжения постоянного тока 6 В на CON3. В обоих случаях режим рабочего поля проверять, контролируя напряжение на калибровочной катушке, и при необходимости настроить.

d) Записывать сигнал с катушки считывания в течение периода времени не менее 250 мкс.

e) Вычислить стандартные отклонения шума на частотах *fc* + *fs* и *fc* – *fs*, используя подходящее программное обеспечение, как, например, программа из приложения J. Проверить, что полученные стандартные отклонения шума в три раза меньше $V_{E,PICC}$.

5.5.3.2 Протокол испытаний

В протоколе испытаний должны быть зафиксированы стандартные отклонения шума на частотах *fc* + *fs* и *fc* – *fs* и должно быть указано, удовлетворяют ли результаты испытания предъявляемым требованиям.

(Введены дополнительно. Изм. A2:2012.)

6 Испытания по параметрам, установленным в ИСО/МЭК 14443-1

6.1 Испытания PCD

6.1.1 Переменное магнитное поле

6.1.1.1 Назначение

Данное испытание устанавливает, что PCD генерирует поле, не превышающее среднего значения, установленного в ИСО/МЭК 14443-1:2008, во всех возможных положениях PICC.

6.1.1.2 Процедура испытаний

a) Настроить эталонную PICC на 19 МГц, как описано в перечислениях a) — g) 5.4.3.

b) Откалибровать испытательную установку PCD для получения среднего значения поля, установленного в ИСО/МЭК 14443-1:2008 (пункт 4.4), на калибровочной катушке.

c) Поместить эталонную PICC в положение DUT в испытательной установке PCD. Переключить перемычку J1 в положение b и настроить R2 для получения напряжения постоянного тока 3 В. Измеряемого на соединителе CON3. Режим рабочего поля проверяют, контролируя напряжение на калибровочной катушке, и при необходимости настраивают.

ПРЕДУПРЕЖДЕНИЕ — Значение R2 должно быть в пределах от 55 до 65 В.

d) Установить эталонную PICC в любое возможное положение PICC. Напряжение постоянного тока на CON3 не должно превышать 3 В.

e) Если оно превышает, то использовать один тот же коэффициент пересчета, чтобы измерить максимальное и среднее напряжение постоянного тока и преобразовать в напряженность поля для проверки максимального и среднего значений поля, установленных в ИСО/МЭК 14443-1:2008 (пункт 4.4), за период 30 с.

6.1.1.3 Протокол испытаний

Протокол испытаний должен содержать значение напряжения постоянного тока. Измеренное на CON3.

6.2 Испытания PICC

6.2.1 Переменное магнитное поле

Цель данного испытания — проверить поведение PICC при воздействии переменного магнитного поля на частоте 13,56 МГц.

6.2.1.1 Оборудование

Для создания переменного магнитного поля используют испытательную установку PCD.

6.2.1.2 Процедура испытаний

Применяют следующую процедуру.

a) Отрегулировать радиочастотную мощность, подаваемую генератором сигналов на испытательную антенну PCD, до достижения напряженности поля среднего уровня по ИСО/МЭК 14443-1:2008 (пункт 4.4). Измеряемой при помощи калибровочной катушки.

(Измененная редакция. Изм. А1:2012.)

b) Поместить испытуемую PICC в положение DUT и при необходимости сразу перенастроить радиочастотный передатчик в испытательной антенне PCD на требуемую напряженность поля.

c) По истечении 5 мин убрать PICC из положения DUT как минимум на 5 с.

d) Отрегулировать радиочастотную мощность, подаваемую генератором сигналов на испытательную антенну PCD, до достижения напряженности поля максимального уровня по ИСО/МЭК 14443-1:2008 (пункт 4.4). Измеряемой при помощи калибровочной катушки.

(Измененная редакция. Изм. А1:2012.)

e) Поместить испытуемую PICC в положение DUT и при необходимости сразу перенастроить радиочастотный передатчик в испытательной антенне PCD на требуемую напряженность поля.

f) Применять в течение 5 мин. 100 % амплитудную модуляцию (ASK) для этого поля со следующим циклом нагрузки:

- 5 с при 0 А/м (среднеквадратичное значение);
- 25 с при максимальном уровне, указанном в ИСО/МЭК 14443-1:2008 (пункт 4.4).

(Измененная редакция. Изм. А1:2012.)

g) Проверить, что PICC работает надлежащим образом.

6.2.1.3 Протокол испытаний

В протоколе испытаний должно быть указано, работает PICC надлежащим образом или нет.

6.2.2 Испытание на воздействие статического электричества

Цель данного испытания — проверить работу интегральной схемы карты при воздействии электростатического разряда (ESD) на испытуемый образец. Испытуемую PICC подвергают воздействию имитируемого электростатического разряда (ESD, модель человеческого тела) и затем проверяют ее основные функции (см. рисунок 7).



Рисунок 7 — Цепь испытания на воздействие ESD

6.2.2.1 Оборудование

См. МЭК 61000-4-2:2008.

a) Основные технические характеристики генератора ESD:

- накопительная емкость — 150 пФ ± 10 %;
- разрядное сопротивление — 330 Ом ± 10 %,
- зарядное сопротивление — от 50 до 100 МОм;
- время нарастания — от 0,7 до 1 нс.

b) Предпочтительные характеристики оборудования:

- тип оборудования: настольный;
- метод разряда: прямое воздействие воздушного разряда на испытуемое оборудование;
- разрядные электроды генератора ESD: зонд диаметром 8 мм с круглым наконечником.

6.2.2.2 Процедура испытания

Подключить штырь заземления оборудования к электропроводной пластине, на которой размещена PICC.

Приложить разряд последовательно в нормальной полярности к каждой из 20 зон, показанных на рисунке 8. Далее повторить эту же процедуру с обратной полярностью. Необходимо выдерживать интервал не меньше 10 с для охлаждения между последовательными разрядными импульсами.

ВНИМАНИЕ — Если PICC содержит контакты, то их поверность должна быть обращена вверх, а зона, включающая в себя контакты, не должна подвергаться воздействию этого разряда.

По завершении испытания проверить, работает ли PICC надлежащим образом.



Рисунок 8 — Тестовые зоны PICC при испытании на воздействие ESD

6.2.2.3 Протокол испытаний

В протоколе испытаний должно быть указано, работает ли PICC надлежащим образом или нет.

7 Испытания по параметрам, установленным в ИСО/МЭК 14443-2

7.1 Испытания PCD

Все испытания PCD, описанные ниже, должны быть проведены в его рабочей области, как определено изготовителем PCD, для каждого поддерживаемого класса.

Все испытания PCD по параметрам ИСО/МЭК 14443-2 должны быть выполнены с использованием эталонных PICC классов 1, 2 и 3 и других эталонных PICC (последнее необязательно), соответствующих дополнительным классам, поддерживаемым PCD, с соответствующими параметрами и испытательной установкой PCD, как определено в таблице 3.

Таблица 3 — Параметры классов

Класс	Эталонная PICC	V_{load}, В	$R2_{min}$, Ом	$R2_{max}$, Ом	Испытательная установка PCD
1	1	6	870	1070	Испытательная установка PCD 1
2	2	4,5	1030	1260	Испытательная установка PCD 1
3	3	4,5	1080	1320	Испытательная установка PCD 1
4	4	4,5	990	1210	Испытательная установка PCD 2
5	5	4,5	960	1170	Испытательная установка PCD 2
6	6	4,5	700	900	Испытательная установка PCD 2

Примечание — Для V_{load} может быть установлено значение 4,5 В для всех классов в следующих изданиях ИСО/МЭК 10373-6.

(Измененная редакция. Изм. А1:2012 и Поправка Cor.1:2013.)

7.1.1 Напряженность поля PCD

7.1.1.1 Цель

Данное испытание предназначено для измерения напряженности поля, создаваемого PCD в его рабочей области.

Примечание — В данном испытании учитывают нагрузку PICC на PCD.

Максимальное и минимальное значения напряженности поля, используемые с каждой эталонной PICC, указаны в ИСО/МЭК 14443-2:2010/Изм. А2:2012 (таблица 1).

(Введен дополнительно. Изм. А1:2012.)

7.1.1.2 Процедуры испытаний

Для H_{max} применяют следующую процедуру.

a) Настроить эталонную PICC на частоту 19 МГц, как описано в перечислениях a) — g) 5.4.3.

b) Откалибровать испытательную установку PCD, чтобы создать рабочий режим с H_{max} на калибровочной катушке.

c) Поместить эталонную PICC в положение DUT в испытательной установке PCD. Переключить перемычку J1 в положение b и настроить R2 таким образом, чтобы получить напряжение постоянного тока 3 В. Измеряемое на соединителе CON3. В качестве альтернативы перемычку J1 установить в положение c, и подаваемое на CON2 напряжение настраивать до получения напряжения постоянного тока 3 В на соединителе CON3. В обоих случаях режим рабочего поля проверять, контролируя напряжение на калибровочной катушке, и при необходимости настроить.

ПРЕДУПРЕЖДЕНИЕ — Значение R2 должно быть в пределах от 75 до 85 Ом. Проверить этот диапазон хотя бы один раз перед использованием альтернативного метода.

d) Расположить эталонную PICC в пределах рабочей области испытуемого PCD. Напряжение постоянного тока на CON3 не должно превышать 3 В.

Для H_{min} применяют следующую процедуру.

a) Настроить эталонную PICC на частоту 13,56 МГц, как описано в 5.4.3.

b) Поместить эталонную PICC в положение DUT в испытательной установке PCD, создающей рабочий режим с H_{min} на калибровочной катушке. Проверить, что перемычка J1 установлена в положение b и напряжение постоянного тока V_{load}, как определено в таблице 3. Измеряют на соединителе

CON3. В качестве альтернативы перемычка J1 может быть установлена в положение c, и напряжение на CON2 настраивают для получения напряжения постоянного тока V_{load} согласно таблице 3 на соединителе CON3. В обоих случаях режим рабочего поля должен быть проверен путем контроля напряжения на калибровочной катушке и при необходимости настроен.

ПРЕДУПРЕЖДЕНИЕ — Значение R2 должно быть в пределах от $R2_{min}$ до $R2_{max}$, как определено в таблице 3. Проверить этот диапазон хотя бы один раз перед использованием альтернативного метода.

c) Расположить эталонную PICC в пределах рабочей области испытуемого PCD. Напряжение постоянного тока на CON3 должно превышать V_{load}, как определено в таблице 3.

7.1.1.3 Протокол испытаний

Протокол испытаний должен подтверждать рабочую область, в которой напряжение постоянного тока. Измеряемое на CON3 для R2, или напряженность поля, настраиваемая на H_{min} и H_{max} с помощью резистора переменной нагрузки, удовлетворяет требованиям, определенным в перечислениях d) и c) двух процедур по 7.1.1.2.

7.1.2 Исключен.

7.1.3 Исключен.

(Измененная редакция. Изм. А1:2012.)

7.1.4 Коэффициент модуляции и форма колебаний сигнала

7.1.4.1 Цель

Цель данного испытания — определить коэффициент модуляции поля PCD, а также время нарастания и время спада и значения выбросов, как определено в ИСО/МЭК 14443-2, для всех поддерживаемых скоростей передачи от PCD к PICC.

(Измененная редакция. Изм. А4:2012.)

7.1.4.2 Процедура испытаний

a) Установить калибровочную катушку в произвольном положении в установленной рабочей области и вывести данные напряжения катушки индуктивности на подходящий осциллограф. Определить коэффициент модуляции и характеристики формы колебаний сигнала, используя инструментарий для анализа, определенный в приложении E.

b) Настроить эталонную PICC на частоту 16,5 МГц, как описано в перечислениях a) — g) 5.4.3, и переключить перемычку J1 в положение c.

c) Поместить эталонную PICC в определенном положении в рабочей области PCD.

d) Приложить и настроить напряжение постоянного тока на CON2 таким образом, чтобы получить напряжение постоянного тока на соединителе CON3 V_{load}, как определено в таблице 3.

(Измененная редакция. Изм. А1:2012.)

Примечание 1 — Если напряжение постоянного тока 6 В не может быть достигнуто при выбранном положении, то для испытания используют максимально достижимое напряжение.

e) Если немодулированное напряжение на CON4. Измеренное с помощью подходящего осциллографа (требования по 5.1.1), ниже 1 В (двойная амплитуда), то следует использовать другую воспринимающую катушку для определения характеристик формы колебания сигнала.

Примечание 2 — Другая воспринимающая катушка должна иметь форму восьмерки с радиусом 15 мм и быть расположена дальше всего от эталонной PICC для минимизации взаимодействия и как можно ближе к антенне PCD для минимизации наведенного напряжения.

f) Определить коэффициент модуляции и характеристики формы колебаний сигнала по напряжению на CON4 или на противоположной воспринимающей катушке, используя инструментарий для анализа, определенный в приложении E.

g) Повторить этапы с c) по f) для различных положений в пределах рабочей области и для всех поддерживаемых скоростей передачи от PCD к PICC.

(Измененная редакция. Изм. А4:2012.)

Примечание 3 — Выбранное положение калибровочной катушки в пределах рабочей области не влияет на результат.

Примечание 4 — Нагрузка эталонной PICC не означает наихудший эффект нагрузки PICC. Самый высокий эффект нагрузки может быть достигнут с помощью резонансных частот, близких к несущим частотам (т. е. 15 или 13,56 МГц).

7.1.4.3 Протокол испытаний

Протокол испытаний должен содержать измеренное значение коэффициента модуляции поля PCD, время нарастания и время спада, а также значения выбросов в пределах рабочей области в нагруженном и ненагруженном состояниях.

7.1.5 Прием сигнала нагрузочной модуляции

7.1.5.1 Цель

Цель данного испытания — проверить, что PCD безошибочно регистрирует нагрузочную модуляцию PICC, соответствующую требованиям ИСО/МЭК 14443-2, для скоростей передачи *fc*/128, *fc*/8, *fc*/4 и *fc*/2 от PICC к PCD, если такие поддерживаются.

(Измененная редакция. Изм. А4:2012.)

7.1.5.2 Процедура испытаний

Эталонная PICC и процедура ее калибровки позволяют определить чувствительность PCD к нагрузочной модуляции. Эталонная PICC не эмулирует эффект нагрузки всех типов PICC.

a) Настроить эталонную PICC на частоту 13,56 МГц, как описано в 5.4.3, и переключить перемычку J1 в положение c.

b) Поместить эталонную PICC в определенном положении в рабочей области PCD.

c) Приложить и настроить напряжение постоянного тока на CON2 таким образом, чтобы получить напряжение постоянного тока на соединителе CON3 V_{load} согласно таблице 3.

(Измененная редакция. Изм. А1:2012.)

d) Увеличивать амплитуду модулированного сигнала на CON1 для получения откликов до тех пор, пока PCD не обнаружит не менее 10 последовательных откликов.

e) Поместить эталонную PICC в положение DUT в испытательной установке PCD.

f) Настроить испытательную установку PCD для создания напряженности поля *H*, которое даст такое же напряжение на CON3, и зафиксировать соответствующую напряженность поля путем считывания напряжения на калибровочной катушке.

g) Измерить амплитуду нагрузочной модуляции эталонной PICC V_{LMA} согласно описанию в 7.2.1 и сравнить ее с нормальным предельным значением, соответствующим зафиксированному значению напряженности поля. Этот измеренный уровень V_{LMA} определяет критерий чувствительности PCD для сравнения с нормальным предельным значением и выполнения этих контрольных измерений.

h) Повторить этапы от b) до g) для различных положений в пределах рабочей области для скоростей передачи *fc*/128, *fc*/8, *fc*/4 и *fc*/2 от PICC к PCD, если такие поддерживаются.

(Измененная редакция. Изм. А4:2012.)

i) Повторить этапы с a) по h) с резонансной частотой эталонной PICC 15 МГц.

Любое положение, в котором чувствительность PCD больше нормального предельного значения, считают не соответствующим рабочей области.

Примечание 1 — Эффективность испытания может быть расширена при использовании дополнительных резонансных частот ниже 13,56 МГц, например, 12 и 10 МГц.

Примечание 2 — Чувствительность PCD должна быть ниже нормального предельного значения для обеспечения хорошего приема сигнала нагрузочной модуляции PICC.

Примечание 3 — В данном испытании не проверяют, что прием PCD не зависит от фазы нагрузочной модуляции PICC. Следовательно, данное испытание не гарантирует безошибочный прием никакой PICC, соответствующей требованиям ИСО/МЭК 14443-2:2010.

7.1.5.3 Протокол испытаний

Протокол испытаний должен содержать данные о чувствительности PCD к нагрузочной модуляции в положениях, подвергаемых испытанию.

7.1.6 Испытание на устойчивость PCD к EMD

7.1.6.1 Цель

Цель данного испытания — определить, является ли PCD нечувствительным к амплитуде нагрузочной модуляции ниже $V_{E,PCD}$.

7.1.6.2 Процедура испытаний

a) Настроить эталонную PICC на частоту 13,56 МГц, как описано в 5.4.3, и переключить перемычку J1 в положение c.

b) Поместить эталонную PICC в определенном положении в рабочей области PCD.

c) Приложить и настроить напряжение постоянного тока на CON2 таким образом, чтобы получить напряжение постоянного тока 3 или 6 В на соединителе CON3, если поддерживается «Класс 1» в этом положении.

d) Отправить тестовую последовательность, как показано на рисунке Amd.2.1. Тестовая последовательность представляет собой действительный стандартный кадр, включающий в себя один одиночный байт (01011101)b. Амплитуда модулированного сигнала исходной нагрузки V_{EMD} тестовой последовательности должна быть достаточно низкой, чтобы PCD обнаружило ответ PICC, посланный на этапе e).

e) Непосредственно после этой тестовой последовательности, без интервала, послать соответствующий ответ PICC на команду PCD с амплитудой модулирующего сигнала нагрузки V_{LMA}. Измеренной в соответствии с 7.2.1, уровнем выше, например, в два раза, чем минимальное значение применяемой напряженности поля *H*.

f) Увеличивать V_{EMD} путем настройки напряжения на CON1 до тех пор, пока PCD не обнаружит ответ корректно. Это может быть определено путем контроля следующей команды PCD, идущей за ответом PICC (см. рисунок Amd.2.1).

g) Поместить эталонную PICC в положение DUT в испытательной установке PCD.

h) Настроить испытательную установку PCD для создания напряженности поля *H*, позволяющей получить то же самое напряжение на CON3, и записать соответствующую напряженность поля путем считывания напряжения на калибровочной катушке.

i) Установить текущее значение V_{EMD} на эталонной PICC путем применения зависимости мощности от времени согласно описанию в 5.5.2.

j) Сравнить измеренное значение V_{EMD} с $V_{E,PCD}$.

Повторить этапы с b) по j) для других заданных положений в пределах рабочей области.

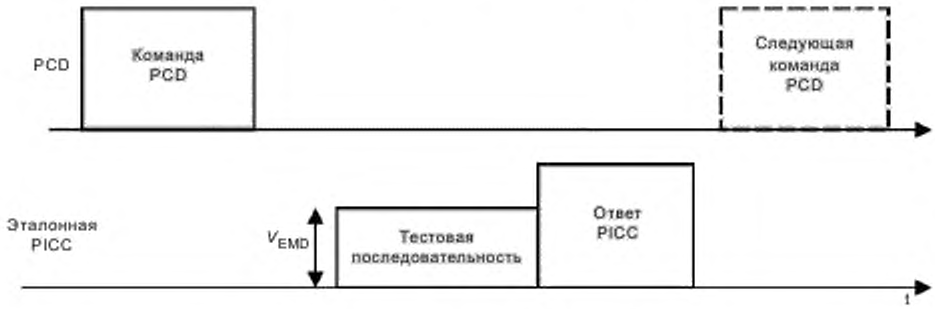


Рисунок Amd.2.1 — Испытание на устойчивость PCD (общее для типа A и типа B)

7.1.6.3 Протокол испытаний

В протоколе испытаний указывают, было ли PCD нечувствительно к амплитуде модулированного сигнала ниже $V_{E,PCD}$.

7.1.7 Испытание на восстановление PCD после воздействия EMD

7.1.7.1 Цель

Цель данного испытания — определить, нарушается ли настройка PCD тестовой последовательностью, посланной за время $t_{E,PCD}$ до ответа PICC.

7.1.7.2 Процедура испытания

a) Настроить эталонную PICC на 13,56 МГц (см. 5.4.3).

b) Откалибровать испытательную установку PCD, чтобы создать рабочий режим с H_{min} на калибровочной катушке.

c) Поместить эталонную PICC в положение DUT в испытательной установке PCD. Переключить перемычку J1 в положение c и настроить напряжение постоянного тока на CON2 таким образом, чтобы получить напряжение постоянного тока 6 В на соединителе CON3. Режим рабочего поля проверять, контролируя напряжение на калибровочной катушке, и настроить напряжение при необходимости.

d) Найти соответствующее управляющее напряжение на CON1, чтобы создать амплитуду нагрузочной модуляции V_{LMA}. Измеряемую по 7.2.1, выше, чем предел для H_{min}, определяемый по ИСО/МЭК 14443-2.

e) Поместить эталонную PICC в положение в пределах рабочей области PCD при напряжении постоянного тока на CON3 6 В.

f) Послать в последовательности, показанной на рисунке Amd.2.2, используя время $t_{E,PCD}$, которое соответствует минимуму FDT/TR0:

Примечание 1 — Время $t_{E,PCD}$ низкого уровня EMD является функцией FDT/TR0 в соответствии с ИСО/МЭК 14443-3:2011/Изм. А1:2011.

- тестовую последовательность, которая запускает передачу двух бит данных b1 = (0)b и (следом за b1) b2 = (1)b допустимым способом на PCD, но прерывается сразу после того, как послан второй бит, как показано на рисунке Amd. 2.3 для типа А и рисунке Amd.2.4 для типа В;

Примечание 2 — В зависимости от FDT/TR0 тестовая последовательность может начаться до завершения команды PCD.

- период без нагрузочной модуляции продолжительностью $t_{E,PCD}$;
- соответствующий ответ на команду PCD.

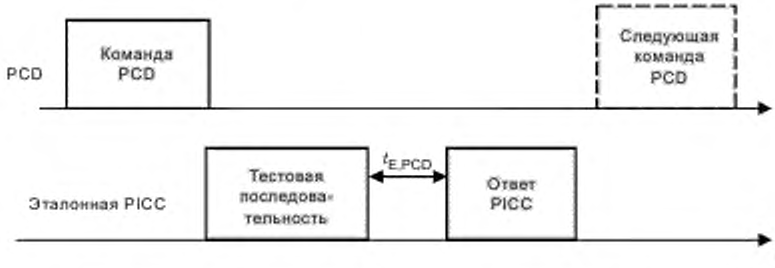


Рисунок Amd.2.2 — Последовательность испытания на восстановление после воздействия EMD (общая для типа А и типа В)



Рисунок Amd.2.3 — Тестовая последовательность для испытания на восстановление после воздействия EMD (тип А)

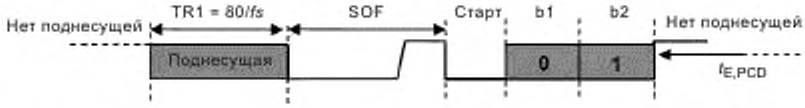


Рисунок Amd.2.4 — Тестовая последовательность для испытания на восстановление после воздействия EMD (тип В)

g) Проверить, работает ли PCD в том же порядке, как если бы тестовой последовательности не было. Это можно определить путем проверки следующей команды PCD, идущей за ответом PICC (см. рисунок Amd.2.2).

h) Повторить этапы с f) по g) 10 раз;

i) Повторить этапы с f) по h) путем замены минимума FDT/TR0 на максимум FDT/TR0.

7.1.7.3 Протокол испытаний

В протоколе испытаний должно быть указано, не были ли созданы помехи в PCD тестовой последовательностью, посланной за время $t_{E,PCD}$ перед ответом PICC (или была ли обеспечена возможность восстановления после тестовой последовательности).

(Введены дополнительно. Изм. А2:2012.)

7.2 Испытания PICC

7.2.1 Передача PICC

7.2.1.1 Цель

Цель данного испытания — определить амплитуду нагрузочной модуляции V_{LMA} PICC в пределах диапазона [H_{min}, H_{max}] рабочего поля, как установлено в ИСО/МЭК 14443-2, для скоростей передачи *fc*/128, *fc*/8, *fc*/4 и *fc*/2 от PICC к PCD, если такие поддерживаются. А также определить функциональные возможности PICC типа А и типа В в пределах их соответствующих диапазонов модуляции по ИСО/МЭК 14443-2:2010.

(Измененная редакция. Изм. А4:2012.)

7.2.1.2 Процедура испытаний

Этап 1. Необходимо использовать схему испытаний нагрузочной модуляции, показанную на рисунке 2, и испытательную установку PCD, показанную на рисунке 3, определенную для класса PICC [см. ИСО/МЭК 14443-2:2010/Изм. А2:2012 (пункт 8.2.2)]. Если PICC не требует соответствия требованиям одного конкретного класса, как определено в ИСО/МЭК 14443-1:2008/Изм. А1.2012, то следует выбирать испытательную установку PCD 1.

(**Измененная редакция. Изм. А1:2012.)**

Отрегулировать радиочастотную мощность, подаваемую генератором сигналов на антенну для испытаний PCD, чтобы создать электромагнитное поле необходимой напряженности. Измеряемой с помощью калибровочной катушки. Соединить выход схемы испытания нагрузочной модуляции (см. рисунок 2) с цифровым импульсным осциллографом. Потенциометр Р1 на 10 Ом должен быть поднастроен так, чтобы минимизировать остаточную несущую. Этот сигнал должен быть по меньшей мере на 40 дБ ниже уровня сигнала, полученного путем короткого замыкания одной из катушек считывания.

ПРЕДУПРЕЖДЕНИЕ — Испытание PICC методом амплитудной нагрузочной модуляции следует проводить путем увеличения напряженности поля, начиная с 0 А/м (среднеквадратичное значение), и таким образом проверять правильность операций PICC, начиная с H_{min}.

Этап 2. Испытуемая PICC должна быть установлена в положение DUT концентрично с катушкой считывания *a*. Радиочастотный передатчик испытательной антенны PCD должен быть повторно отрегулирован на требуемую напряженность поля.

Последовательность команд REQA или REQB по ИСО/МЭК 14443-3 должна быть послана испытательным PCD для получения сигнала или отклика нагрузочной модуляции от PICC, когда происходит испытание передачи PICC при скорости передачи *fc*/128. Последовательность S(PARAMETERS), определенная в ИСО/МЭК 14443-4, и I-блок должны быть посланы испытательным PCD для получения сигнала или отклика нагрузочной модуляции от PICC, когда происходит испытание передачи PICC при дополнительных скоростях передачи *fc*/8, *fc*/4 и *fc*/2.

(Измененная редакция. Изм. А4:2012.)

Примечание 1 — Для скоростей передачи *fc*/64, *fc*/32 и *fc*/16 не требуется проведения испытания нагрузочной модуляции, т. к. эти скорости передачи используют ту же частоту поднесущей, что и *fc*/128.

(Введено дополнительно. Изм. А4:2012.)

Вывести на экран цифрового импульсного осциллографа сегмент из не менее чем шести циклов колебаний сигнала нагрузочной модуляции поднесущей и сохранить выбранные данные в файл для анализа с помощью программного обеспечения (см. приложение F).

Примечание 2 — Следует внимательно отнестись к выбору метода синхронизации для нагрузочной модуляции низкой амплитуды.

(Измененная нумерация. Изм. А4:2012.)

Ровно шесть циклов поднесущей из отобранных модулированных колебаний должны быть преобразованы в ряд Фурье с окнами Бартлетта, используя подходящее программное обеспечение (например, из приложения F). Использовать дискретное преобразование Фурье с таким масштабом, чтобы чистый синусоидальный сигнал давал в результате свою максимальную величину. Чтобы свести к

минимуму влияние переходных процессов, не следует учитывать при анализе цикл поднесущей, который следует сразу после немодулированного периода или сдвига фазы поднесущей. Дискретное преобразование Фурье должно быть выполнено на частотах боковых полос, генерируемых испытуемой PICC, т. е. *fc* + *fs* и *fc* – *fs*.

Результирующие пиковые амплитуды верхней и нижней боковых полос на частотах *fc* + *fs* и *fc* – *fs* должны быть выше значения, определенного в ИСО/МЭК 14443-2:2010 (пункт 8.2.2).

Примечание 3 — Для испытания нагрузочной модуляцией PICC типа B опция осциллографа FFT (быстрое преобразование Фурье) может быть также использована при большом количестве поднесущих циклов без влияния переходного процесса либо без сдвига фазы [т. е. во время устойчивой синхронизации TR1, как определено в ИСО/МЭК 14443-2:2010 (пункт 9.2.5), или на устойчивой части SOF, как определено в ИСО/МЭК 14443-3:2011 (пункт 7.1.4)].

(Измененная нумерация. Изм. А4:2012.)

7.2.1.3 Протокол испытаний

Протокол испытаний должен содержать измеренные пиковые амплитуды сигналов верхней и нижней боковых полос на частотах *fc* + *fs* и *fc* – *fs*, а также внешних полей и модуляций.

7.2.2 Уровень EMD PICC и время низкого уровня EMD

7.2.2.1 Цель

Цель данного испытания — определить, что PICC не генерирует электромагнитную помеху, амплитуда V_{EMD} которой выше, чем $V_{E,PICC}$, в течение времени $t_{E,PICC}$, за исключением случаев, определенных в ИСО/МЭК 14443-2:2010/Изм. А1:2011.

Примечание 1 — Время низкого уровня EMD $t_{E,PICC}$ является функцией FDT/TR0, как определено в ИСО/МЭК 14443-2:2010/Изм. А1:2011.

Примечание 2 — Предел EMD $V_{E,PICC}$ представляет собой функцию напряженности поля.

7.2.2.2 Требования к уровню шума

Для того чтобы обеспечить высокий динамический диапазон и достаточную чувствительность к EMD, предварительное испытание на шумовой порог, определенное в 5.5.3, должно быть выполнено перед данным испытанием.

7.2.2.3 Команды для испытания

Испытание на EMD PICC должно быть выполнено для команд по ИСО/МЭК 14443-3. В зависимости от области применения PICC в план испытания должны быть включены дополнительные команды более высокого уровня.

7.2.2.4 Процедура испытания

Данное испытание должно быть проведено как минимум при H_{min} и H_{max}. Используя испытательную установку PCD, выполнить следующие шаги.

a) Отрегулировать радиочастотную мощность, подаваемую генератором сигналов на испытательную антенну PCD на требуемую напряженность поля. Измеряемую калибровочной катушкой.

b) Поместить испытуемую PICC в положение DUT. Радиочастотный передатчик испытательной антенны PCD должен быть повторно отрегулирован на требуемую напряженность поля при необходимости.

c) Провести сброс PICC путем переключения радиочастотного поля в состояния «выключено» и «включено»; затем при необходимости послать передачу последовательности команд, чтобы ввести PICC в исходное состояние испытания (см. G.3.3.2.1 приложения G для типа A и G.4.4.1.1 приложения G для типа B).

d) Послать команду для испытания.

e) Записывать сигнал катушки считывания в течение периода времени как минимум 200 мкс до начала генерации поднесущей PICC. Дополнительно записывать как минимум в течение 50 мкс после обнаружения первой поднесущей для того, чтобы точно определить положение ответа PICC.

f) Определить значение $t_{E,PICC}$ по полученному сигналу: если имеется модуляция PCD на записи, то измерить время между последним нарастающим фронтом модуляции PCD и началом генерации поднесущей PICC и вычислить $t_{E,PICC}$ с помощью формулы, приведенной в ИСО/МЭК 14443-3:2011/Изм. А1:2011. Если модуляции PCD нет на записи, то $t_{E,PICC}$ равно максимальному значению, определенному в ИСО/МЭК 14443-3:2011/Изм. А1:2011.

g) Вычислить мощность сигнала на частотах *fc* + *fs* и *fc* – *fs* как функцию времени согласно 5.5.2.

h) Используя данные, полученные на этапе g), определить время t_{START}, соответствующее половине амплитуды верхней боковой полосы частот во время передачи нарастающего фронта PICC. Проверить, соответствует ли требованиям ИСО/МЭК 14443-2:2010/Изм. А1:2011 амплитуда сигнала в течение времени [$t_{START} - t_{E,PICC}$; $t_{START} - 1/fs$].

i) Повторить этап h) для нижней боковой полосы частот.

j) Повторить этапы с d) по i) для следующей команды испытания.

7.2.2.5 Протокол испытаний

В протоколе испытаний должно быть указано, соответствует ли требованиям ИСО/МЭК 14443-2:2010/Изм. А1:2011 уровень EMD PICC в течение времени $t_{E,PICC}$.

Кроме того, в протоколе испытания должны быть указаны уровни измеренного максимума электромагнитной помехи верхней и нижней боковых полос при частотах $fc + fs$ и $fc - fs$ в течение времени $t_{E,PICC}$. График, показывающий уровни EMD в течение $t_{E,PICC}$, должен быть включен в протокол в случае неуспешного испытания.

(Введены дополнительно. Изм. А2:2012.)

(Измененная редакция. Поправка Cor.1:2013.)

7.2.3 Прием PICC[1]

7.2.3.1 Цель

Цель данного испытания — проверить способность PICC принимать команды от PCD.

7.2.3.2 PICC типа А для скоростей передачи *fc*/128, *fc*/64, *fc*/32 и *fc*/16

(Измененная редакция. Изм. А4:2012.)

7.2.3.2.1 Условия испытаний для скорости передачи *fc*/128

Определены три условия испытаний, характеризующиеся временными соотношениями на границе области временных параметров модулированного сигнала PICC типа А, определенной в ИСО/МЭК 14443-2:2010 (рисунок 4).

Условие 1: максимум $t_1 - t_2$ и максимум t_3, выброса нет.

Условие 2: минимум достижимого (испытательной установкой PCD) значения $t_1 - t_2$ и максимум связанного с ним значения t_3, максимум положительного выброса.

Условие 3: минимум достижимого (испытательной установкой PCD) значения t_3 и максимум связанного с ним значения $t_1 - t_2$, максимум положительного выброса.

Примечание — Амплитуда несущей в конце t_2 должна быть менее 4 %.

Эти три условия должны быть испытаны как минимум при H_{min} и H_{max}, при этом параметр t_1 является максимальным установленным значением при использовании условий 1 и 3 и минимальным установленным значением, если используют условие 2.

7.2.3.2.2 Условия испытаний для скоростей передачи *fc*/64, *fc*/32 и *fc*/16

Определены три условия испытаний, характеризующиеся временными соотношениями на границе области временных параметров модулированного сигнала PICC типа А, определенной в ИСО/МЭК 14443-2:2010 (рисунки 7, 8 и 9).

Условие 1: максимум $t_1 - t_5$ и максимум t_6, положительного выброса нет.

Условие 2: минимум достижимого (испытательной установкой PCD) значения $t_1 - t_5$ и максимум связанного с ним значения t_6, максимум положительного выброса.

Условие 3: минимум достижимого (испытательной установкой PCD) значения t_6 и максимум связанного с ним значения $t_1 - t_5$, максимум положительного выброса.

Эти три условия должны быть испытаны как минимум при H_{min} и H_{max}, при этом:

- параметр *a* является максимальным установленным значением при использовании условия 1 и минимальным достижимым значением для испытательной установки PCD, когда используют условия 2 и 3;
- параметр t_1 является максимальным установленным значением при использовании условий 1 и 3 и минимальным установленным значением, когда используют условие 2.

7.2.3.2.3 Процедура испытаний

При условиях, определенных в 7.2.3.2.1, PICC должна отвечать на REQA посылкой ATQA.

[1] Отсюда и до конца подраздела 7.2 нумерация структурных элементов последовательно изменена из-за включения в текст стандарта дополнительного пункта 7.2.2 (Изм. А2:2012) и дополнительного подпункта 7.2.3.4 (Изм. А4:2012). Изменения к ИСО/МЭК 10373-6:2011 внесены в настоящий и последующие пункты подраздела 7.2 с учетом изменения нумерации. Ссылки на структурные элементы настоящего стандарта изменены с учетом изменившейся нумерации.

PICC, поддерживающая дополнительную скорость передачи *fc*/64, должна работать при условиях, определенных в 7.2.3.2.2, после выбора скорости передачи *fc*/64. Эта PICC должна корректно отвечать на I-блок, переданный при скорости передачи *fc*/64.

PICC, поддерживающая дополнительную скорость передачи *fc*/32, должна работать при условиях, определенных в 7.2.3.2.2, после выбора скорости передачи *fc*/32. Эта PICC должна корректно отвечать на I-блок, переданный при скорости передачи *fc*/32.

PICC, поддерживающая дополнительную скорость передачи *fc*/16, должна работать при условиях, определенных в 7.2.3.2.2, после выбора скорости передачи *fc*/16. Эта PICC должна корректно отвечать на I-блок, переданный при скорости передачи *fc*/16.

7.2.3.3 PICC типа B для скоростей передачи *fc*/128, *fc*/64, *fc*/32 и *fc*/16

7.2.3.3.1 Условия испытаний

Определены три условия испытаний, характеризующиеся временными соотношениями на границе области временных параметров модулированного сигнала PICC, определенной в ИСО/МЭК 14443-2:2010 (пункт 9.1.2).

(Измененная редакция. Изм. A4:2012.)

Условие 1: максимум t_f и максимум t_r, отрицательного и положительного выбросов нет.

Условие 2: минимум достижимого (испытательной установкой PCD) значения t_f и максимум связанного с ним значения t_r, максимум отрицательного и положительного выброса.

Условие 3: минимум достижимого (испытательной установкой PCD) значения t_r и максимум связанного с ним значения t_f, максимум отрицательного и положительного выброса.

Эти три условия должны быть испытаны как минимум при использовании:

- H_{min} и H_{max};
- минимального и максимального коэффициента модуляции *m* для соответствующей приложенной напряженности поля [см. ИСО/МЭК 14443-2:2010 (пункт 9.1.2)].

(Измененная редакция. Изм. A4:2012.)

7.2.3.3.2 Процедура испытаний

При условиях, определенных в 7.2.3.3.1, PICC, работающая при скорости передачи *fc*/128, должна отвечать на REQB посылкой ATQB.

Для каждой дополнительной скорости передачи от PCD к PICC, поддерживаемой PICC, PICC должна работать при условиях, определенных в 7.2.3.3.1, после выбора этой дополнительной скорости передачи. Эта PICC должна отвечать корректно на I-блок, переданный при этой дополнительной скорости передачи.

(Измененная редакция. Изм. A4:2012.)

7.2.3.4 PICC типа A или типа B для скоростей передачи *fc*/8, *fc*/4 и *fc*/2

См. 7.2.3.3.

(Введен дополнительно. Изм. A4:2012.)

7.2.3.5 Протокол испытаний

Протокол испытаний должен подтвердить заданную операцию при обязательной скорости передачи *fc*/128. Для PICC, поддерживающих одну или более дополнительную более высокую скорость передачи, протокол испытаний должен подтвердить заданную операцию на поддерживаемых скоростях передачи. Используемые условия испытаний должны быть указаны в протоколе испытаний.

7.2.4 Резонансная частота PICC (для справки)

7.2.4.1 Цель

Данное испытание может быть использовано для измерения резонансной частоты PICC.

Если две или более PICC разместить в одном и том же возбуждающем поле, то резонансная частота каждой PICC уменьшится.

При разработке каждой PICC выбор резонансной частоты требует тщательного подхода.

ПРЕДУПРЕЖДЕНИЕ — Резонансная частота может зависеть от напряженности поля, используемого во время измерения.

7.2.4.2 Процедура

Резонансную частоту PICC измеряют с помощью анализатора импеданса или сетевого анализатора или индуктивно-резистивно-емкостного измерителя, соединенных с калибровочной катушкой. PICC должна быть размещена на калибровочной катушке на расстоянии 10 мм, при этом оси двух катушек должны совпадать. Резонансная частота — это частота, при которой резистивная часть измеренного комплексного импеданса максимальна.

7.2.4.3 Протокол испытаний

Протокол испытаний должен содержать резонансную частоту PICC и условия измерений.

7.2.5 Максимальный эффект нагрузки PICC

7.2.5.1 Цель

Данное испытание используют для измерения эффекта нагрузки PICC.

7.2.5.2 Процедура испытаний

В зависимости от заявленного класса PICC выбирают:

- соответствующее значение H_{min} согласно ИСО/МЭК 14443-2:2010/Изм. А2:2012 (таблица 2);
- соответствующую эталонную PICC согласно таблице 3 и ее эталонное напряжение V_{load},
- соответствующую испытательную установку PCD согласно таблице 3.

Если PICC не требует какого-либо конкретного класса, определенного в ИСО/МЭК 14443-1:2008/Изм. А1:2012, то для этого испытания используют параметры, испытательное оборудование и схемы для класса 1.

Эффект нагрузки PICC при H_{min} должен быть измерен с помощью испытательной установки PCD. Он не должен превышать эффекта нагрузки эталонной PICC, настроенной на частоту 13,56 МГц и откалиброванной для получения V_{load} на CON3 при H_{min}. Процедура такого метода замещения состоит в следующем.

a) Настроить выбранную эталонную PICC на частоту 13,56 МГц, как описано в 5.4.3.

b) Откалибровать испытательную установку PCD, чтобы создать рабочий режим с H_{min} на калибровочной катушке.

c) Поместить эталонную PICC в положение DUT в испытательной установке PCD. Переключить перемычку J1 в положение b и настроить R2 для получения напряжения постоянного тока V_{load}. Измеряемого на соединителе CON3. В качестве альтернативы перемычка J1 может быть установлена в положение c и приложенное напряжение на CON2 настраивают для получения напряжения постоянного тока V_{load} на соединителе CON3. В обоих случаях режим рабочего поля проверяют, контролируя напряжение на калибровочной катушке, и при необходимости настраивают.

ПРЕДУПРЕЖДЕНИЕ — Значение R2 должно быть в пределах от $R2_{min}$ до $R2_{max}$ согласно таблице 3. Необходимо проверить этот диапазон хотя бы один раз перед использованием альтернативного метода.

d) Убрать эталонную PICC.

e) Расположить испытуемую PICC в положение DUT в испытательной установке PCD.

f) Измерить напряженность поля, наблюдаемую на калибровочной катушке.

Эта напряженность поля должна быть выше, чем H_{min}.

7.2.5.3 Протокол испытаний

Протокол испытаний должен содержать значение измеренной напряженности поля.

(Измененная редакция. Изм. А1:2012.)

8 Испытание по параметрам, установленным в ИСО/МЭК 14443-3 и ИСО/МЭК 14443-4

8.1 Испытания PCD

См. приложение H и приложение I.

8.2 Испытания PICC

См. приложение G.

Приложение А
(обязательное)

Испытательная антенна PCD 1

А.1 Топология испытательной антенны PCD 1, включающая в себя схему согласования импедансов

На рисунках А.1, А.2, А.3 и А.4 показана топология испытательной антенны PCD 1. Рисунки не в масштабе.

(Измененная редакция. Изм. А1:2012.)

Ширина токопроводящей дорожки антенной катушки — 1,8 мм (за исключением сквозных металлизированных отверстий).

Начиная со схемы согласования импедансов, на плате имеются пересечения через каждые 45°.

Печатная плата (PCB): материал FR4, толщина 1,6 мм, двухсторонняя с медным покрытием толщиной 35 мкм.

Примечание 1 — Топология схемы согласования импедансов носит информативный характер.

Примечание 2 — Такие печатные платы и резисторы Rext имеются в различных коммерческих источниках.

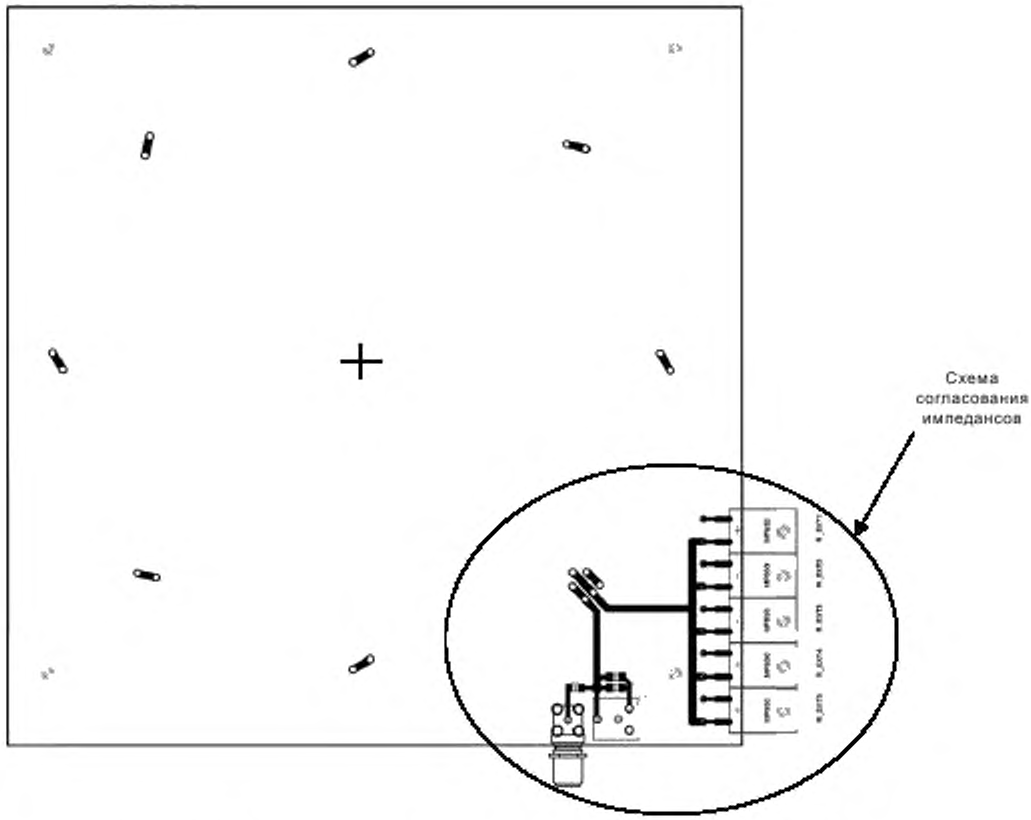


Рисунок А.1 — Топология испытательной антенны PCD 1, включающая в себя схему согласования импедансов для скорости передачи *fc*/128 (вид спереди)

(Измененная редакция. Изм. А1:2012.)

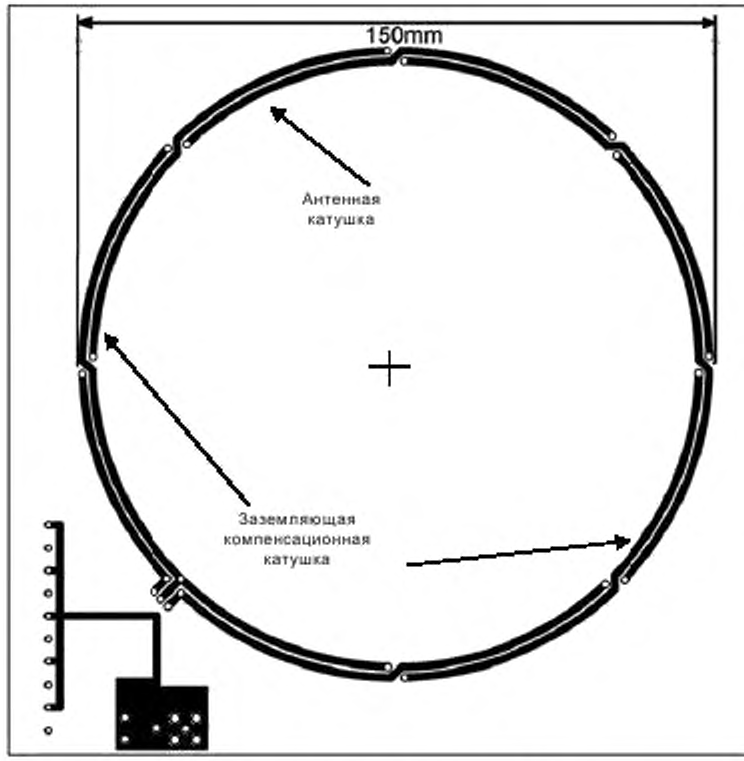


Рисунок А.2 — Топология испытательной антенны PCD 1, включающая в себя схему согласования импедансов для скорости передачи *fc*/128 (вид сзади)

(Измененная редакция. Изм. А1:2012.)

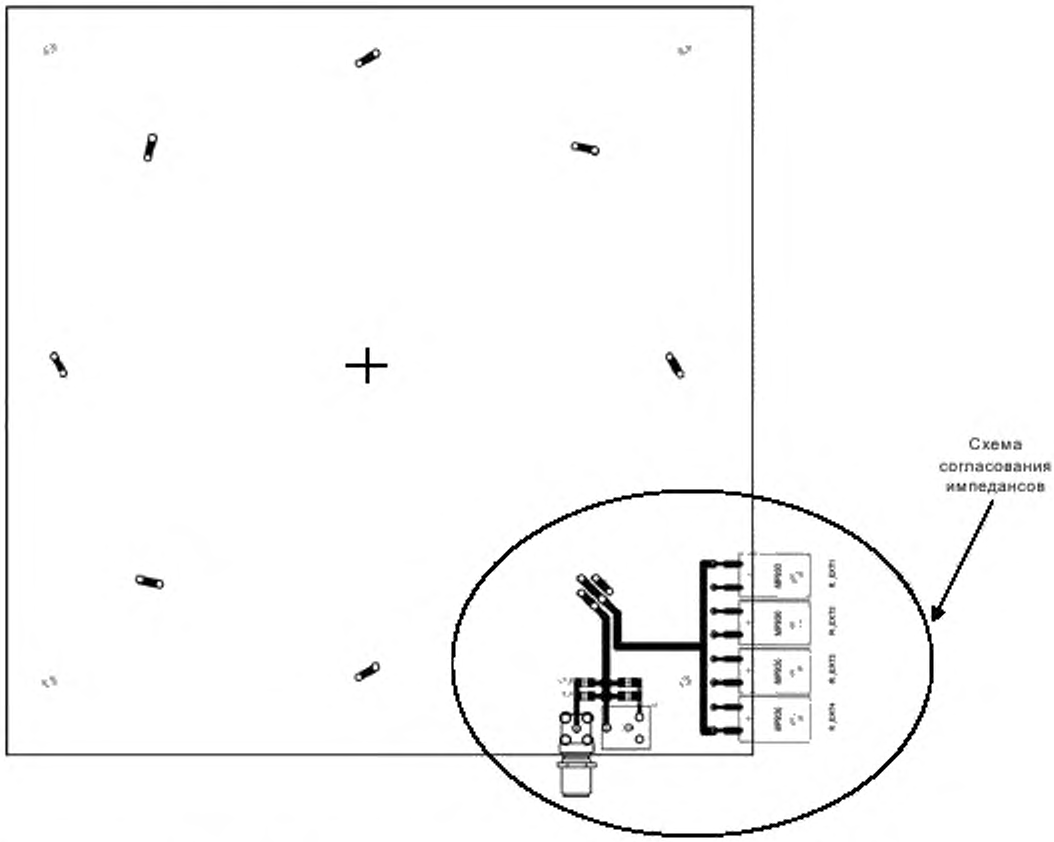


Рисунок А.3 — Топология испытательной антенны PCD 1, включающая в себя схему согласования импедансов для скорости передачи выше *fc*/128 (вид спереди)

(Измененная редакция. Изм. А1:2012 и А4:2012.)

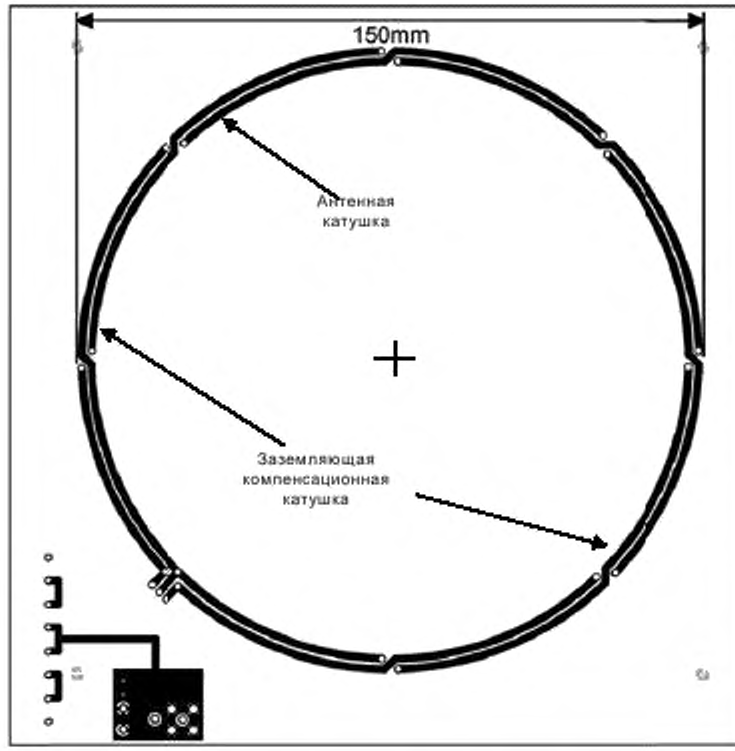


Рисунок А.4 — Топология испытательной антенны PCD 1, включающая в себя схему согласования импедансов для скорости передачи выше *fc*/128 (вид сзади)

(Измененная редакция. Изм. А1:2012 и А4:2012.)

А.2 Схема согласования импедансов

Импеданс антенны (R_{ant}, L_{ant}) настраивают на выходной импеданс генератора сигналов (Z = 50 Ом) с помощью согласующей схемы (см. А.2.1 и А.2.2). Конденсаторы C1, C1a, C1b, C2 и C3 имеют фиксированные значения. Фаза входного импеданса может быть поднастроена конденсатором переменной емкости C4.

Испытательная установка PCD, как определено в 5.3 и в настоящем приложении, предназначена для использования при измерениях, ограниченных во времени, чтобы избежать перегрева отдельных компонентов. Если испытание проходит без остановки, то теплоотдача должна повышаться. Следует принимать меры, чтобы сохранить максимальное напряжение и максимальную теплоотдачу в пределах установленных границ отдельных компонентов.

В настоящем стандарте приведено описание двух схем согласования импедансов: схема согласования импедансов для скорости передачи *fc*/128 и схема согласования импедансов для скорости передачи выше *fc*/128. В таблице А.1 представлены случаи их использования.

(Измененная редакция. Изм. А4:2012.)

Таблица А.1 — Случаи использования схем согласования импедансов

Схема согласования импедансов	Предельное значение напряженности магнитного поля (способность)	Предельное значение скорости передачи
Для скорости передачи *fc*/128	До 12 А/м (среднеквадратичное значение)	Только *fc*/128
Для скорости передачи выше *fc*/128	До 7,5 А/м (среднеквадратичное значение)	Все скорости передачи данных **(Измененная редакция. Изм. А1:2012 и А4:2012.)**

Примечание 1 — Докускаемое отклонение согласованного импеданса антенны составляет ± 5 Ом и ± 10°.

Примечание 2 — Rext располагают на нижней стороне антенной катушки.

Примечание 3 — Диапазоны значений мощности и напряжения включают в себя резервные значения.

Примечание 4 — Линейный усилитель переменной мощности на 50 Ом с низким уровнем искажения должен быть способен эмитировать модуляции типа A и типа B для передачи REQA и REQB, как определено в 7.2.3. Выходная мощность должна быть настраиваемой для получения полей с *H* в диапазоне от 1,5 А/м (среднеквадратичное значение) до 12 А/м (среднеквадратичное значение). Следует принимать во внимание длительность полей, превышающих верхний рабочий диапазон 7,5 А/м (среднеквадратичное значение).

А.2.1 Схема согласования импедансов для скорости передачи *fc*/128

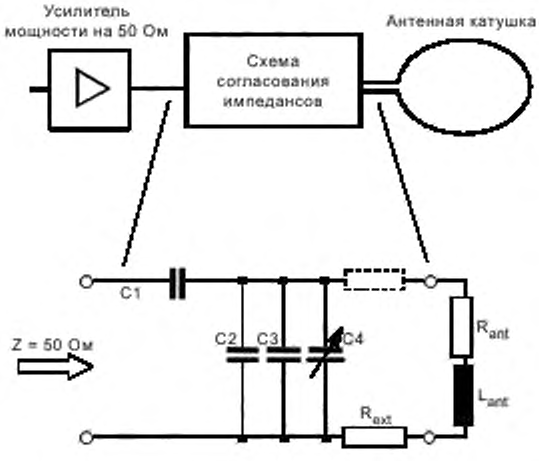


Таблица компонентов			
	Значение	Единица	Примечание
C1	47	пФ	Диапазон напряжения 200 В
C2	180	пФ	Диапазон напряжения 200 В
C3	22	пФ	Диапазон напряжения 200 В
C4	2—27	пФ	Диапазон напряжения 200 В
R_{ext}	0,94	Ом	Диапазон мощности 10 Вт

Рисунок А.5 — Схема согласования импедансов для скорости передачи *fc*/128

Примечание 1 — Rext может быть образован путем соединения пяти резисторов на 4,7 Ом, 2 Вт параллельно.

Примечание 2 — Кроме того, Rext может быть помещен в положении, отмеченном пунктирной линией, как показано на рисунке А.5.

Примечание 3 — Rext может быть на 4 Вт, если максимальное поле достигает 7,5 А/м (среднеквадратичное значение).

Примечание 4 — Паразитная емкость антенны не показана на рисунке А.5.

А.2.2 Схема согласования импедансов для скорости передачи выше *fc*/128
(Измененная редакция. Изм. А4:2012.)

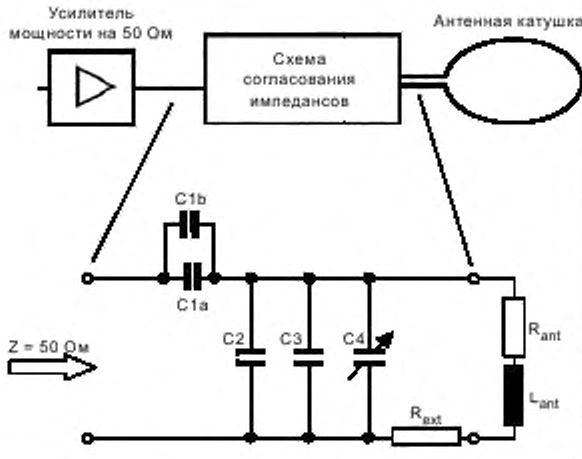


Таблица компонентов

	Значение	Единица	Примечание
C1a	82	пФ	Диапазон напряжения 200 В
C1b	8,2	пФ	Диапазон напряжения 200 В
C2	150	пФ	Диапазон напряжения 200 В
C3	10	пФ	Диапазон напряжения 200 В
C4	2—27	пФ	Диапазон напряжения 200 В
R_{ext}	4,7	Ом	Диапазон мощности 20 Вт

Рисунок А.6 — Схема согласования импедансов для скорости передачи выше *fc*/128

(Измененная редакция. Изм. А4:2012.)

Примечание 1 — Rext может быть образован с помощью параллельной схемы из двух резисторов на 4,7 Ом, 5 Вт, соединенных последовательно.

Примечание 2 — Rext должен быть помещен на стороне GND антенны, как показано на рисунке.

Примечание 3 — Паразитная емкость антенны не показана на рисунке А.6.

А.3 Испытательная антенна PCD 2

А.3.1 Топология испытательной антенны PCD 2, включающая схему согласования импедансов

На рисунках А.7 и А.8 показана топология испытательной антенны PCD 2. Рисунки не в масштабе.

Ширина токопроводящей дорожки антенной катушки — 1,8 мм (за исключением сквозных металлизированных отверстий).

Начиная со схемы согласования импедансов, на плате имеются пересечения через каждые 45°.

Печатная плата (PCB): материал FR4, толщина 1,6 мм, двухсторонняя с медным покрытием толщиной 35 мкм.

Примечание 1 — Топология и положение схемы согласования импедансов носит информативный характер. Внешние размеры PCB даны для информации.

Примечание 2 — Такие печатные платы и резисторы Rext имеются в различных коммерческих источниках.

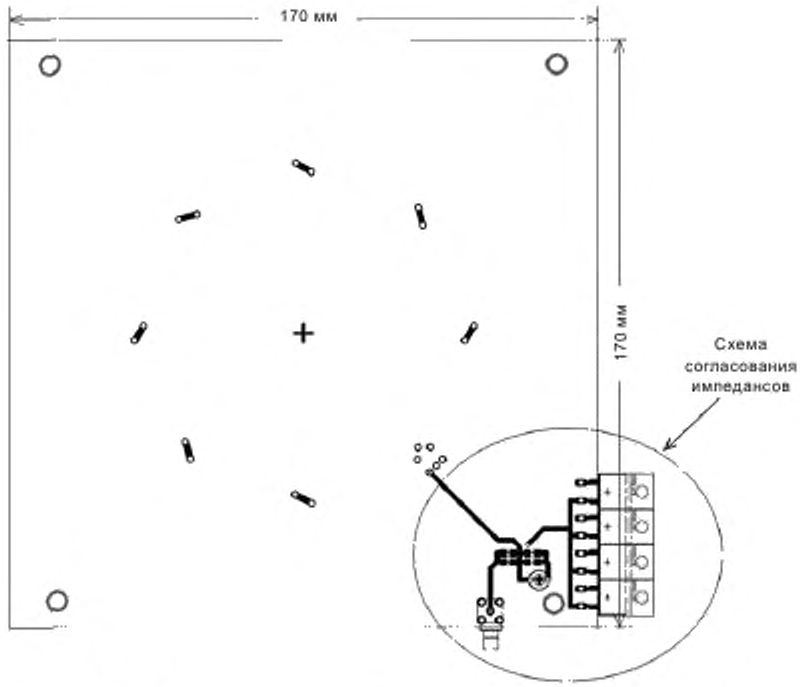


Рисунок А.7 — Топология испытательной антенны PCD 2, включающая в себя схему согласования импедансов (вид спереди)

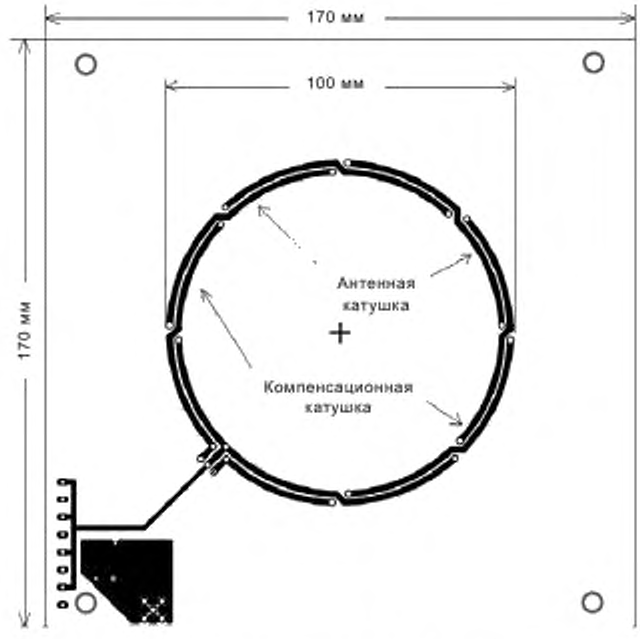


Рисунок А.8 — Топология испытательной антенны PCD 2, включающая в себя схему согласования импедансов (вид сзади)

А.3.2 Схема согласования импедансов 2

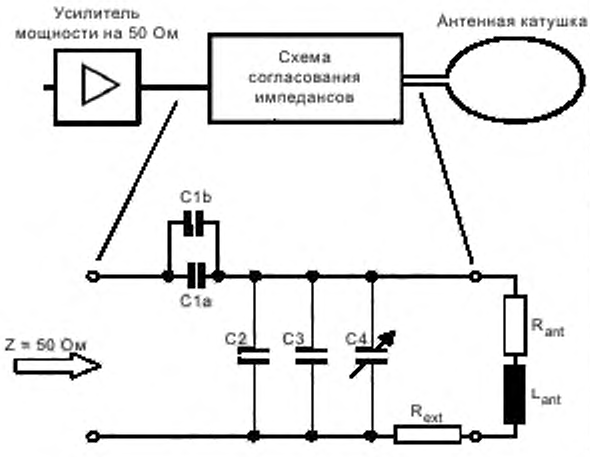


Таблица компонентов			
	Значение	Единица	Примечание
C1a	100	пФ	Диапазон напряжения 200 В
C1b	12	пФ	Диапазон напряжения 200 В
C2	270	пФ	Диапазон напряжения 200 В
C3	18	пФ	Диапазон напряжения 200 В
C4	2—27	пФ	Диапазон напряжения 200 В
R_{ext}	2,7	Ом	Диапазон мощности 20 Вт

Рисунок А.9 — Схема согласования импедансов 2

Примечание 1 — Rext может быть образован или с помощью параллельной схемы из двух равных ветвей, имеющих два резистора на 2,7 Ом мощностью 5 Вт каждый, или параллельной схемы из 10 Ом, 10 Ом, 10 Ом и 15 Ом мощностью 5 Вт.

Примечание 2 — Rext должен быть помещен на стороне GND антенны, как показано на рисунке.

Примечание 3 — Паразитная емкость антенны не показана на рисунке А.9.

(Введены дополнительно. Изм. А1:2012.)

Приложение В
(справочное)

Настройка испытательной антенны PCD

На рисунках В.1 и В.2 показаны два этапа простой фазовой настройки для согласования импеданса антенны и задающего генератора. После такой двухэтапной настройки генератор сигналов должен быть непосредственно подсоединен к выходу антенны для испытаний.

Этап 1

Резистор высокого класса точности на 50 Ом ± 1 % (например, BNC-резистор на 50 Ом) вставляют в сигнальную линию между выходом генератора сигналов и антенным соединителем. Два пробника осциллографа подсоединяют к обеим сторонам последовательного эталонного резистора. На осциллографе отображается фигура Лиссажу, когда он настроен для представления сигнала по осям Y и X. Установки параметров генератора сигналов:

- форма сигнала — синусоидальная;
- частота — 13,56 МГц;
- амплитуда — от 2 до 5 В (среднеквадратичные значения).

Выход заканчивается вторым резистором высокой точности на 50 Ом ± 1 % (например, BNC-оконечный резистор на 50 Ом). Пробник, включенный параллельно с выходным соединителем, имеет малую паразитную емкость C_{probe}. Калибровочный конденсатор C_{cal}, включенный параллельно эталонному резистору, компенсирует эту емкость пробника при условии $C_{cal} = C_{probe}$. Емкость пробника считают компенсированной при отображении полностью замкнутой фигуры Лиссажу.

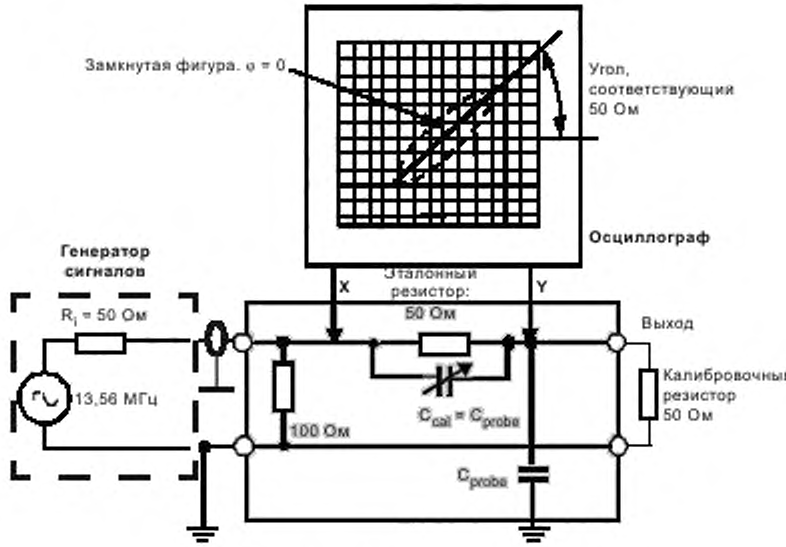


Рисунок В.1 — Схема настройки (этап 1)

Примечание — Заземляющий кабель должен быть проложен близко к пробнику, чтобы избежать напряжений, наведенных магнитным полем.

Этап 2

Используя те же самые значения, что на этапе 1, на втором этапе схему согласования подключают к выходу антенны. Конденсатор С4 на антенной плате используют для настройки фазы на ноль.

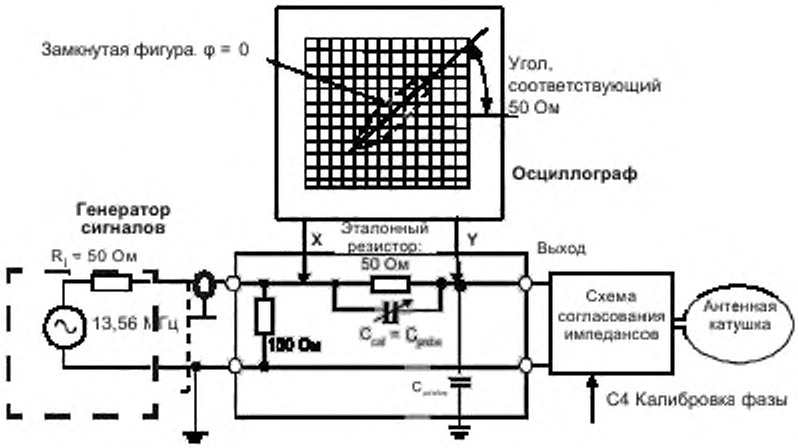


Рисунок В.2 — Схема настройки (этап 2)

Приложение С
(обязательное)

Катушки считывания

С.1 Топология катушек считывания

С.1.1 Топология катушек считывания 1

На рисунке С.1 показана топология катушек считывания 1.

(Измененная редакция. Изм. А1:2012.)

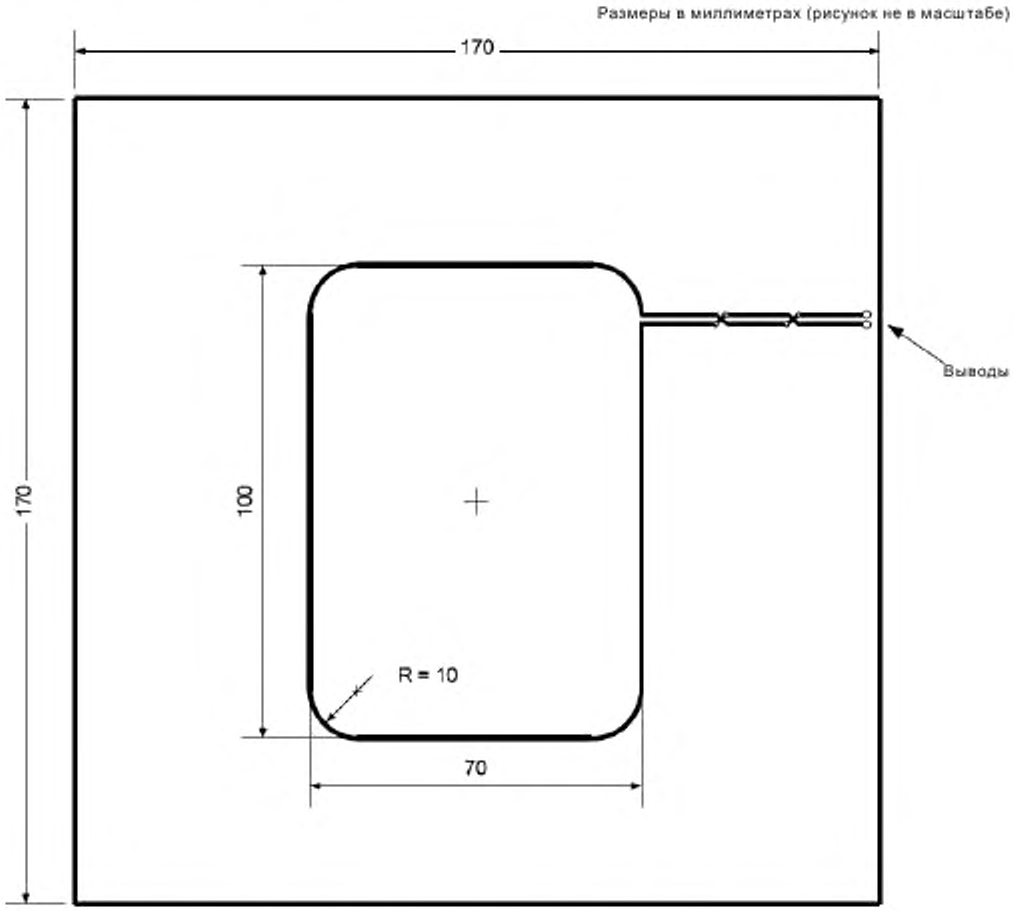


Рисунок С.1 — Топология катушек считывания 1 (*a* и *b*)

(Измененная редакция. Изм. А1:2012.)

Ширина токопроводящих дорожек катушек считывания 0,5 мм с допускаемым отклонением ± 20 % (за исключением сквозных металлизированных отверстий). Размеры катушек относятся к наружным размерам.

Печатная плата (PCB): материал FR4, толщина 1,6 мм, двухсторонняя с медным покрытием толщиной 35 мкм.

Примечание — Такие печатные платы имеются в различных коммерческих источниках.

С.1.2 Топология катушек считывания 2

На рисунке С.2 показана топология катушек считывания 2. Рисунок не в масштабе.

Ширина токопроводящих дорожек катушек считывания 0,5 мм с допускаемым отклонением ± 20 % (за исключением сквозных металлизированных отверстий). Размеры катушек относятся к наружным размерам.

Печатная плата (PCB): материал FR4, толщина 1,6 мм, двухсторонняя с медным покрытием толщиной 35 мкм.

Примечание — Такие печатные платы имеются в различных коммерческих источниках.

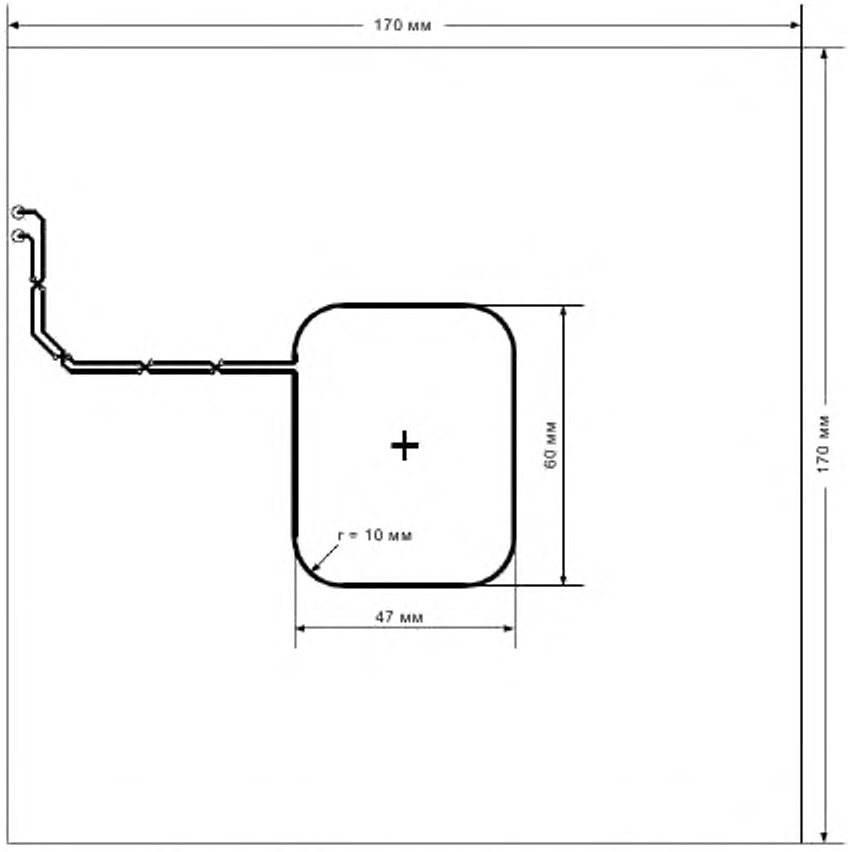


Рисунок С.2 — Топология катушек считывания 2 (*a* и *b*)

(Введены дополнительно. Изм. А1:2012.)

С.2 Установка катушек считывания

На рисунке С.3[1] показана установка катушек считывания.

[1] Номер рисунка изменен из-за включения в текст стандарта дополнительного рисунка С.2 (Изм. А1:2012).

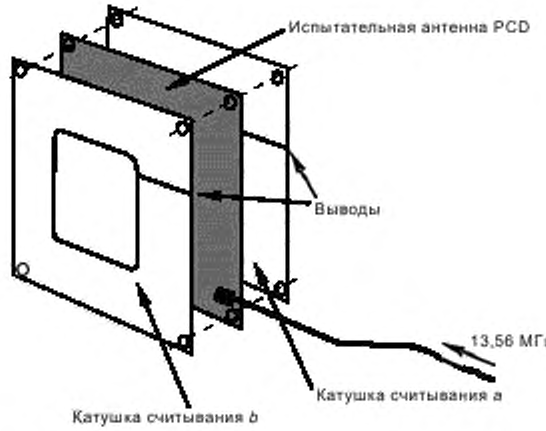


Рисунок С.3 — Установка катушек считывания

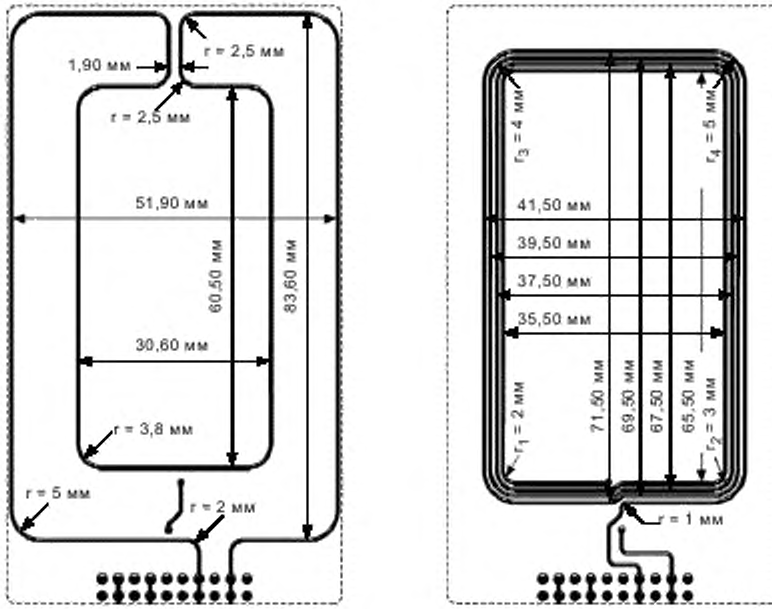
Приложение D
(обязательное)

Эталонные PICC

D.1 Топология катушек эталонной PICC 1

На рисунке D.1 определена топология воспринимающей катушки и главной катушки эталонной PICC 1.

(Измененная редакция. Изм. А1:2012.)



Вид спереди, воспринимающая катушка Вид сзади, главная катушка

Рисунок D.1 — Топология воспринимающей катушки и главной катушки эталонной PICC 1

(Измененная редакция. Изм. А1:2012.)

Размеры считать от середины токопроводящей дорожки (рисунок не в масштабе).

Воспринимающая катушка и главная катушка должны быть концентричны.

Ширина токопроводящих дорожек обеих катушек и расстояние между ними должны составлять 0,5 мм с допускаемым отклонением ± 20 %.

Печатная плата (PCB): материал FR4, толщина 0,76 мм с допускаемым отклонением ± 10 %, двухсторонняя с медным покрытием толщиной 35 мкм.

Примечание 1 — При частоте 13,56 МГц индуктивность главной катушки L1 составляет 2,3 мкГн ± 10 %, а сопротивление — 1,8 Ом ± 10 %.

Примечание 2 — При частоте 13,56 МГц индуктивность воспринимающей катушки L2 составляет 375 нГн ± 10 %, а сопротивление — приблизительно 0,65 Ом ± 10 %.

Примечание 3 — Такие печатные платы имеются в различных коммерческих источниках.

D.2 Топология катушек эталонной PICC 2

На рисунке D.2 определена топология воспринимающей катушки и главной катушки эталонной PICC 2.

Размеры считать от середины токопроводящей дорожки (рисунок не в масштабе).

Размеры главной катушки: 75 × 24 мм (наружные размеры).

Воспринимающая катушка и главная катушка должны быть концентричны.

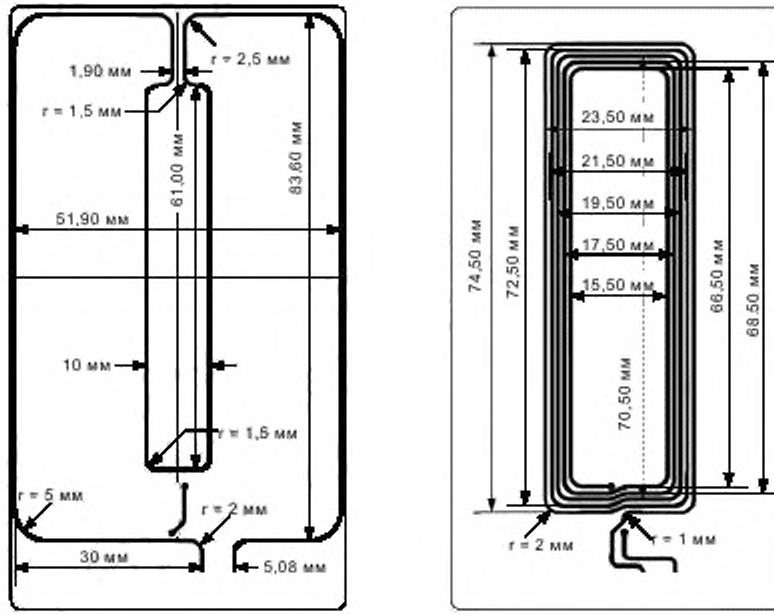
Ширина токопроводящих дорожек обеих катушек и расстояние между ними должны составлять 0,5 мм с допускаемым отклонением ± 20 %.

Радиусы закругления всех углов главной катушки должны составлять 2 мм.

Печатная плата (PCB): материал FR4, толщина 0,76 мм с допускаемым отклонением ± 10 %, двухсторонняя с медным покрытием толщиной 35 мкм.

Примечание 1 — При частоте 13,56 МГц индуктивность главной катушки L1 составляет 2,4 мкГн ± 10 %, а сопротивление — 1,9 Ом ± 10 %.

Примечание 2 — При частоте 13,56 МГц индуктивность воспринимающей катушки L2 составляет 417 нГн ± 10 %, а сопротивление — приблизительно 0,8 Ом ± 10 %.



Вид спереди, воспринимающая катушка Вид сзади, главная катушка

Рисунок D.2 — Топология воспринимающей катушки и главной катушки эталонной PICC 2

D.3 Топология катушек эталонной PICC 3

На рисунке D.3 определена топология воспринимающей катушки и главной катушки эталонной PICC 3.

Размеры считать от середины токопроводящей дорожки (рисунок не в масштабе).

Размеры главной катушки: 46 × 32 мм (наружные размеры).

Воспринимающая катушка и главная катушка должны быть концентричны.

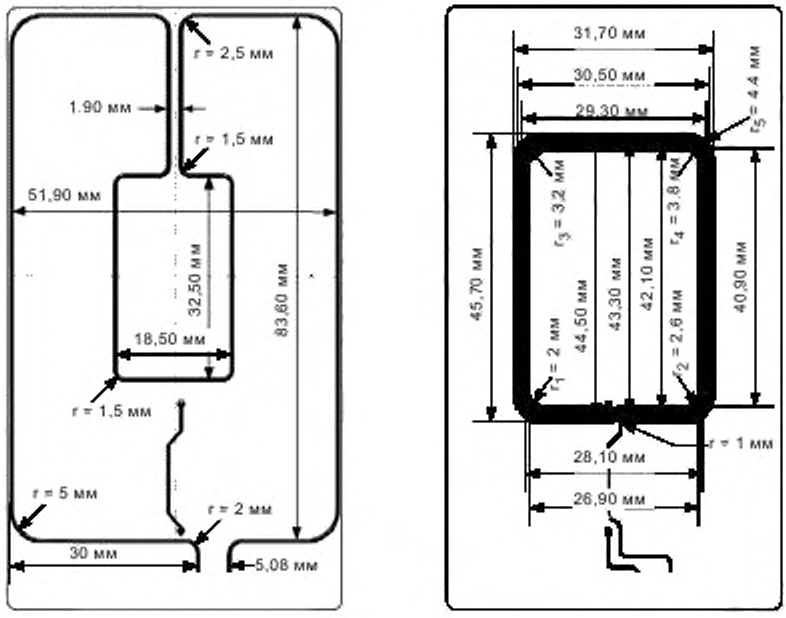
Ширина токопроводящей дорожки воспринимающей катушки должна составлять 0,5 мм с допускаемым отклонением ± 20 %.

Ширина токопроводящих дорожек главной катушки и расстояние между ними должны составлять 0,3 мм с допускаемым отклонением ± 20 %.

Печатная плата (PCB): материал FR4, толщина 0,76 мм с допускаемым отклонением ± 10 %, двухсторонняя с медным покрытием толщиной 35 мкм.

Примечание 1 — При частоте 13,56 МГц индуктивность главной катушки L1 составляет 2,39 мкГн ± 10 %, а сопротивление — 2,18 Ом ± 10 %.

Примечание 2 — При частоте 13,56 МГц индуктивность воспринимающей катушки L2 составляет 405 нГн ± 10 %, а сопротивление — приблизительно 0,76 Ом ± 10 %.

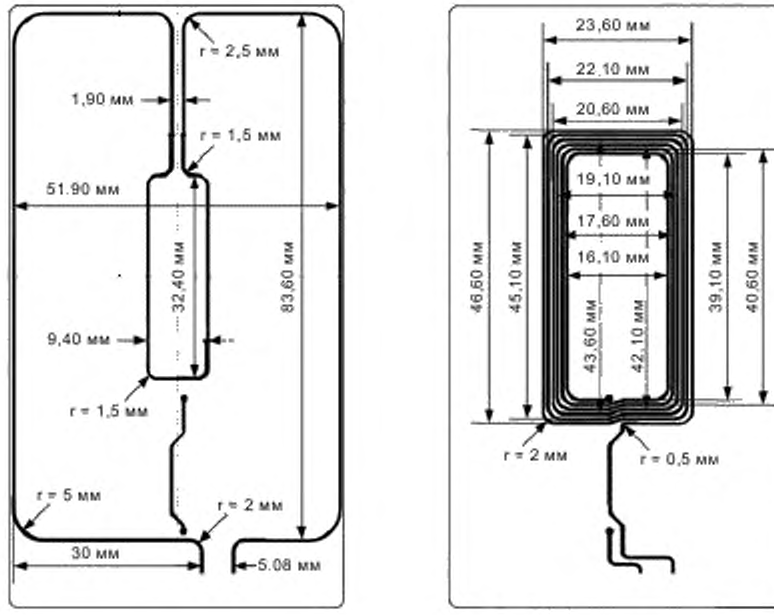


Вид спереди, воспринимающая катушка Вид сзади, главная катушка

Рисунок D.3 — Топология воспринимающей катушки и главной катушки эталонной PICC 3

D.4 Топология катушек эталонной PICC 4

На рисунке D.4 определена топология воспринимающей катушки и главной катушки эталонной PICC 4.



Вид спереди, воспринимающая катушка Вид сзади, главная катушка

Рисунок D.4 — Топология воспринимающей катушки и главной катушки эталонной PICC 4

Размеры считать от середины токопроводящей дорожки (рисунок не в масштабе).

Размеры главной катушки: 47 × 24 мм (наружные размеры).

Воспринимающая катушка и главная катушка должны быть концентричны.

Ширина токопроводящей дорожки воспринимающей катушки должна быть 0,5 мм с допускаемым отклонением ± 20 %.

Радиусы закругления всех углов главной катушки должны составлять 2 мм.

Ширина токопроводящих дорожек главной катушки должна составлять 0,4 мм, а расстояние между ними — 0,35 мм с допускаемым отклонением ± 20 %.

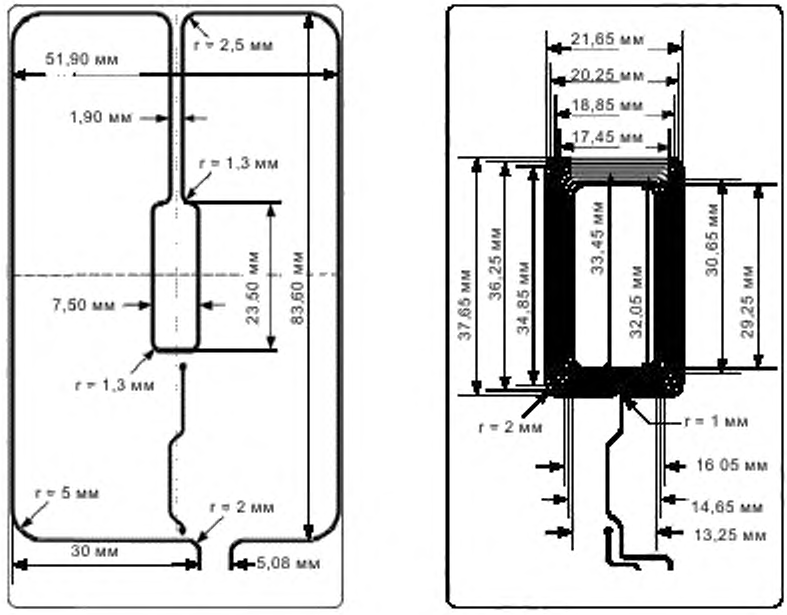
Печатная плата (PCB): материал FR4, толщина 0,76 мм с допускаемым отклонением ± 10 %, двухсторонняя с медным покрытием толщиной 35 мкм.

Примечание 1 — При частоте 13,56 МГц индуктивность главной катушки L1 составляет 2,3 мкГн ± 10 %, а сопротивление — 1,8 Ом ± 10 %.

Примечание 2 — При частоте 13,56 МГц индуктивность воспринимающей катушки L2 составляет 390 нГн ± 10 %, а сопротивление — приблизительно 0,7 Ом ± 10 %.

D.5 Топология катушек эталонной PICC 5

На рисунке D.5 определена топология воспринимающей катушки и главной катушки эталонной PICC 5.



Вид спереди, воспринимающая катушка Вид сзади, главная катушка

Рисунок D.5 — Топология воспринимающей катушки и главной катушки эталонной PICC 5

Размеры считать от середины токопроводящей дорожки (рисунок не в масштабе).

Размеры главной катушки: 38 × 22 мм (наружные размеры).

Воспринимающая катушка и главная катушка должны быть концентричны.

Ширина токопроводящей дорожки воспринимающей катушки должна быть 0,5 мм с допускаемым отклонением ± 20 %.

Радиусы закругления всех углов главной катушки должны составлять 2 мм.

Ширина токопроводящих дорожек главной катушки и расстояние между ними должны составлять 0,35 мм с допускаемым отклонением ± 20 %.

Печатная плата (PCB): материал FR4, толщина 0,76 мм с допускаемым отклонением ± 10 %, двухсторонняя с медным покрытием толщиной 35 мкм.

Примечание 1 — При частоте 13,56 МГц индуктивность главной катушки L1 составляет 2,4 мкГн ± 10 %, а сопротивление — 1,9 Ом ± 10 %.

Примечание 2 — При частоте 13,56 МГц индуктивность воспринимающей катушки L2 составляет 380 нГн ± 10 %, а сопротивление — приблизительно 0,7 Ом ± 10 %.

D.6 Топология катушек эталонной PICC 6

На рисунке D.6 определена топология воспринимающей катушки и главной катушки эталонной PICC 6.

Размеры считать от середины токопроводящей дорожки (рисунок не в масштабе).

Размеры главной катушки: 25 × 20 мм (наружные размеры).

Воспринимающая катушка и главная катушка должны быть концентричны.

Ширина токопроводящей дорожки воспринимающей катушки должна быть 0,5 мм с допускаемым отклонением ± 20 %.

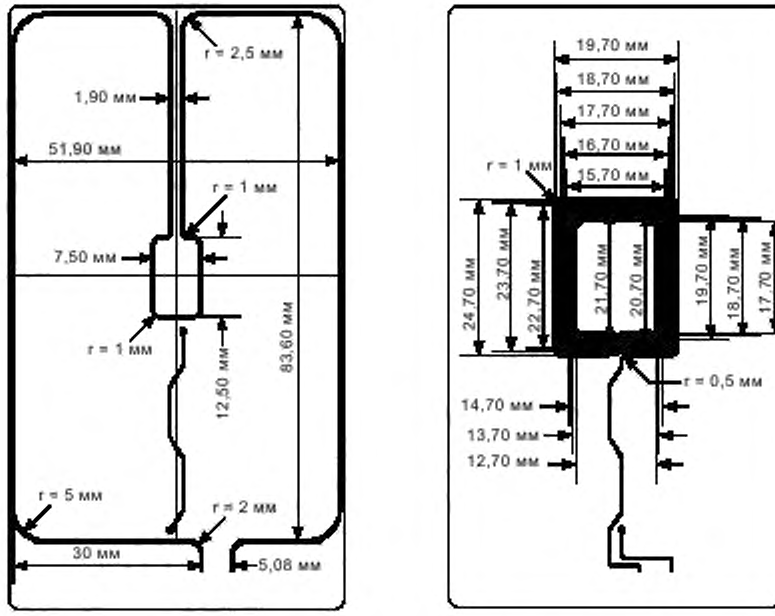
Радиусы закругления всех углов главной катушки должны составлять 1 мм.

Ширина токопроводящих дорожек главной катушки должна быть 0,3 мм, а расстояние между ними — 0,2 мм с допускаемым отклонением ± 20 %.

Печатная плата (PCB): материал FR4, толщина 0,76 мм с допускаемым отклонением ± 10 %, двухсторонняя с медным покрытием толщиной 35 мкм.

Примечание 1 — При частоте 13,56 МГц индуктивность главной катушки L1 составляет 2,3 мкГн ± 10 %, а сопротивление — 2,2 Ом ± 10 %.

Примечание 2 — При частоте 13,56 МГц индуктивность воспринимающей катушки L2 составляет 370 нГн ± 10 %, а сопротивление — приблизительно 0,7 Ом ± 10 %.



Вид спереди, воспринимающая катушка Вид сзади, главная катушка

Рисунок D.6 — Топология воспринимающей катушки и главной катушки эталонной PICC 6

(Введены дополнительно. Изм. А1:2012.)

Приложение Е
(обязательное)

Инструментарий для анализа формы сигнала и определения коэффициента модуляции

Е.1 Обзор

Принцип работы инструментария для анализа формы сигнала и определения коэффициента модуляции показан на рисунке Е.1.



Рисунок Е.1 — Блок-схема инструментария для анализа формы сигнала и определения коэффициента модуляции

Каждый блок описан в последующих разделах.

Е.2 Выборка дискретных данных

Осциллограф, используемый для фиксации сигнала, должен соответствовать требованиям, установленным в 5.1.1.

Е.2.1 Выборка дискретных значений для скоростей передачи *fc*/128, *fc*/64, *fc*/32 и *fc*/16

Данные о времени и напряжении одного импульса модуляции (см. рисунок Е.2) должны быть перенесены на подходящий компьютер.

(Измененная редакция. Изм. А4:2012.)

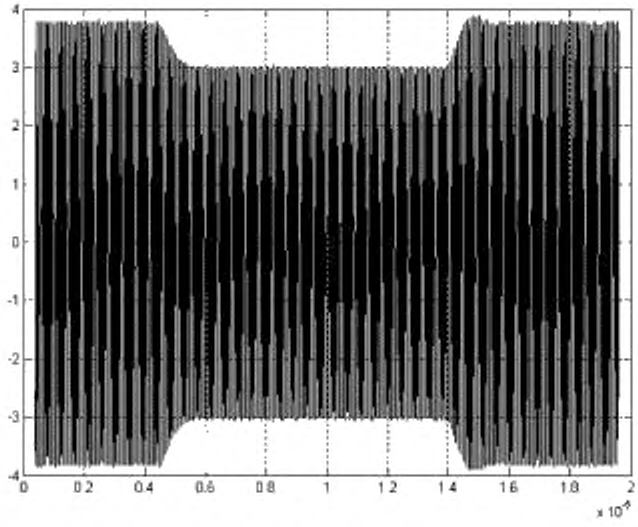


Рисунок Е.2 — Импульс модуляции

Е.2.2 Выборка дискретных значений для скоростей передачи *fc*/8, *fc*/4 и *fc*/2

Данные о времени и напряжении кадра PCD, содержащего короткие и длинные импульсы модуляции [предпочтительно завершенная команда S(DESELECT)], показанные на рисунке Е.3, с по меньшей мере 20 периодами несущей до первого и после последнего импульса модуляции, должны быть перенесены на подходящий компьютер.

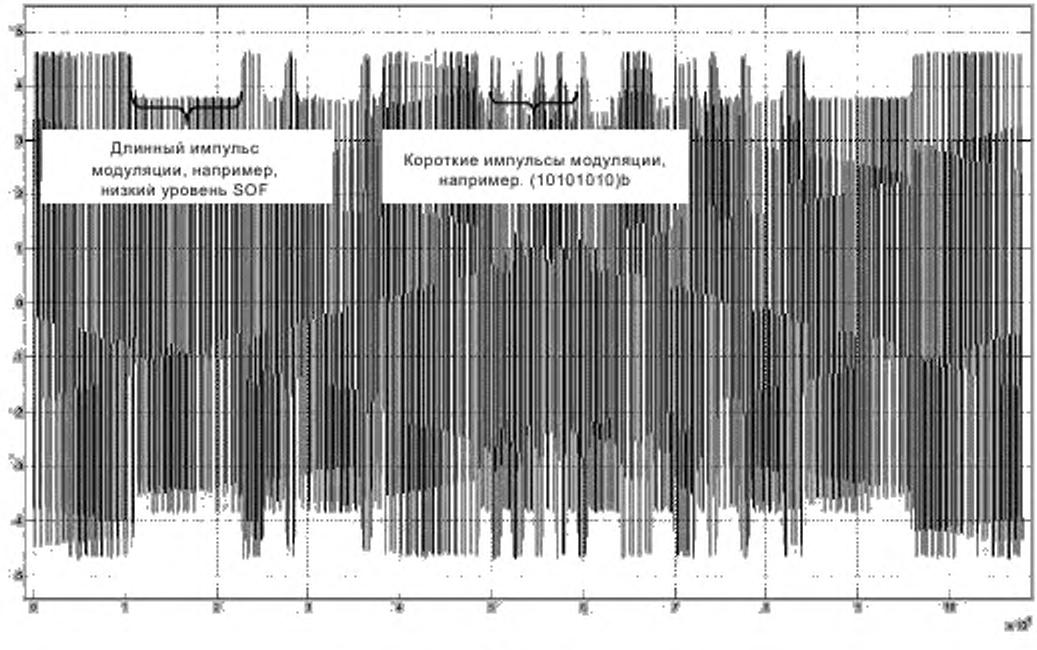


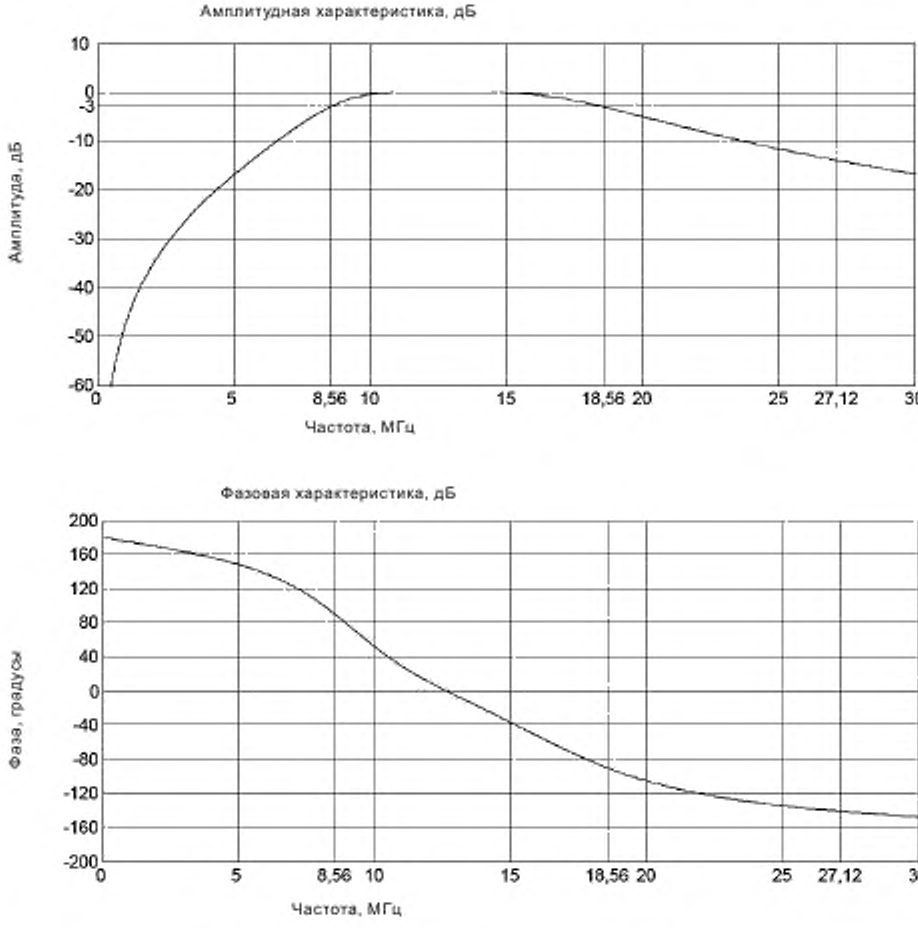
Рисунок Е.3 — Импульсы модуляции

(Введены дополнительно. Изм. А4:2012.)

Е.3 Фильтрация

Е.3.1 Фильтрация для скоростей передачи *fc*/128, *fc*/64, *fc*/32 и *fc*/16

Для фильтрации составляющих постоянного тока и высших гармоник используют полосовой фильтр Баттерворта 4-го порядка с центральной частотой 13,56 МГц и неравномерностью 3 дБ в полосе пропускания шириной 10 МГц. Характеристики фильтра показаны на рисунке Е.3.



(Измененная редакция. Изм. А4:2012.)

Е.3.2 Фильтрация для скоростей передачи *fc*/8, *fc*/4 и *fc*/2

Для фильтрации составляющих постоянного тока и высших гармоник используют полосовой фильтр Баттерворта 4-го порядка с центральной частотой 13,56 МГц и неравномерностью 3 дБ в полосе пропускания шириной 15 МГц.

(Введен дополнительно. Изм. А4:2012.)

Е.4 Генерация огибающей

К отфильтрованному сигналу необходимо применить преобразование Гильберта, и модуль этого комплексного преобразования будет представлять собой огибающую сигнала.

Е.5 Сглаживание огибающей

Е.5.1 Сглаживание огибающей для скоростей передачи *fc*/128, *fc*/64, *fc*/32 и *fc*/16

Огибающая сигнала должна быть сглажена с помощью фильтра скользящего среднего, а период фильтрации должен быть равен одному периоду несущей. Сглаженная огибающая сигнала показана на рисунке Е.4.

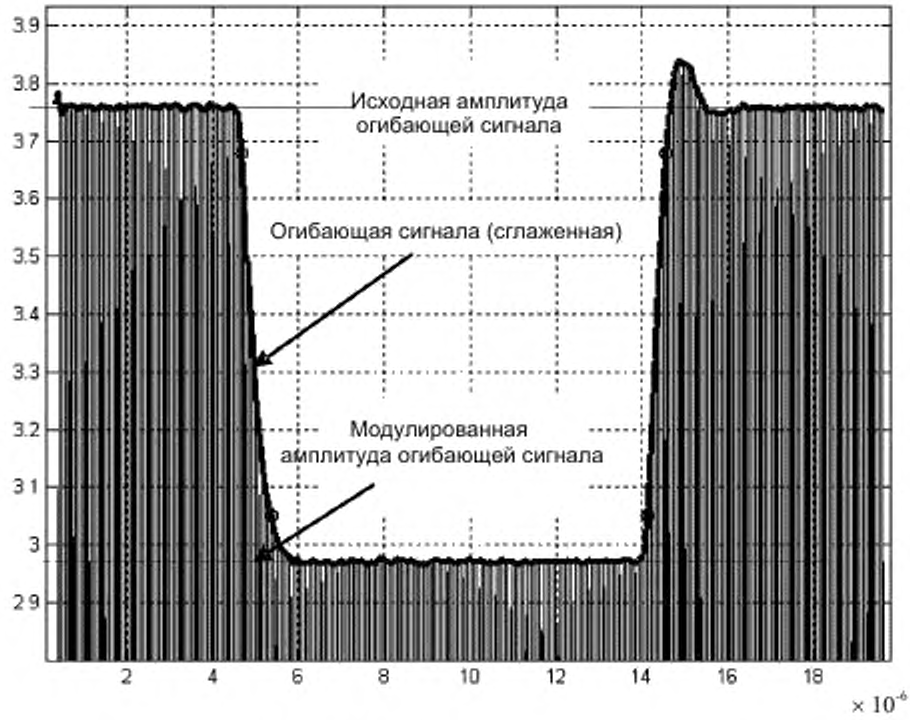


Рисунок Е.4 — Сглаживание огибающей

(Измененная редакция. Изм. А4:2012.)

Е.5.2 Сглаживание огибающей для скоростей передачи *fc*/8, *fc*/4 и *fc*/2

К огибающей сигнала сглаживание не применяют.

(Введен дополнительно. Изм. А4:2012.)

Е.6 Определение коэффициента модуляции

Исходная и модулированная амплитуда огибающей сигнала должны быть определены с помощью вычисления гистограммы сглаженной огибающей сигнала. Наиболее частые значения соответствуют исходной и модулированной амплитуде огибающей сигнала. Для модулированных сигналов типа А с помощью гистограммы определяют только исходную амплитуду огибающей сигнала.

Для скоростей передачи *fc*/8, *fc*/4 и *fc*/2 минимальное значение коэффициента модуляции *m* следует определять по полному кадру PCD (см. рисунок Е.5). Кадр PCD должен содержать (10101010)b.

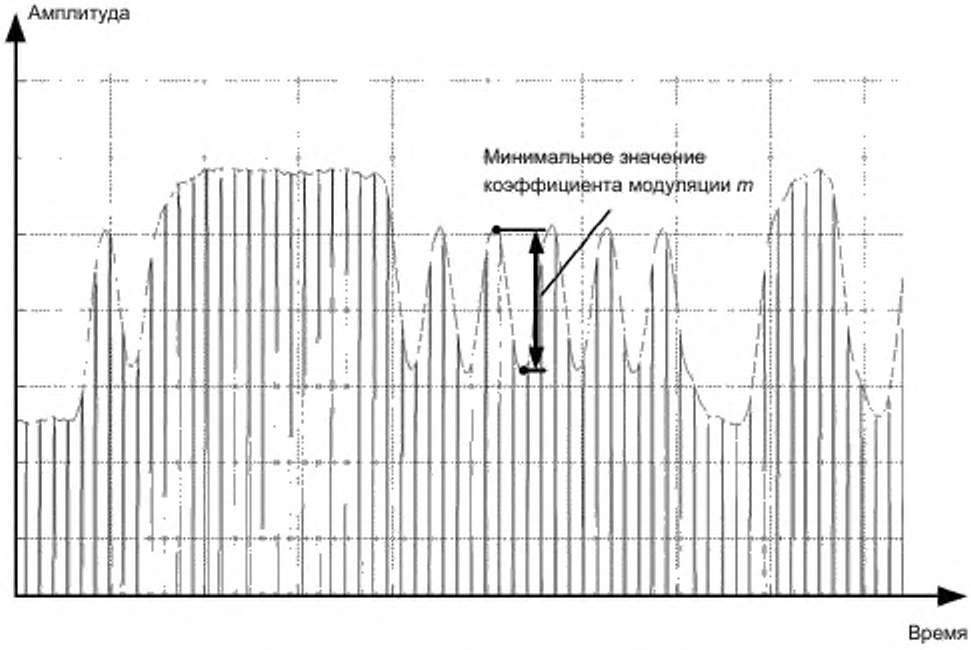


Рисунок Е.5 — Минимальное значение коэффициента модуляции *m*

(Введены дополнительно. Изм. А4:2012.)

Е.7 Определение временных соотношений

Время нарастания и время спада определяют в соответствии с ИСО/МЭК 14443-2:2010.

Для скоростей передачи *fc*/8, *fc*/4 и *fc*/2 временные соотношения следует определять в положениях длинного импульса модуляции, например, время t_f на переходе к низкому уровню SOF и время t_r на переходе к высокому уровню EOF.

(Введен дополнительно. Изм. А4:2012.)

Е.8 Определение положительного и отрицательного выбросов

Уже сглаженная огибающая сигнала должна быть дополнительно сглажена с помощью фильтра скользящего среднего по трем периодам несущей до определения значений положительного и отрицательного выбросов в соответствии с определениями по ИСО/МЭК 14443-2:2010.

Е.9 Программа инструментария для анализа формы сигнала и определения коэффициента модуляции (для справки)

Следующая программа, написанная на языке С, дает пример реализации инструментария для анализа формы сигнала и определения коэффициента модуляции.

Реализация алгоритма на языке С состоит из шести различных файлов, которые должны быть помещены в одну папку.

E.9.1 structures.h

```
/*****************************************************************/
/*structures.h                                                   */
/*This code contains the structures to save important results    */
/*****************************************************************/
#ifndef STRUCTURES_H
#define STRUCTURES_H

typedef struct esl
{
      double volt;
      double time;
      struct esl *sig;
}ESL;

typedef struct times
{
      double tf; // Type B
      double tr; // Type B
      double b; // Type B
      double bVHBR; //Bit rates of fc/8, fc/4 and fc/2
```

(Введено дополнительно. Изм. А4:2012.)

```
      Double trstartind; // Type B
      double trendind; // Type B
      double tfstartind; // Type B
      double tfendind; // Type B

      double t1; // Type A (all bit rates)
      double t1startind; // Type A (all bit rates)
      double t1start; // Type A (all bit rates)
      double t1endind; // Type A (all bit rates)

      double t2; // Type A
      double t2startind; // Type A
      double t2start; // Type A
      double t3; // Type A
      double t3end; // Type A
      double t3endind; // Type A
      double t4; // Type A
      double t4endind; // Type A

      double t5; // Type A (higher bit rates)
      double t5startind; // Type A (higher bit rates)
      double t6; // Type A (higher bit rates)
      double t6end; // Type A (higher bit rates)
      double t6endind; // Type A (higher bit rates)
      double a; // Type A (higher bit rates)
      double tploone; //Type A (higher bit rates)
}TIMES;

typedef struct shootreader
{
      double shooting;
      double shooting_b;
      double hf_reader;
      double hr_reader;
      double above;
      double above_b;
}SHOOTREADER;

#endif
```

E.9.2 fftrm.h

```
/*****************************************************************/
/*fftrm.h                                                        */
/*This is the header file for fftrm.c                            */
/*****************************************************************/

#ifndef FFTRM_H
#define FFTRM_H

#define RE(z) ((z).r)
#define IM(z) ((z).i)

typedef float real;
typedef double doublereal;
typedef struct { real r, i; } complex;
typedef struct { doublereal r, i; } doublecomplex;

int zffts (int debug, doublecomplex *X, int M);
int ziffts (int debug, doublecomplex *X, int M);
void zfftrmc(doublecomplex *X, int M, int P, float D);
void rmpo (int *rv, int *rvp );

#endif
```

E.9.3 fftrm.c

```
/*****************************************************************/
/*fftrm.c                                                        */
/*This code contains the necessary function for Fourier and      */
/*inverse Fourier transformation                                 */
/*****************************************************************/

#include <stdio.h>
#include <math.h>
#include <malloc.h>
#include "fftrm.h"

#ifndef M_PI
#define M_PI 3.14159265358979323846264338327950
#endif

float *WR;
float *WI;

doublereal *DWR;
doublereal *DWI;

void rmpo( int *rv, int *rvp )
{
     int value_h;
     int n;

     n = 1;
     *rvp = -1;
     value_h = 1;

     while ( value_h > 0 )
     {
          value_h = *rv - n;
          (*rvp)++;
          n += n;
     }
}
```

```
void zfftrmc( doublecomplex *X, int M, int P, float D )
{
      int MV2,MM1,J,I,K,L,LE,LE1,IP,IQ,IND,IND1,R;
      int I1,J1;
      float A,B;
      float WCOS,WSIN;
      float VR,VI;
      float ARG;

      static int IPOTC;
      static float DALT;

      DWR = (doublereal *)calloc(M,sizeof(doublereal));
      DWI = (doublereal *)calloc(M,sizeof(doublereal));

      /* if (IPOTC == P & D == DALT) goto warmstart; */

      IPOTC = P;
      DALT = (float)D;
      LE = 1;
      IND = 0;

      for (L=1;L<=P;L++)
      {
          LE1 = LE;
          LE = LE*2;
          DWR[IND] = 1.0;
          DWI[IND] = 0.0;
          ARG = (float)M_PI/(float)LE1;
          WCOS = (float)cos(ARG);
          WSIN = (float)(D*sin(ARG));

          for (R=1;R<=LE1;R++)
          {
              IND1 = IND+1;
              A = (float)DWR[IND];
              B = (float)DWI[IND];
              DWR[IND1] = A*WCOS - B*WSIN;
              DWI[IND1] = B*WCOS + A*WSIN;
              ++IND;
          }
      }
      /* warmstart: */

      MV2=M/2;
      MM1=M-1;
      J=1;

      for (I=1; I<=MM1; I++)
      {
          if (I >= J)
          goto P1;

          J1 = J-1;
          I1 = I-1;

          VR = (float)RE(X[J1]);
          VI = (float)IM(X[J1]);

          RE(X[J1]) = RE(X[I1]);
          IM(X[J1]) = IM(X[I1]);
```

```
        RE(X'I1') = VR;
        IM(X'I1') = VI;

    P1: K = MV2;
    P2: if (K >= J) goto P3;
        J = J-K;
        K = K/2;
        goto P2;
    P3: J = J+K;
    }
    IND = 0;
    LE = 1;

    for (L=1; L<=P; L++)
    {
         LE1 = LE;
         LE = LE*2;

         for (R=0; R<LE1; R++)
         {
             WCOS = (float)DWR[IND];
             WSIN = (float)DWI[IND];
             IND = IND+1;
             for (IQ=R; IQ<M; IQ+=LE)
             {
                 IP = IQ+LE1;

                 A = (float)RE(X[IP]);
                 B = (float)IM(X[IP]);

                 VR = A*WCOS - B*WSIN;
                 VI = B*WCOS + A*WSIN;

                 RE(X[IP]) = RE(X[IQ]) - VR;
                 IM(X[IP]) = IM(X[IQ]) - VI;

                 RE(X[IQ]) = RE(X[IQ]) + VR;
                 IM(X[IQ]) = IM(X[IQ]) + VI;
             }
         }
    }

    free(DWR);
    free(DWI);
}


/*-------------------------------------------------------------------*/
/*___1-D FFT with respect to a spatial coordinate___________________*/
/*-------------------------------------------------------------------*/
int zffts( int debug, doublecomplex *X, int M )
{
    int P;
    float D;

    D = -1.0;

    rmpo( &M, &P);

    if ( debug )
    {
```

```
            printf("P = %d\n",P);
            printf("FFT ...\n");
        }
        zfftrmc( X, M, P, D); /* fftrm.c */

        return 0;
}

/*=============================================================*/
/*___1-D Inverse FFT with respect to a spatial coordinate_____*/
/*=============================================================*/
int ziffts( int debug, doublecomplex *X, int M )
{
        int i;
        int P;
        float D;

        D = 1.0;

        rmpo( &M, &P);

        if ( debug )
        {
            printf("P = %d\n",P);
            printf("IFFT ...\n");
        }

        zfftrmc( X, M, P, D); /* fftrm.c */

        /*___Multiply with 1/M____*/

        for (i=0; i<M; i++)
        {
            RE(X[i]) /= (doublereal)M;
            IM(X[i]) /= (doublereal)M;
        }

        return 0;
}/*End of fftrm.c*/
```

E.9.4 hilbert.h

```
/*****************************************************************/
/*hilbert.h                                                      */
/*This code contains the necessary functions for extracting      */
/*envelope                                                       */
/*****************************************************************/

#ifndef HILBERT_H_
#define HILBERT_H_

/*This function reads the sampled data recorded in the file*/
int ReadData(void);

/*This function performs the Fourier transform*/
void Fft(void);

/*This function performs the necessary phase shift*/
void PhaseShifting(void);

/*This function performs the inverse Fourier transform*/
void Ifft(void);
```

```
/*Envelope reconstruction is done by this function*/
int EnvelopeReconstruction(void);

/*Hilbert main function*/
void hilbert(char *fnamep);

#endif /* HILBERT_H_ */
```

E.9.5 hilbert.c

```
/ ** * * *
```

```
char *InputFileName ="input.txt";

int SampledPoints=0;
int N;
int row;
const int col=2;

int ReadData(void)
{
     float a,b;
     int i=0;
     FILE *fp1;
     i=0;
     SampledPoints=0; //IA

     if ((fp1 = fopen(InputFileName,"r")) == NULL)
     {
         printf("Cannot open input file.\n");
         return 1;
     }

     while(!feof(fp1))
     {
         fscanf(fp1,"%e,%e\n", &a, &b);

         Gtime[SampledPoints] = a;
         Gvalue[SampledPoints] = b;
         SampledPoints++;
         if (SampledPoints>= MAX_POINT) break;
     }
     fclose(fp1);

     fp1=fopen("inputfile.txt","w");
     if (!fp1)
     {
         fprintf(stdout,"Can't write the sampled data in inputfile.txt. \n");
         return 1;
     }
     for(i=0; i<SampledPoints; i++)
     fprintf(fp1,"%e\n",Gvalue[i]); /*Gtime[i] has been omitted*/
     fclose(fp1);

     if(debug)
     {
         fp1=fopen("inputtime.txt","w");
         if (!fp1)
         {
              fprintf(stdout,"Can't write the sampled data in inputtime.txt. \n");
              return 1;
         }
         for(i=0; i<SampledPoints; i++)
         fprintf(fp1,"%e\n",Gtime[i]); /*Gtime[i] has been omitted*/
         fclose(fp1);
     }

     if(debug)
     {
         if((fp1=fopen("inputfile.bin","wb"))!=NULL)
         {
             fwrite(Gvalue,sizeof(double),SampledPoints,fp1);
             fclose(fp1);
         }
```

```
    }
    if(SampledPoints<N)
    {
        for(i=SampledPoints;i<=N;i++)
        {
        Gvalue[i] = 0;
        }
    }
    return 0;
}/*End Of Function ReadData;*/


void Fft(void)
{
    doublecomplex *Gt_freq;

    FILE *fp1,*fp2,*fp3;
    int k,num1,num2;

    Gt_freq = (doublecomplex *)calloc(sizeof(doublecomplex),row);

    /* FFT Procedure Starts for Sampled Data*/
    for(k=0;k<=N;k++)
    {
        RE(Gt_freq[k])=Gvalue[k];
        IM(Gt_freq[k])=0.0;
    }

    if(debug)
    {
        if((fp3=fopen("f.bin","wb"))!=NULL)
        {
            fwrite(Gvalue,sizeof(double),row,fp3);
            fclose(fp3);
        }
    }

    zffts(fftdebug,Gt_freq,row);/*FFT is done in spatial coordinate*/

    for (k=0;k<=N;k++)
    {
        Gr[k]=RE(Gt_freq[k]);
        Gi[k]=IM(Gt_freq[k]);
    }
    /* FFT Procedure Ends for Sampled Data*/

    /* Writing The Real And Imaginary Part Of Reflected Part for Debuging*/
    /* Writing the real part of sampled data*/

    if(debug)
    {
        if((fp1=fopen("Gr.bin","wb"))!=NULL)
        {
            num1=fwrite(Gr,sizeof(double),row,fp1);
            fclose(fp1);
        }
        else
            fprintf(stdout,"Can't Open Gr.bin");

        // Writing the img part of sampled data
        if((fp2=fopen("Gi.bin","wb"))!=NULL)
        {
```

```
            num2=fwrite(Gi,sizeof(double),row,fp2);
            fclose(fp2);
        }
        else
            fprintf(stdout,"Can't Open Gi.bin");

        fprintf(stdout,"Num of Real Part Data after FFT = %d\n",num1);
        fprintf(stdout,"Num of Img Part Data after FFT = %d\n",num2);
    }

    free(Gt_freq);

}/* End Of The Function Fft */


void PhaseShifting(void)
{
    double *tempr, *tempi;
    int k;
    FILE *fp1;

    tempr = (double *)calloc(sizeof(double),row);
    tempi = (double *)calloc(sizeof(double),row);

    for ( k=0; k<=N; k++ )
    {
        tempr[k]=Gr[k];
        tempi[k]=Gi[k];
    }

    for ( k=0; k<=ceil(N/2); k++ )
    {
        Gr[k] = tempi[k];
        Gi[k] = -tempr[k];
    }

    for ( k=(int)ceil(N/2)+1; k<=N; k++ )
    {
        Gr[k] = -tempi[k];
        Gi[k] = tempr[k];
    }

    if(debug)
    {
        if((fp1=fopen("ffrpt.bin","wb"))!=NULL)
        {
            fwrite(Gr,sizeof(double),row,fp1);
            fclose(fp1);
        }
        if((fp1=fopen("ffipt.bin","wb"))!=NULL)
        {
            fwrite(Gi,sizeof(double),row,fp1);
            fclose(fp1);
        }
    }
    free (tempr);
    free (tempi);
}/*End of PhaseShifting() function*/


void Ifft(void)
{
```

```
    double *Gt_tmp; /* It takes the real part of R_ifft*/
    double *Gt_tmpi;
    FILE *fp1;
    int k,i;

    Gt_tmp = (double *)calloc(sizeof(double),row);
    Gt_tmpi = (double *)calloc(sizeof(double),row);

    for (k=0;k<=N;k++)
    {
        Gt_ifft[k].r=Gr[k];
        Gt_ifft[k].i=Gi[k];
    }

    ziffts(fftdebug,Gt_ifft,row);/*IFFT of the signal in spatial coordinate*/

    // End of IFFT
    for (k=0;k<=N;k++)
    {
        Gt_tmp[k]=Gt_ifft[k].r;
    }

    if(debug)
    {
        fp1=fopen("ifft.txt","w");
        if (!fp1)
            fprintf(stdout,"Can't write in file");
        for(i=0; i<=N; i++)
            fprintf(fp1,"%.4e\n",(Gt_ifft[i].r));
        fclose(fp1);
    }

    if(debug)
    {
        if((fp1=fopen("ifftpt.bin","wb"))!=NULL)
        {
            fwrite(Gt_tmp,sizeof(double),row,fp1);
            fclose(fp1);
        }
        if((fp1=fopen("iffipt.bin","wb"))!=NULL)
        {
            fwrite(Gt_tmpi,sizeof(double),row,fp1);
            fclose(fp1);
        }
    }
    free(Gt_tmp );
    free(Gt_tmpi );
}/* End Of Function Ifft*/


int EnvelopeReconstruction(void)
{
    FILE *fp1;
    int k;

    doublecomplex *G; /*Input signal read from input file in complex form*/
    doublecomplex *Ganalytical;/*Analytical function of our input signal*/

    double *test;
    double *sqrtr;
    double *sqrti;
```

```
    G = (doublecomplex *)calloc(sizeof(doublecomplex),row);
    Ganalytical = (doublecomplex*)calloc(sizeof(doublecomplex),row;

    test = (double *)calloc(sizeof(double),row);
    sqrtr=(double *)calloc(sizeof(double),row);
    sqrti=(double *)calloc(sizeof(double),row);

    for (k=0;k<=N;k++)
    {
        RE(G[k]) = Gvalue[k];
        IM(G[k]) = 0.0;
    }

    for (k=0;k<=N;k++)
    {
        RE(Ganalytical[k])=G[k].r;
        IM(Ganalytical[k])=Gt_ifft[k].r;
    }

    for (k=0;k<=N;k++)
    {

sqrtr[k]=sqrt(Ganalytical[k].r*Ganalytical[k].r+Ganalytical[k].i*Ganalytical[k].i);
    }

    fp1=fopen("output.txt","w");
    if (!fp1)
    {
        fprintf(stdout,"Can't write extracted envelope in output.txt.\n");
        free(G);
        free(Ganalytical);
        free(test);
        free(sqrtr);
        free(sqrti);
        return 1;
    }
    for(k=0; k<SampledPoints; k++)
        fprintf(fp1,"%e,%e\n",Gtime[k],sqrtr[k]);
        fclose(fp1);

    free(G);
    free(Ganalytical);
    free(test);
    free(sqrtr);
    free(sqrti);
    return 0;
}

/*Main Function*/
void hilbert(char *fnamep)
{
    int status=0,i=1;
    char fname[256];
    strcpy(fname, fnamep);
    InputFileName= fname;

    //Reading the sampled data
    do
    {
        N=(int)pow(2,i)-1;
        i++;
```

```
    }while (MAX_POINT > N);

    if (debug)
        printf("N= %d\n",N);

    row=N+1;

    Gvalue = (double *)calloc(sizeof(double),row);
    Gtime = (double *)calloc(sizeof(double),row);
    Gr = (double *)calloc(sizeof(double),row);
    Gi = (double *)calloc(sizeof(double),row);
    Gt_ifft = (doublecomplex *)calloc(sizeof(doublecomplex),row);
    Gc = (double *)calloc(sizeof(double),row);

    status = ReadData();
    if (status== 1) goto MainExit;

    /*Does FFT*/
    Fft();

    /*Appropriate Phase has been Shifted*/
    PhaseShifting();

    /*Does IFFT*/
    Ifft();

    /*Envelope Reconstruction */
    status = EnvelopeReconstruction();
    if (status== 1) goto MainExit;

    MainExit:
    free(Gvalue);
    free(Gtime);
    free(Gr);
    free(Gi);
    free(Gt_ifft);
    free(Gc);

}/*End Of Main*/
```

E.9.6 functs.c

```
/*********************************************************************/
/*functs.c                                                           */
/*This code contains all functions which provide the program*//*functionality.*/
/*Main function of the whole program can be found at the end of    */
/*this file.                                                        */
/*********************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <math.h>
#include "structures.h"
#include "hilbert.h"

#define MAX_SAMPLES 200000
#ifndef M_PI
#define M_PI 3.14159265358979323846264338327
#endif
```

```
// Reads a line from a file (f) and returns two char arrays (s and t)
// It is used to read files written in "comma separation" format.
void read_line (FILE *f, char *s, char *t)
{
     int a=0;
     int i=0;

     a=fgetc(f); /* Takes chars from the file pointed by f */
     while (isspace(a)) /* spaces at the beginning of line are taken out */
     {
         a=fgetc(f);
     }
     while (a!=',' && a!=EOF) /* spaces at the beginning of line are taken out */
     {
         t[i++]=(char)a;
         a=fgetc(f);
     }
     t[i]='\0';
     i=0;
     a=fgetc(f);
     while (a!='\n' && a!=EOF)
     {
         s[i++] =(char)a;
         a=fgetc(f);
     }
     s[i]='\0'; /* We add the line end */
}

// Reads a line from a file (f) and discards it.
void skip_line (FILE *f)
{
     int a=0;

     a=fgetc(f);
     while (a!='\n' && a!=EOF)
     {
         a=fgetc(f);
     }
}

// Creates a ESL node with the given volt and time parameters
ESL *createnode(double voltf, double timef)
{
     ESL *new=NULL;
     new=(ESL *)malloc(sizeof(ESL));
     if (new!=NULL)
     {
         new->volt=voltf;
         new->time=timef;
         new->sig=NULL;
     }
     else
         fprintf(stderr,"Memory Error");
     return new;
}

// Frees the allocated memory for ESL nodes
void freelist(ESL *first) // frees ESL list
{
     ESL *to_free;
     while (first->sig!=NULL)
     {
```

```
            to_free=first;
            first=first->sig;
            free(to_free);
        }
}

// Creates a TIME node with the given volt and time parameters void createtime(TIMES *new,
double tr, double tf, double b, double bVHBR, double trstartind, double trendind, double
tfstartind, double tfendind, double t1, double t1startind, double t1start, double t1endind,
double t2, double t2startind, double t2start, double t3, double t3end, double t3endind,
double t4, double t4endind, double t5, double t5startind, double t6, double t6end, double
t6endind, double a, double tploone)
```

(Измененная редакция. Изм. А4:2012.)

```
{
     new->tf=tf;
     new->tr=tr;
     new->b=b;
     new->bVHBR=bVHBR;
```

(Введено дополнительно. Изм. А4:2012.)

```
     new->trstartind=trstartind;
     new->trendind=trendind;
     new->tfstartind=tfstartind;
     new->tfendind=tfendind;
     new->t1=t1;
     new->t1startind=t1startind;
     new->t1start=t1start;
     new->t1endind=t1endind;
     new->t2=t2;
     new->t2startind=t2startind;
     new->t2start=t2start;
     new->t3=t3;
     new->t3end=t3end;
     new->t3endind=t3endind;
     new->t4=t4;
     new->t4endind=t4endind;
     new->t5=t5;
     new->t5startind=t5startind;
     new->t6=t6;
     new->t6end=t6end;
     new->t6endind=t6endind;
     new->a=a;
     new->tploone=tploone;
}

// Inserts a ESL node (new) in a list pointed by "first"
void insert_node(ESL **first, ESL *new)
{
     ESL *p=NULL;
     ESL *previous=NULL;
     if (new!=NULL)
     {
         p=*first;
         if (p==NULL)
         {
              *first=new;
         }
         else
         {
             while (p!=NULL)
             {
                 previous=p;
```

```
            p=p->sig;
            }
            previous->sig=new;
        }
    }
}

/* Multiplies order polynomials supposing (x^2 + b*x + c) */
/* b and c are complex values stored in a table where even elements are */
/* real and odd elements imaginary */
double *mult_poli (int num_pol, double *b, double *c)
{
    int i=0;
    int y=0;
    double *real;
    double *imag;
    double *vector;
    double *new_real;
    double *new_imag;
    double real_b=0;
    double real_c=0;
    double imag_b=0;
    double imag_c=0;

    real=(double *)calloc(4*num_pol, sizeof(double));
    imag=(double *)calloc(4*num_pol, sizeof(double));
    new_real=(double *)calloc(4*num_pol, sizeof(double));
    new_imag=(double *)calloc(4*num_pol, sizeof(double));
    vector=(double *)calloc(4*num_pol, sizeof(double));

    real[0]=c[0];
    real[1]=b[0];
    real[2]=1;
    imag[0]=c[1];
    imag[1]=b[1];
    imag[2]=0;

    for (i=3; i<(4*num_pol); i++)
    {
        real[i]=0;
        imag[i]=0;
    }

    for (y=1; y<num_pol; y++)
    {
        // Selects values b and c
        real_b=b[2*y];
        real_c=c[2*y];
        imag_b=b[2*y+1];
        imag_c=c[2*y+1];

        for (i=0; i<=(2*num_pol-2); i++)
        {
            // Starts with coeff "c"
            new_real[i]+=real[i]*real_c-imag[i]*imag_c;
            new_imag[i]+=real[i]*imag_c+imag[i]*real_c;
            // Continues with coeff "b"
            new_real[i+1]+=real[i]*real_b-imag[i]*imag_b;
            new_imag[i+1]+=real[i]*imag_b+imag[i]*real_b;
            // Finishes with coeff "1"
            new_real[i+2]+=real[i];
            new_imag[i+2]+=imag[i];
```

```
        }

        // Update Values
        for (i=0;i<(4*num_pol); i++)
        {
             real[i]=new_real[i];
             imag[i]=new_imag[i];
             new_real[i]=0;
             new_imag[i]=0;
        }
    }

    for (y=0; y<(2*num_pol); y++)
    {
        vector[2*y]=imag[y];
        vector[2*y+1]=real[y];
    }

    free (new_imag);
    free (new_real);
    free (imag);
    free (real);

    return (vector);
}

// Part of the calculation of the butterworth coeffs.
double *butter_d_coeffs(double freq1, double freq2)
{
    int butter_order=2;
    int index=0;
    double theta=0;        // M_PI *(freq2-freq1)/2.0
    double cp=0;           // cosine of phi
    double *vec_r=0;       // z^-2 coefficients
    double *vec_t=0;       // z^-1 coefficients
    double *dcoeff=0;      // d coefficients
    double pole_ang=0;     // pole angle
    double divisor=0;

    cp = cos(M_PI*(freq2+freq1)/2.0);
    theta = M_PI*(freq2-freq1)/2.0;

    vec_r=(double *)calloc(2*butter_order,sizeof(double));
    vec_t=(double *)calloc(2*butter_order,sizeof(double));


    for(index=0;index<butter_order;++index)
    {
        pole_ang=M_PI*(double)(2*index+1)/(double)(2*butter_order);
        divisor=sin(2*theta)*sin(pole_ang)+1.0;
        vec_r[2*index]=cos(2*theta)/divisor;
        vec_r[2*index+1]=sin(2*theta)*cos(pole_ang)/divisor;
        vec_t[2*index]=-2.0*cp*(cos(theta)+sin(theta)*sin(pole_ang))/divisor;
        vec_t[2*index+1]=-2.0*cp*sin(theta)*cos(pole_ang)/divisor;
    }

    dcoeff=mult_poli(butter_order,vec_t,vec_r);

    dcoeff[0]=dcoeff[1];
    dcoeff[1]=dcoeff[3];
    dcoeff[2]=dcoeff[5];
    dcoeff[3]=dcoeff[7];
```

```
        acoeff[0]=1;

        for(index=5;index<=2*butter_order;index++)
                acoeff[index]=0;

        free(vec_t);
        free(vec_r);

        return(acoeff);
}

// Calculates the Butterworth filter coefficients
void butterworth_coeffs(double freq1, double freq2, double *dfiltercoeff,
double*cfiltercoeff)
{
        // n filter order
        // freq1, freq2 lower/uppercutoff frequencies

        double sf;          // scaling factor
        double *acoeff;     // a coefficients
        double cotan=0;     // cotangent of theta

        /* calculate the d coefficients */
        acoeff=butter_d_coeffs(freq1,freq2);

        /* d coefficients for 4th order butterworth */
        dfiltercoeff[0]=acoeff[0];    // Always 1
        dfiltercoeff[1]=acoeff[1];
        dfiltercoeff[2]=acoeff[2];
        dfiltercoeff[3]=acoeff[3];
        dfiltercoeff[4]=acoeff[4];

        /* scaling factor for the c filter coefficients (Butterworth 4th order */
        cotan=1.0/tan(M_PI*(freq2-freq1)/2.0);
        sf=(1.0/(((cotan+sqrt(2)/2)*(cotan+sqrt(2)/2))+1/2));

        /* c coefficients for 4th order butterworth*/
        cfiltercoeff[0]=1*sf;
        cfiltercoeff[1]=0*sf;
        cfiltercoeff[2]=-2*sf;
        cfiltercoeff[3]=0*sf;
        cfiltercoeff[4]=1*sf;

        free(acoeff);
}

// Checks if the data input is adequate to our algorithms
int datacheck(int posval, int negval, int samplesp, double tlast,FILE *pointfile)
{
      double diffr=0.0;
      char timestr1[25];
      char timestr2[25];
      char voltstr[25];
      double timestrf1=0;
      double timestrf2=0;
      double cut_sample=0;
      double delta_t=0;
      double val_t=0;
      int loop=0;
      double tinf=0;
      int line=0;
```

```
    // Checks that there are (nearly) much positive as negative values
    if (posval>negval)
            diffr=(posval-negval)/((posval+negval)/2);
    else
            diffr=(negval-posval)/((posval+negval)/2);

    if (diffr>0.8)
            fprintf(stdout,"Data Corrupted: Too little negative (or positive)
     values\n");

    // L=n*p with P=2*pi and n=1,2,3... - Cuts data
    rewind (pointfile);
    read_line(pointfile,voltstr,timestr1);
    read_line(pointfile,voltstr,timestr1);
    read_line(pointfile,voltstr,timestr1);      // Skips csv header if present or not

    read_line(pointfile,voltstr,timestr1);
    while (voltstr[0]!='\0')
    {
        read_line(pointfile,voltstr,timestr2);
        loop++;
    }
    loop=loop+3;

    rewind(pointfile);
    for (lind=0; lind<loop; lind++)
    {
        read_line(pointfile,voltstr,timestr2);
    }
    timestrf1=atof(timestr1);                   // t(4)
    timestrf2=atof(timestr2);                   // t(end)
    cut_sample=((1/13.56e6)/((timestrf2-timestrf1)/(loop-1)));
    linf=samplesp;

    while (linf>0)
    {
        linf=linf-cut_sample;
    }
    linf+=cut_sample;
    samplesp=samplesp-linf-3;

    // At least 7 points per sample
    delta_t=tlast-timestrf1;
    val_t=delta_t/samplesp;
    if (val_t>(1/13.56e6)/7)
        fprintf(stdout, "More samples points needed - Nyquist\n");

    return (samplesp);
}

// Finds the most frequent value(s) of the given signal, Hmax (Types A/B) and Hmin(Type
B).
void Hmaxfinder(double *env, double *Hmax, double *Hmin, int numsamples)
{
    int hist[2001]={0}; // IA Changed memory access violation. Increased +1
    int hi_low_i=0;
    double tophist=0;
    double bothist=100;
    double diffhist=0;
    double value=0;
    int histind=0;
```

```
    int max_i=0;
    int min_i=0;
    double max=0;
    double min=0;

    // Finds higher and lower values of samples
    for (hi_low_i=0; hi_low_i<MAX_SAMPLES; hi_low_i++)
    {
        if (env[hi_low_i]!=0)
        {
            if (env[hi_low_i]<bothist)
                bothist=env[hi_low_i];
            if (env[hi_low_i]>tophist)
                tophist=env[hi_low_i];      // Finds limits for the histogram
        }
    }
    diffhist=tophist-bothist;

    for (hi_low_i=0; hi_low_i<numsamples; hi_low_i++)
    {
        if (env[hi_low_i]!=0)
        {
            value=env[hi_low_i];
            histind=(int)(2000*((value-bothist)/diffhist)); // Performs a lineal
     quantization
            hist[histind]++;
        }
    }

    for (hi_low_i=0; hi_low_i<1000; hi_low_i++)
    {
        if (hist[hi_low_i]>min)     // Searches most frequent value in the lower half of
    the form
        {
            min=hist[hi_low_i];
            min_i=hi_low_i;
        }
        *Hmin=(bothist+(diffhist/2000)*(min_i));
    }

    for (hi_low_i=1001; hi_low_i<2000; hi_low_i++)
    {
        if (hist[hi_low_i]>max)    // Search most frequent value in the upper half of the
    form
        {
            max=hist[hi_low_i];
            max_i=hi_low_i;
        }
        *Hmax=(bothist+(diffhist/2000)*(max_i));
    }
}

// Finds the value of m_min for bit rates of fc/8, fc/4 and fc/2

void Mminfinder(double *env, double Hmax, double Hmin, double *HmaxVHBR, TIMES *timeres,
int numsamples)
{
      int i=0;
      int j=0;
      double compare_hi=0.0;
      double compare_lo=0.0;
      double compare=0.0;
```

```
        double difference=0.0;
        int going_up=0;
        double ampl=0.0;
        double ampl_max=0.0;     // represents the amplitude (Hmax-b), and indirectly "m".
        double m_deviation=0.0; // countermeasure 1: m_min < 0.2*m is not considered
        double Hmax_cm=0.0;      // countermeasure 2: m_min does not start or end on
borders
        double b_cm=0.0;
        double mmin=0.0;
        double mmin_cum=0.0;

        // Skip all zeros
        while (env[j]==0)
           j++;

        // where do we start?
        difference=env[j]-env[j+1];
        if (difference<0)
        {
              going_up=1; // going up
              compare_lo=env[j];
        }
        else if (difference>0)
        {
              going_up=0; // going down
              compare_hi=env[j];
        }
        compare=env[j];

        ampl_max=(Hmax-Hmin);
        m_deviation=ampl_max*0.2;
        Hmax_cm=Hmax*0.95;
        b_cm=Hmin*1.05;
        timeres->bVHBR=0;

        for (i=j; i<=numsamples-j; i++)
        {
              if (going_up==0) // GOING DOWN
              {
                    if (compare>=env[i])
                    {
                          compare=env[i];
                    }
                    else if (compare<env[i])
                    {
                          compare=env[i];
                          compare_lo=env[i];
                          going_up=1; // change direction
                          ampl=(compare_hi-compare_lo);
                          mmin=(ampl/(compare_hi+compare_lo))*100;
                          if (ampl>m_deviation && ampl<ampl_max && (compare_hi<Hmax_cm ||
compare_lo>b_cm))     //Countermeasures
                          {
                                *HmaxVHBR=compare_hi;
                                timeres->bVHBR=compare_lo;
                                ampl_max=ampl;
                          }
                    }
              }
              if (going_up==1) // GOING UP
              {
                    if (compare<=env[i])
```

```
                    {
                        compare=env[i];
                    }
                    else if (compare>env[i])
                    {
                        compare=env[i];
                        compare_hi=env[i];
                        going_up=0; // change direction
                        ampl=(compare_hi-compare_lo);
                        mmin=(ampl/(compare_hi+compare_lo))*100;
                        if (ampl>m_deviation && ampl<ampl_max && (compare_hi<Hmax_cm ||
compare_lo>b_cm)) //Countermeasures
                        {
                            *HmaxVHBR=compare_hi;
                            timeres->bVHBR=compare_lo;
                            ampl_max=ampl;
                            mmin_cum=mmin;
                        }
                    }
                }
            }
            if (*HmaxVHBR==0 || timeres->bVHBR==0) // in case Waveform has only two levels
(typical IM7) Hmin=M
            {
                *HmaxVHBR=Hmax;
                timeres->bVHBR=Hmin;
            }
}
```

(Введено дополнительно. Изм. А4:2012.)

```
// Linear convolution (z= x convolve y)
void LinearConvolution(double X [],double Y[], double Z[] , int lenx, int leny)
{
    double *zptr,s,*xp,*yp;
    int lenz;
    int i,n,n_lo,n_hi;

    lenz=lenx+leny-1;
    zptr=Z;

    for (i=0;i<lenz;i++)
    {
        s=0.0;
        n_lo=0>(i-leny+1)?0:i-leny+1;
        n_hi=lenx-1<i?lenx-1:i;
        xp=X+n_lo;
        yp=Y+i-n_lo;

        for (n=n_lo;n<=n_hi;n++)
        {
            s+=*xp * *yp;
            xp++;
            yp--;
        }

        *zptr=s;
        zptr++;
    }
}
```

```
int envfilt(int rate, double *output, double *toutput, int filterlength,double tbin, double
tend, int lengthp, double *envelope)
```

(Измененная редакция. Изм. А4:2012.)

```
{
    int cofp1=0;
    int xx=0;
    double cofp=0.0;
    int lengthp1=0;
    double lengthf=0;
    double cof[2000]={0};
    double points=0.0;
    int points1=0;
    int lengthtotal=0;
    cofp=(73.75e-9)/((tend-tini)/(lengthp));
    cofp1=cofp+0.5;
    lengthf=cofp1*filterlength;
    points=(5*73.75e-9)/((tend-tini)/(lengthp-1));
    points1=(int)points+1;
    lengthp1=lengthp;

    for (xx=0; xx<lengthf; xx++)
        cof[xx]=1/lengthf;

    for (xx=lengthf+1; xx<2000; xx++)
        cof[xx]=0;

    if (rate==106 || rate==212 || rate==424 || rate==848)
    {
        LinearConvolution(cof, output, envelope, lengthf, lengthp);
    }

    else if (rate==1700 || rate==3400 || rate==6800)
    {
         cof[0]=1;
         for (xx=1; xx<2000; xx++)
             cof[xx]=0;
         lengthf=1;
         LinearConvolution(cof, output, envelope, lengthf, lengthp);
    }
```

(Измененная редакция. Изм. А4:2012.)

```
    for (xx=0; xx<(points1); xx++) // "Cuts" envelope
    {
        envelope[xx]=0.0;
        toutput[xx]=0.0;
        envelope[lengthp1-xx]=0.0;
        toutput[lengthp1-xx]=0.0;
    }

    for (xx=lengthp1+1; xx<MAX_SAMPLES; xx++)
    {
        envelope[xx]=0.0;
        toutput[xx]=0.0;
    }

    lengthtotal=lengthp1-2*(points1);
    return (lengthtotal);
}

// Performs the search of a certain level (target) in the envelope, i.e. 5% ,60%, 90% in
Type A, 106 kbit/s
int localizador(double *env, double *toutput, double target, ESL **crosses, int env_
length)
{
```

```
    int flag=0;
    double diff;
    ESL *new;
    double v;
    double t;
    int crosscounter=0;
    int locat_index=0;
    int locat_index_start=0;

    while (env[locat_index]==0.0) // Leaves 0s out
        locat_index++;

    locat_index_start=locat_index;
    if (env[locat_index]-target>0)
    {
         for (locat_index=locat_index_start;
locat_index<env_length+locat_index_start-1;locat_index++)
         {
             diff=env[locat_index]-target;
             if (diff<0 && flag==0 && env[locat_index]!=0.0) // At the beginning or
after an odd occurrence, envelope is over "target" level
             {
                  flag=1; // down!
                  v=target;
                  t=toutput[locat_index-1]+((toutput[locat_index]-toutput[locat_
index-1])/(env[locat_index]-env[locat_index-1])*(target-env[locat_index-1]));
                  new=createnodef(v,t);
                  insert_node(crosses,new);
                  crosscounter++;
             }
             if (diff>0 && flag==1 && env[locat_index]!=0.0) // After first (or even)
occurrence, envelope is under "target" level
             {
                  flag=0; // up!
                  v=target;
                  t=toutput[locat_index-1]+((toutput[locat_index]-toutput[locat_
index-1])/(env[locat_index]-env[locat_index-1])*(target-env[locat_index-1]));
                  new=createnodef(v,t);
                  insert_node(crosses,new);
                  crosscounter++;
             }
                // Returns all occurrences with time and volt level in a list
         }
    }
    else
        fprintf(stdout,"Signal is not ---  ___  --- \n");

    return (crosscounter);            // Also returns how many occurrences appeared
}

// Function that calculates the relevant times
void tfinder(char type, double *env, double *toutput, double tini, double Hmax, double
Hmin, int rate, int env_length, TIMES *timeres)
{
    double *envc=NULL;
    double *envc2=NULL;
    ESL *crosses=NULL;
    ESL *crosses2=NULL;
    ESL *crosses3=NULL;
    ESL *crosses_WORK=NULL;
    double ninety=0.0;
    double five=0.0;
```

```
double sixty=0.0;
double tp90one=0.0;
double tp90two=0.0;
double tp5one=0.0;
double tp5two=0.0;
double tp60two=0.0;
double vp90one=0.0;
double vp90two=0.0;
double vp5one=0.0;
double vp5two=0.0;
double vp60two=0.0;
double tphione=0.0;
double tphitwo=0.0;
double vphione=0.0;
double vphitwo=0.0;
double tpmidone=0.0;
double vpmidone=0.0;
double tploone=0.0;
double tplotwo=0.0;
double vplotwo=0.0;
double t1=0.0;
double t2=0.0;
double t3=0.0;
double t4=0.0;
double t5=0.0;
double t6=0.0;
int flag=0;
int flag2=0;
int flag3=0;
int flag_improv=0;
int x_improv=0;
double minvolt=0.0;
double highrate_low=0.0;
double highrate_mid=0.0;
double highrate_hi=0.0;
double a=0.0;
double t6end=0.0;
double t5startind=0.0;
double t6endind=0.0;
double b=0.0;
double B_low=0.0;
double B_hi=0.0;
double tr=0.0;
double tf=0.0;
double tfstartind=0.0;
double tfendind=0.0;
double trstartind=0.0;
double trendind=0.0;
double t2startind=0.0;
double t2start=0.0;
double t1startind=0.0;
double t1start=0.0;
double t3end=0.0;
double t4endind=0.0;
double t3endind=0.0;
double t1endind=0.0;
double oscmin=0.0;
double osctmin=0.0;
double oscmax=0.0;
double osctmax=0.0;
ESL *crossescopy=NULL;
double tim3=0.0;
```

```
int index_A=0;
int index_A2=0;
int index_chain=0;
int i=0;
double *toutput2=NULL;
int counter=0;
int rev_counter=0;
int VHBR_step=0;
double VHBR_tr=0.0;
double VHBR_tf=0.0;
double tr_accum=0.0;
double tf_accum=0.0;
int tr_counter=0;
int tf_counter=0;
double t_one_sample=0.0;
double tlo=0.0;
double vlo=0.0;
double thi=0.0;
double vhi=0.0;

toutput2=toutput;
```

(Введено дополнительно. Изм. А4:2012.)

```
b=Hmin;
envc=env;
envc2=envc;

switch (type)
{
    case 'A':
    {
        switch (rate)
        {
         case 106:
         {
             ninety=Hmax*0.9;
             five=Hmax*0.05;
             sixty=Hmax*0.6;
             flag2=localizador(envc,toutput,five,&crosses2,env_length);     // Finds
5% of Hmax
             if (flag2==2)      // if there are two occurrences, there s no
problem...
             {
                 tp5one=crosses2->time;     // Temporary values are stored for
future use
                 vp5one=crosses2->volt;
                 tp5two=crosses2->sig->time;
                 vp5two=crosses2->sig->volt;
                 freelist(crosses2);
             }
             else if (flag2==0)             // ...if there is no occurrence...
                 fprintf(stdout,"5 percent of Hmax not reached - maybe wrong type
or bitrate? \n");

             else if (flag2>2)              // ...if there are more than two
occurrences...
             {
                                            // ...it must be checked that "peaks"
comply the ISO restrictions

                 while (toutput[index_A]<crosses2->sig->time)
                     index_A++;
```

```
                    oscmin=envc2[index_A];
                    for (index_chain=0; index_chain<flag2-2; index_chain++)
                    {
                        crosses2=crosses2->sig;
                    }

                    while (toutput[index_A]<=crosses2->time)
                    {
                        if (envc2[index_A]>oscmax)
                        {
                            oscmax=envc2[index_A];
                            osctmax=toutput[index_A];
                        }
                        index_A++;
                    }

                    while (envc2[index_A2]==0)
                    {
                        index_A2++;
                    }

                    while (envc2[index_A2]>oscmax)
                    {
                        index_A2++;
                    }

                    oscmin=envc2[index_A2];
                    osctmin=toutput[index_A2];

                    if (osctmax-osctmin>5e-7)
                        fprintf(stdout,"Monotony not fulfilled \n");

                    tp5one=crosses2->time;                    // Temporary values are
stored for future use
                    vp5one=crosses2->volt;
                    tp5two=crosses2->sig->time;
                    vp5two=crosses2->sig->volt;
                    freelist(crosses2);
                }

                flag=local_zador(envc,toutput,n_nety,&crosses,env_length); //Finds 90%
of Hmax
                if (flag>=2)
                {
                    crosses_WORK=crosses;                     // Copy of crosses to work
with
                    while (x_improv<flag)
                    {
                        if (crosses_WORK->time<tp5one)
                        {
                            tp90one=crosses_WORK->time; // Temporary values are
stored for future use
                            vp90one=crosses_WORK->volt;
                        }

                        if (crosses_WORK->time>tp5two && flag_improv==0)
                        {
                            tp90two=crosses_WORK->time; // Temporary values are
stored for future use
                            vp90two=crosses_WORK->volt;
                            flag_improv=1;
                        }
```

```
                        crosses_WORK=crosses_WORK->sig;
                        x_improv++;
                    }
                    freelist(crosses);
                }

                else            // ...otherwise...
                {
                    fprintf(stdout,"90 %% of Hmax not found - Noise Too High \n");
                }


                flag3=localizador(envc,toutput,s_xty,&crosses3,env_length); // Finds
90% of Hmax
                if (flag3==2) // If there are two occurrences, there's no problem...
                {
                    tp60two=crosses3->sig->time;    // Temporary values are stored
for future use
                    vp60two=crosses3->sig->volt;
                    freelist(crosses3);
                }

                t1=tp5two-tp90one;    // Definitive values are calculated and stored
for display
                t2=tp5two-tp5one;
                t3=tp90two-tp5two;
                t4=tp60two-tp5two;

                t1start=tp90one;    // Other important values for the coming
functions
                t2start=tp5one;
                t3end=tp90two;
                t1startind=vp90one;
                t1endind=vp5two;
                t2startind=vp5one;
                t3endind=vp90two;
                t4endind=vp60two;

createtime(timeres,0,0,0,0,0,0,0,0,t1,t1startind,t1start,t1endind,t2,t2startind,t2s
tart,t3,t3end,t3endind,t4,t4endind,0,0,0,0,0,0,0);
```

(Измененная редакция. Изм. A4:2012.)

```
            }
            break;

            case 212:
            case 424:
            case 848:
            {
                ninety=Hmax*0.9;
                while (env[index_A]==0.0)        // Finds first value different of 0.0
                    index_A++;

                minvolt=env[index_A];
                while (env[index_A]!=0.0)        // All values are considered
                {
                    if (env[index_A]<minvolt)
                    {
                        minvolt=env[index_A];    // Finds minimal voltage
                    }
                    index_A++;
                }

                highrate_low=minvolt+0.1*(Hmax-minvolt);        // Calculates target
```

```
                flag=localizador(envc,toutput,highrate_low,&crosses,env_length);
//Finds target
                if (flag==2)      // ...if there are two occurrences, there's no
problem...
                {
                    tploone=crosses->time;    // Temporary values are stored for
future use
                    tplotwo=crosses->sig->time;
                    vplotwo=crosses->sig->volt;
                }

                else if (flag>2) // if there are more than two occurrences...
                {                // ...it must be checked that "peaks" comply the ISO
restrictions
                    while (toutput[index_A2]<crosses->time || toutput[index_A2]==0)
                        index_A2++;

                    oscmin=envc2[index_A2];
                    while (envc2[index_A2]<=oscmin)
                    {
                        oscmin=envc2[index_A2];
                        index_A2++;
                    }
                    osctmin=toutput[index_A2];

                    crossescopy=crosses;
                    for (i=1; i<(flag-1); i++)
                        crossescopy=crossescopy->sig;

                    tim3=crossescopy->time;
                    while (toutput[index_A2]<tim3)
                        index_A2++;

                    oscmax=envc2[index_A2];
                    while (toutput[index_A2]<tim3)
                    {
                        if (oscmax<envc2[index_A2])
                        {
                            oscmax=envc2[index_A2];
                            osctmax=toutput[index_A2];
                        }
                        index_A2++;
                    }
                    if (oscmax-oscmin>(0.09*(Zmax-oscmin)))
                        fprintf(stdout,"Monotony not fulfilled \n");

                    for (i=1; i<(flag-1); i++)
                        crosses=crosses->sig;

                    tploone=crosses->time; // Temporary values are stored for future use
                    tplotwo=crosses->sig->time;
                    vplotwo=crosses->sig->volt;
                }
                freelist(crosses);

                highrate_hi=ninety+0.1*minvolt;    // Calculates target
                flag=localizador(envc,toutput,highrate_hi,&crosses2,env_length);
//Finds target
                if (flag>=2)
                {
                     crosses_WORK=crosses2;
                     while (x_improv<flag)
                     {
                         if (crosses_WORK->time<tploone)
```

```
                        {
                            tphione=crosses_WORK->time; // Temporary values are
stored for future use
                            vphione=crosses_WORK->volt;
                        }

                        if (crosses_WORK->time>tplotwo && flag_improv==0)
                        {
                            tphitwo=crosses_WORK->time; // Temporary values are
stored for future use
                            vphitwo=crosses_WORK->volt;
                            flag_improv=1;
                        }
                            crosses_WORK=crosses_WORK->sig;
                            x_improv++;
                    }
                    freelist(crosses2);
                }
                else                // ...otherwise...
                {
                    fprintf(stdout,"90 % of Hmax not reached! - Noise Too High?\n");
                }

                highrate_mid=(Hmax+minvolt)/2;      // Calculates target
                flag=localizador(envc,toutput,highrate_mid,&crosses3,env_length);
//Finds target
                if (flag==2)                        // ...if there are two occurrences,
there's no problem...
                {
                    tpmidone=crosses3->time;      // Temporary values are stored for
future use
                    vpmidone=crosses3->volt;
                    freelist(crosses3);
                }
                else                // ...otherwise...
                    fprintf(stdout,"Noise Too High \n");
                t1=tplotwo-tphione;// Definitive values are calculated and stored for
display
                t5=tplotwo-tpmidone;
                t6=tphitwo-tplotwo;
                a=minvolt;

                t6end=tphitwo;
                                    // Other important values for the coming functions
                t1start=tphione;
                t1startind=vphione;
                t5startind=vpmidone;
                t1endind=vplotwo;
                t6endind=vphitwo;


createtime(timeres,0,0,0,0,0,0,0,0,t1,t1startind,t1start,t1endind,0,0,0,0,0,0,0,0,t
5,t5startind,t6,t6end,t6endind,a,tploone);
```

(Измененная редакция. Изм. А4:2012.)

```
            }
            break;
        }
    }
    break;

    case 'B':
    {
        switch (rate)
        {
```

```
                    case 106:
                    case 212:
                    case 424:
                    case 848:
                    {
                         B_low=b+0.1*(Hmax-b);       // Calculates target
                         flag=localizador(envc,toutput,B_low,&crosses,env_length); //
                    Finds target
                         if (flag>=2)
                         {
                              crosses_WORK=crosses;
                              tploone=crosses_WORK->time;     // Temporary values are
stored for future use
                              while (x_improv<flag)
                              {
                                   tplotwo=crosses_WORK->time; // Temporary values are
stored for future use
                                   vplotwo=crosses_WORK->volt;
                                   crosses_WORK=crosses_WORK->sig;
                                   x_improv++;
                              }
                              freelist(crosses);
                         }
                         else
                         {
                              fprintf(stdout,"Monotony not fulfilled\n");
                         }

                         B_hi=Hmax-0.1*(Hmax-b);     // Calculates target
                         flag=localizador(envc,toutput,B_hi,&crosses2,env_length); //
Finds target
                         if (flag>=2)
                         {
                              x_improv=0;
                              flag_improv=0;
                              crosses_WORK=crosses2;
                              while (x_improv<flag)
                              {
                                   if (crosses_WORK->time<tploone)
                                   {
                                        tphione=crosses_WORK->time; // Temporary values
are stored for future use
                                        vphione=crosses_WORK->volt;
                                   }

                                   if (crosses_WORK->time>tplotwo && flag_improv==0)
                                   {
                                        tphitwo=crosses_WORK->time; // Temporary values
are stored for future use
                                        vphitwo=crosses_WORK->volt;
                                        flag_improv=1;
                                   }
                                        crosses_WORK=crosses_WORK->sig;
                                        x_improv++;
                              }
                              freelist(crosses2);
                         }

                         else
                         {
                              fprintf(stdout,"Monotony not fulfilled\n");
                         }

                         tf=tploone-tphione; // Definitive values are calculated and
```

```
stored for display
                    tr=tphitwo-tplotwo;
                    tfstarting=tphione; // Other important values for the coming
functions
                    tfending=tploone;
                    trstarting=tplotwo;
                    trendind=tphitwo;
createtime(timeres,tr,tf,h,0,trstartind,trendind,tfstartind,tfendind,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0);
                }
                break;

                case 1700:
                case 3400:
                case 6800:
                {
                    B_low=b+0.1*(Hmax-b);       // Calculates target
                    B_hi=Hmax-0.1*(Hmax-L);     // Calculates target

                    flag=localizador(envc,toutput,B_low,&crosses,env_length);
                    flag2=localizador(envc,toutput,B_hi,&crosses2,env_length);

                    tfstartind=crosses2->time; // Reused as start point for
overshoot
                    tfendind=crosses->time;    // and undershoot

                    // LOCATE ADJACENT POINTS
                    while (crosses->sig!=NULL && crosses2->sig!=NULL)
                    {
                        tlo=crosses->time;
                        thi=crosses2->time;
                        if (thi<tlo)          // FALLING EDGE
                        {
                            if (crosses2->sig->time < tlo)     // Discard Point
                                crosses2=crosses2->sig;
                            else if (crosses2->sig->time > tlo) // Analysis if
                            {
                                vlo=crosses->volt;
                                vhi=crosses2->volt;
                                while (toutput2[counter]==0) // set counters
                                {
                                    counter++;
                                    rev_counter++;
                                }
                                t_one_sample=toutput2[counter+2]-toutput2[counter+1];
                                while (toutput2[counter]<=thi) // set counters
                                {
                                    counter++;
                                    rev_counter++;
                                }
                                while (toutput2[rev_counter]<=tlo) // set
                            counters
                                    rev_counter++;

                                while (vlo<vhi)
                                {
                                    vlo=envc2[rev_counter-VHBR_step];
                                    vhi=envc2[counter+VHBR_step];
                                    VHBR_step++;
                                }
                                if (vlo==vhi)
                                    VHBR_step=VHBR_step*2;
                                else if (vlo>vhi)
                                    VHBR_step=VHBR_step*2-1;
```

```
                    VHBR_tf=VHBR_step*t_one_sample;
                    tf_counter++;
                    tf_accum=tf_accum+VHBR_tf;

                    VHBR_step=0.0; // Reset Counters
                    VHBR_tf=0.0;
                    counter=0;
                    rev_counter=0;
                    crosses2=crosses2->sig;
               }
          }

          else if (tlo<thi)         // RISING EDGE
          {
               if (crosses->sig->time < thi) // Discard Point
                    crosses=crosses->sig;
               else if (crosses->sig->time > thi) // Analysis tr
               {
                    vlo=crosses->volt;
                    vhi=crosses1->volt;
                    while (toutput2[counter]==0) // set counters
                    {
                         counter++;
                         rev_counter++;
                    }
                    t_one_sample=toutput2[counter+2]-
                    toutput2[counter+1];
                    while (toutput2[counter]<=tlo) // set counters
                    {
                         counter++;
                         rev_counter++;
                    }
                    while (toutput2[rev_counter]<=thi) // set
               counters
                         rev_counter++;

                    while (vlo<vhi)
                    {
                         vhi=envc2[rev_counter-VHBR_step];
                         vlo=envc2[counter+VHBR_step];
                         VHBR_step++;
                    }
                    if (vlo==vhi)
                         VHBR_step=VHBR_step*2;
                    else if (vlo>vhi)
                         VHBR_step=VHBR_step*2-1;

                    VHBR_tr=VHBR_step*t_one_sample;
                    tr_counter++;
                    tr_accum=tr_accum+VHBR_tr;

                    VHBR_step=0.0;      // Reset Counters
                    VHBR_tr=0.0;
                    counter=0;
                    rev_counter=0;
                    crosses=crosses->sig;
               }
          }
     }
     // Calculate and Save Parameters
     tf=tf_accum/tf_counter;     // Definitive values are calculated
and stored for display
     tr=tr_accum/tr_counter;     // Reused as end point for overshoot
     trending=crosses2->time;
```

```
                        trstarting=crosses->time;
createtime(timeres,tr,tf,b,0,trstartind,trendind,tfstartind,tfendind,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0);
                    }
                    break;
                }
            }
            break;
        }
}
```

(Измененная редакция. Изм. А4:2012.)

```
// Checks monotony on the falling edge
void monocheck(double *env, double *toutput, double Hmax, TIMES *timesp, int rate,char
type)
{
    double tinit=0.0;
    double tend=0.0;
    double compare=0.0;
    double timer0=0.0;
    double timer1=0.0;
    double volt0=0.0;
    double volt1=0.0;
    int counter=0;
    int flag_mono=0;

    switch (type)
    {
        case 'A':
        {
            switch (rate)
            {
            case 106:
                {
                    while (env[counter]==0)
                        counter++;

                    tinit=timesp->t1start;
                    tend=timesp->t2start;

                    while (toutput[counter]<tinit)
                        counter++;   //find first value

                    while (toutput[counter]<tend)
                    {
                        compare=env[counter];
                        if (compare<env[counter+1])
                        {
                            timer0=toutput[counter];
                            volt0=env[counter];
                            while (volt0<env[counter+1]) // growing values...
                            {
                                counter++;
                                volt0=env[counter];
                            }
                            timer1=toutput[counter]; // ...max.value -> time
                            if (timer1-timer0>5e-6)
                                fprintf(stdout,"Monotony not fulfilled \n");
                        }
                        else
                        counter++;
                    }
                }
```

```
            break;

            case 212:
            case 424:
            case 848:
            {
                while (env[counter]==0)
                    counter++;

                tinit=timesp->tlstart;
                tend=timesp->tlendind;
                while (toutput[counter]<tinit)
                    counter++; //find first value

                while (env[counter]>tend)
                {
                    compare=env[counter];
                    if (compare<env[counter+1])
                    {
                        volt0=env[counter];
                        volt1=volt0;
                        while (volt0<env[counter+1]) // growing values...
                        {
                            counter++;
                            volt0=env[counter];
                        }
                        if (volt1-volt0>0.09*(Hmax-volt0))
                            fprintf(stdout,"Monotony not fulfilled \n");
                    }
                    else
                    counter++;
                }
            }
            break;
        }
    }
    break;

    case '3':
    {
        while (env counter'==0)
            counter++;

        tinit=timesp->tlstartind;
        tend=timesp->tfendind;
        while (toutput'counter'<tinit)
            counter++; //find first value

        while (toutput counter'<tend)
        {
            compare=env[counter];
            if (compare<env[counter++])
                flag_mono=1;
        }
        if (flag_mono==1)
            fprintf(stdout, "Monotony not fulfilled \n");
    }
    break;
  }
}

// Function that calculates the overshoot times
```

```
void overshoot(TIMES *timesp, double Hmax, double *env2, double *toutput, int rate,char
type, int samples, SHOOTREADER *shootreader)
{
      double shootind=0.0;
      double shooting_b=0.0;
      double hr_reader=0.0;
      double hf_reader=0.0;
      double above=0.0;
      double above_b=10.0;
      double start=0.0;
      int index_samples=0;

      switch (type)
      {
          case 'A':
          {
              switch (rate)
              {
                  case (106):
                  {
                      start=timesp->t3end;
                      while (toutput[index_samples]<=start)
                          index_samples++;

                      while (env2[index_samples]!=0)
                      {
                           if (env2[index_samples]>above)
```

(Измененная редакция. Изм. А4:2012.)

```
                           {
                              above=env2[index_samples];
                              shooting=toutput[index_samples];
                           }
                          index_samples++;
                      }
                  }
                  break;
                  case (212):
                  case (424):
                  case (848):
                  {
                      start=timesp->t6end;
                      while (toutput[index_samples]<=start)
                          index_samples++;

                      while (index_samples<=samples)
                      {
                          if (env2[index_samples]>above)
                          {
                              above=env2[index_samples];
                              shootind=toutput[index_samples];
                          }
                          index_samples++;
                      }

                      if (above<Hmax)      // In very strange cases if there's no over-
shoot, the highest point
                          above=Hmax;      // in the curve can be cutted off by envfilt,
producing a negative hr

                  }
                  break;
              }
```

```
        ;
        break;
        case 'B':
        {
            start=timesp->trendind;
            while (toutput[index_samples]<start)       // Starts at the rising edge
                index_samples++;

            while (index_samples<=samples)
            {
                if (env2[index_samples]>above)
                {
                    above=env2[index_samples];
                    shootind=toutput[index_samples];
                    hr_reader=(above-Hmax)/(Hmax-timesp->h);
                    if (hr_reader<0)    // In very strange cases if there's no overshoot,
the highest point
                        hr_reader=0;    // in the curve can be cutted off by envfilt,
producing a negative hr
                }
                index_samples++;
            }

            index_samples=0;
            start=timesp->tfendind;
            while (toutput[index_samples]==0)
                index_samples++;
            while (toutput[index_samples]<start)
                index_samples++;

            while (env2[index_samples]!=0)
            {
            if (env2[index_samples]<above_b)
```

(Измененная редакция. Изм. А4:2012.)

```
                {
                    above_b=env2[index_samples];
                    shootind_b=toutput[index_samples];
                    hf_reader=(timesp->b-above_b)/(Hmax-timesp->b);
                }
                index_samples++;
            }
        }
        break;
    }
    shootreader->shootind=shootind;
    shootreader->shootind_b=shootind_b;
    shootreader->hr_reader=hr_reader;
    shootreader->hf_reader=hf_reader;
    shootreader->above=above;
    shootreader->above_b=above_b;
}

// Calculates the modulation index "m"
double modulation(char type, double Hmax, double b)
{
    double m=0;
    switch (type)
    {
        case 'A':
        {
            // m is not defined for Type A
        }
```

```
            break;
            case 'B':
            {
                m=100*(Hmax-n)/(Hmax+b); // In %
            }
            break;
        }
        return (m);
}

// Displays on screen the results of the calculations
void display(char type, int rate, SHOOTREADER *shootreader2, TIMES *timesp, double Hmax,
double m, double mmin)
```

(Измененная редакция. Изм. А4:2012.)

```
{
        double ovs=0;
        double ovsb1=0;
        double ovsb2=0;
        fprintf (stdout,"\n");    // 2nd set of functions, on debug purposes
        switch (type)
        {
            case 'A':
            {
                fprintf(stdout,"---RESULTS-----------\n");
                fprintf(stdout,"Type A - Bitrate %d\n", rate);
                fprintf(stdout,"---Overshoot---------\n");
               ovs=(((shootreader2->above)-Hmax)/(Hmax-timesp->a))*100;
               if (ovs>0)
                  fprintf(stdout,"Overshoot = %f %% \n", ovs);
               else
                  fprintf(stdout,"Overshoot = 0 %% \n");
                   switch (rate)
                   {
                       case (106):
                       {
                           fprintf(stdout,"---timings-----------\n");
                           fprintf(stdout,"t1 = %f microsec. \n", (timesp->t1)*1e6);
                           fprintf(stdout,"t1 = %f/fc \n", (timesp->t1)*13.56e6);
                           fprintf(stdout,"t2 = %f microsec. \n", (timesp->t2)*1e6);
                           fprintf(stdout,"t2 = %f/fc \n", (timesp->t2)*13.56e6);
                           fprintf(stdout,"t3 = %f microsec. \n", (timesp->t3)*1e6);
                           fprintf(stdout,"t3 = %f/fc \n", (timesp->t3)*13.56e6);
                           fprintf(stdout,"t4 = %f microsec. \n", (timesp->t4)*1e6);
                           fprintf(stdout,"t4 = %f/fc \n", (timesp->t4)*13.56e6);
                           fprintf(stdout,"---amplitudes---------\n");
                           fprintf(stdout,"Hmax = %f volts \n", Hmax);
                           fprintf(stdout,"Max. Amplitude = %f volts \n", (shootreader2->above));
                       }
                       break;
                       case (212):
                       case (424):
                       case (848):
                       {
                           fprintf(stdout,"bovs = %f \n", (((shootreader2->above-Hmax)/Hmax)));
                           fprintf(stdout,"---timings-----------\n");
                           fprintf(stdout,"t1 = %f microsec. \n", (timesp->t1)*1e6);
                           fprintf(stdout,"t1 = %f/fc \n", (timesp->t1)*13.56e6);
                           fprintf(stdout,"t5 = %f microsec. \n", (timesp->t5)*1e6);
                           fprintf(stdout,"t5 = %f/fc \n", (timesp->t5)*13.56e6);
                           fprintf(stdout,"t6 = %f microsec. \n", (timesp->t6)*1e6);
                           fprintf(stdout,"t6 = %f/fc \n", (timesp->t6)*13.56e6);
                           fprintf(stdout,"---amplitudes---------\n");
```

```
                    fprintf(stdout,"Hmax = %f volts \n", Hmax);
                    fprintf(stdout,"a = %f %% of Hinitial \n", ((timesp->a)/Hmax));
                }
                break;
            }
        }
        break;
        case 'B':
        {
            fprintf(stdout,"---RESULTS-----------\n");
         if (rate<=848)
            fprintf(stdout,"Type B - bit rate %d\n", rate);
         else
            fprintf(stdout,"Type VHBR - bit rate %d\n", rate);
```

(Измененная редакция. Изм. А4:2012.)

```
            fprintf(stdout,"---timings-----------\n");
            fprintf(stdout,"tf = %f microsec. \n", (timesp->tf)*1e6);
            fprintf(stdout,"tf = %f/fc \n", (timesp->tf)*13.56e6);
            fprintf(stdout,"tr = %f microsec. \n", (timesp->tr)*1e6);
            fprintf(stdout,"tr = %f/fc \n", (timesp->tr)*13.56e6);
            fprintf(stdout,"---modulation---------\n");
            fprintf(stdout,"m = %f %% \n", m);
         if((rate==1700    rate==3400    rate==6800))
            fprintf(stdout,"m_min = %f %% \n", mmin);
```

(Введено дополнительно. Изм. А4:2012.)

```
            fprintf(stdout,"---amplitudes---------\n");
            fprintf(stdout,"a = %f volts \n", Hmax);
            fprintf(stdout,"b = %f volts \n", timesp->b);
            fprintf(stdout,"---Overshoots---------\n");
            fprintf(stdout,"hf = %f %% of Hinitial-b\n", (shootreader2->hf_reader)*100);
            fprintf(stdout,"hr = %f %% of Hinitial-b\n", (shootreader2->hr_reader)*100);
         ovsb1=(timesp->b-(shootreader2->above_b))*1000;
         ovsb2=((shootreader2->above)-Hmax)*1000;
         if (ovsb1>0)
            fprintf(stdout,"hf = %f milivolts \n", ovsb1);
         else
            fprintf(stdout,"hf = 0 milivolts \n");
         if (ovsb2>0)
            fprintf(stdout,"hr = %f milivolts \n", ovsb2);
         else
            fprintf(stdout,"hr = 0 milivolts \n");
        }
        break;
    }
}

// Main Function
int main (int argc, char *argv[])
{
    char type;
    int rate;
    char voltstr[25];      // intermediate char array to modify the voltage values
    char timestr[25];      // intermediate char array to modify the time values
    double snum=0;
    double tnum=0;
    double t=0;
    int filterlength=0;
    double Hmax=0;
    double HmaxVHBR=0;
    double Hmin=0;
    double Hmax2=0;
    double Hmin2=0;
```

```
      FILE *pointfile=NULL;
      FILE *input_u2=NULL;
      FILE *poutput=NULL;
      double m=0.0;
      double mmin=0.0;
      int length=0;
      double val=0;
      int posval=0;
      int negval=0;
      double tini=0;
      double tfin=0;
      int samples=0;
      int out_i=0;
      int length_total=0;
      int sample_ini=0;
      int sample_end=0;
      int flag_cut=0;
      int samplesp=0;
      int fi=0;             // Filter generic index
      double b1=0;          // Filter parameters
      double b2=0;
      double b3=0;
      double b4=0;
      double b5=0;
      double a1=0;
      double a2=0;
      double a3=0;
      double a4=0;
      double a5=0;
      double freq1=0;
      double freq2=0;
      double as[5]={0};
      double bs[5]={0};
      double t0=0;
      double tlast=0;
      int lineskip=0;
      double *voutput=malloc (sizeof(double)*MAX_SAMPLES);
      double *toutput=malloc (sizeof(double)*MAX_SAMPLES);
      double *envelope=malloc (sizeof(double)*MAX_SAMPLES);
      double *vfilter=malloc (sizeof(double)*MAX_SAMPLES);
      double *tfilter=malloc (sizeof(double)*MAX_SAMPLES);
      TIMES *timesp=(TIMES *)malloc(sizeof(TIMES));
      TIMES *timesp2=(TIMES *)malloc(sizeof(TIMES));
      SHOOTREADER *shootreader2=(SHOOTREADER *)malloc(sizeof(SHOOTREADER));

      if (voutput!=NULL && toutput!=NULL && envelope!=NULL && vfilter!=NULL && tfilter!=NULL
&& timesp!=NULL && timesp2!=NULL && shootreader2!=NULL)
      {
            memset(voutput, 0, MAX_SAMPLES);
            memset(toutput, 0, MAX_SAMPLES);
            memset(envelope, 0, MAX_SAMPLES);
            memset(vfilter, 0, MAX_SAMPLES);
            memset(tfilter, 0, MAX_SAMPLES);

            type=*argv[1];
            rate=atoi(argv[2]);
            if (type!='A' && type!='B' && type!='V')
                  fprintf(stdout, "Wrong Type (A, B or VHBR)");
            else if ((type=='A' || type=='B') && (rate!=106 && rate!=212 && rate!=424 &&
rate!=848))
                  fprintf(stdout, "Wrong bit rate (106, 212, 424, 848)");
```

```
        else if ((type=='V') && (rate!=1700 && rate!=3400 && rate!=6800))
            fprintf(stdout, "Wrong bit rate (1700, 3400, 6800)");
        else
        {
            if (type=='V')
                type='B';
                    pointfile=fopen(argv[3],"r");
                input_u2=fopen("pre_Hilbert.txt","w");      // modified-intermediate
amplitude vector

                if(pointfile!=NULL && input_u2!=NULL)
                {
                    //1. LOAD DATA + CHECKING DATA (WITHOUT FILTER)
                    for (lineskip=0; lineskip<10; lineskip++)     // Skips the first 10
lines which are the header of csv files
                    {
                        skip_line (pointfile);
                    }
                    read_line (pointfile,voltstr, timestr);
                    t0=atof(timestr);
                    while (!feof(pointfile))      // Reading the lines of the voltage
input file
                    {
                        if (voltstr[0]!='\0')
                        {
                            snum=atof(voltstr);
                            tnum=atof(timestr);
                            if(snum<0)
                                negval++;
                            else
                                posval++;
                            vfilter[samplesp]=snum;
                            tfilter[samplesp]=tnum;
                            samplesp++;
                            read_line (pointfile,voltstr, timestr);
                        }
                        tlast=tfilter[samplesp-1];
                    }
                    samplesp=samplesp+3;

                    samplesp=datacheck(posval, negval, samplesp, tlast, pointfile);

                    tlast=tfilter[samplesp];

                    //2. DATA FILTER BANDWIDTH (10 MHz OR 20 MHz DEPENDING ON BIT RATE)
                    if (rate==106 || rate==212 || rate==424 || rate==848)
                    {
                        freq1=8.56e6/(1/(2*((tlast-t0)/(samplesp-1))));
                        freq2=18.56e6/(1/(2*((tlast-t0)/(samplesp-1))));
                    }
                    else if (rate==1700 || rate==3400 || rate==6800)
                    {
                        freq1=6.06e6/(1/(2*((tlast-t0)/(samplesp-1))));
                        freq2=21.06e6/(1/(2*((tlast-t0)/(samplesp-1))));
                    }

                    butterworth_coeffs(freq1, freq2, as, bs);
                    b1=bs[0];
                    b2=bs[1];
                    b3=bs[2];
                    b4=bs[3];
```

```
                    b5=bs[4];
                    a1=as[0];
                    a2=as[1];
                    a3=as[2];
                    a4=as[3];
                    a5=as[4];

                    for (fi=0; fi<samplesp; fi++)
                    {
                         if (fi<7 || fi>samplesp-7)
                              voutput[fi]=0;
                         else
                              voutput[fi]=(b1*vfilter[fi]+b2*vfilter[fi-1]+b3*vfilter[fi-2]+
                              b4*vfilter[fi-3]+b5*vfilter[fi-4]-a2*voutput[fi-1]-
                              a3*voutput[fi-2]-a4*voutput[fi-3]-a5*voutput[fi-4])/a1;
                    }

                    rewind (pointfile);
                    lineskip=0;
                    for (lineskip=0; lineskip<10; lineskip++) // Skips the first 10 lines
(header of csv files)
                    {
                         skip_line (pointfile);
                    }
                    for (fi=0; fi<(samplesp-7); fi++)     // Reading the lines of the
voltage input file
                    {
                         val=voutput[fi];
                         read_line (pointfile,voltstr,timestr);
                         fprintf(input_u2,"%s,%f\n",timestr,val);
                         length++;
                    }

                    //3. HILBERT TRANSFORM AND THE COMPLEX ENVELOPE
                    rewind(input_u2);
                    hilbert("pre_Hilbert..txt");      // performs Hilbert transform

                    poutput=fopen("output.txt","r"); // Hilbert transform output vector
                    read_line (poutput,voltstr,timestr);
                    tini=atof(timestr);
                    rewind (poutput);

                    if(poutput!=NULL)
                    {
                         while (!feof(poutput)) // Reading the lines of the voltage
input file */
                         {
                              read_line (poutput,voltstr,timestr);
                              if (timestr[0]!='\0')
                              {
                                   snum=atof(voltstr);
                                   voutput[samples]=snum;
                                   t=atof(timestr);
                                   toutput[samples]=t;
                                   samples++;//-->CS // Same variable as the one in
Hmaxfinder
                                   tfin=t;
                              }
                         }
                    }
                    else
                         fprintf(stdout,"Error in Hilbert transform\n");
```

```
                    fclose(poutput);

                    //4. USING A SMOOTHING FILTER (MOV. AVG) TO REDUCE THE NOISE
                    filterlength=3;
                    length_total=envfilt(rate, voutput, toutput, filterlength, tini, tfin,
samples, envelope);

                    //5. 100% OF H_INITIAL
                    Hmaxfinder(envelope, &Hmax, &Hmin, length_total);

                    //6. COMPUTING THE ISO BASED TIMES
     tfinder(type,envelope,toutput,tini,Hmax,Hmin,rate,length_total,timesp);

                    //7. M_min FOR BIT RATES OF fc/8, fc/4 AND fc/2
                    if (rate==1700 || rate==3400 || rate==6800)
                         Mminfinder(envelope, Hmax, Hmin, &HmaxVHBR, timesp, length_
total);

                    //8. CHECKING FOR ISO DEFINED MONOTONY
                    if (rate==106 || rate==212 || rate==424 || rate==848)
                         monocheck(envelope, toutput, Hmax, timesp, rate, type);
                    out_i=0;
                    while (out_i<MAX_SAMPLES) // Finds how many zeros are at the
beginning of vector envelope
                    {
                         if (envelope[out_i]==0 && flag_out==0)
                         {
                              sample_ini=out_i;
                              tini=toutput[sample_ini+1];
                         }

                         if (envelope[out_i]!=0)
                         {
                              flag_out=1;
                              sample_end=out_i;
                              tfin=toutput[sample_end];
                         }
                         out_i++;
                    }

                    samples=sample_end-sample_ini-1;      //==>JS

                    for (out_i=0; out_i<samples; out_i++)
                    {
                         voutput[out_i]=envelope[out_i+sample_ini+1];
                         toutput[out_i]=toutput[out_i+sample_ini+1];
                    }
                    for (out_i=samples+1; out_i<MAX_SAMPLES; out_i++)
                    {
                         voutput[out_i]=0.0;
                         toutput[out_i]=0.0;
                    }

                    tini=toutput[0];
                    tfin=toutput[samples];

                    //9. OVERSHOOT OF THE READER
                    fprintf (stdout,"\n"); // 2nd set of functions, "New Line" printed
for debug purposes
                    if (rate==106 || rate==212 || rate==424 || rate==848)
                    {
                         filterlength=3;
```

```
                        length_total=envfilt(rate, voutput, toutput, filterlength, tini,
tfin, samples, envelope); // 2nd Filtering to find the alternate envelope
                        Hmaxfinder(envelope, &Hmax2, &Hmin2, length_total);

      tfinder(type,envelope,toutput,tini,Hmax2,Hmin2,rate,length_total, timesp2);
                  monocheck(envelope, toutput, Hmax2, timesp2, rate,type);
            // The parameters of the alternate envelope are calculated
                  overshoot(timesp2,Hmax2,envelope,toutput,rate,type,samples,shootrea-
                  der2); // This time the over- and undershoots are found
                  }
                  else if (rate==1700 || rate==3400 || rate==6800)
                  {
                        filterlength=3;
                        length_total=envfilt(106, voutput, toutput, filterlength, tini,
tfin, samples, envelope);   // 2nd Filtering to find the alternate envelope
                        Hmaxfinder(envelope, &Hmax2, &Hmin2, length_total);
                        overshoot(timesp,Hmax,envelope,toutput,rate,type,samples,
                        shootreader2); // This time the over- and undershoots are
found
                  }

                  //10. MODULATION
                  m=modulation(type, Hmax, timesp->h);
                  if((type=='B') && (rate==1700 || rate==3400 || rate==6800))
                        mmin=modulation(type,HmaxVHBR, timesp->bVHBR);

                  //11. DISPLAY
                  display(type, rate, shootreader2, timesp, Hmax, m, mmin);
            }
            else if (pointfile==NULL || input_u2!=NULL)
                  fprintf(stdout,"file(s) could not be opened \n");

            fclose(pointfile);
            fclose(input_u2);
      }
}

else
      fprintf(stdout, "Memory could not be allocated");

      free (voutput);
      free (toutput);
      free (envelope);
      free (vfilter);
      free (tfilter);
      free (timesp);
      free (timesp2);
      free (shootreader2);

      return 0;
}
```

(Измененная редакция. Изм. А4:2012.)

Приложение F
(справочное)

Программа для вычисления спектра

Следующая программа, написанная на языке С, дает пример вычисления величины спектра у PICC.

```
/*****************************************************************/
/***  This program calculates the Fourier coefficients         ***/
/***  of load modulated voltage of a PICC according            ***/
/***  the ISO/IEC 10373-6 Test methods                         ***/
/***  The coefficients are calculated at the frequencies:      ***/
/***   Carrier:    Fcm (=13.5600 for 13.56 MHz)                ***/
/***   Upper sideband:   Fcm + fs                              ***/
/***   Lower sideband:   Fcm - fs                              ***/
/***  fs is the subcarrier frequency and its value is:         ***/
/***  Fcm/16 for bit rates up to fc/16, Fcm/8 for a bit rate   ***/
/***  of fc/8, Fcm/4 for a bit rate of fc/4 or Fcm/2 for a     ***/
/***  bit rate of fc/2                                         ***/
/*****************************************************************/
/***  Input:                                                   ***/
/***  File in CSV Format containing a table of two             ***/
/***  columns (time and test PCD output voltage vd, clause 7)  ***/
/***                                                           ***/
/***  data format of input-file:                               ***/
/***  --------------------------                               ***/
/***  - one data-point per line:                               ***/
/***    (time[seconds], sense-coil-voltage[volts])             ***/
/***  - contents in ASCII, no headers                          ***/
/***  - data-points shall be equidistant in time               ***/
/***  - modulation waveform centered                           ***/
/***    (max. tolerance: half of subcarrier cycle)             ***/
/***                                                           ***/
/***                                                           ***/
/***  example for spreadsheet file (start in next line):       ***/
/***    (time)       (voltage)                                 ***/
/***  3.00000e-06,1.00                                         ***/
/***  3.00200e-06,1.01                                         ***/
/***  .....                                                    ***/
/*****************************************************************/
/*** RUN:                                                      ***/
/*** "exefilename" [filename[.csv] SubcarrierCode ]            ***/
/*****************************************************************/
/*** ISO/IEC 10373-6 DFT CALCULATION                          ***/
/*** Version history:                                         ***/
/*** JUL 2000, version 1.1: original published version         ***/
/*** APR 2008, version 2.0: add the Bartlett window           ***/
/*** NOV 2008, version 2.1: published version with revision   ***/
/*** SEP 2010, version 3.0: support higher subcarrier freq.   ***/
/*****************************************************************/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#define MAX_SAMPLES 50000
#define MAX_POINTS 500
#define MAX_MOYENNE 200

double pi;    /* pi=3.14.... */
```

```
/* Array for time and sense coil voltage vd */
double vtime[MAX_SAMPLES]; /* time array                              */
double vd[MAX_SAMPLES];    /* Array for different coil voltage */


/*****************************************************************/
/***  Read CSV File Function                                     ***/
/***                                                             ***/
/***  Description:                                               ***/
/***  This function reads the table of time and sense coil       ***/
/***  voltage from a File in CSV Format                          ***/
/***                                                             ***/
/***  Input: filename                                            ***/
/***                                                             ***/
/***  Return: Number of samples (sample Count)                   ***/
/***          0 if an error occurred                             ***/
/***                                                             ***/
/***  Displays Statistics:                                       ***/
/***                                                             ***/
/***  Filename, SampleCount, Sample rate, Max/Min Voltage        ***/
/*****************************************************************/

int readcsv(char* fname)
{
     double a,b;
     double max_vd,min_vd;
     int i;
     FILE     *sample_file;

     /************ Open File ***********************************/
     if (!strchr(fname, '.'))   strcat(fname, ".csv");

     if ((sample_file = fopen(fname, "r"))== NULL)
        {
               printf("Cannot open input file %s.\n",fname);
               return 0;
        }
     /*************************************************************/
     /* Read CSV File                                             */
     /*************************************************************/
     max_vd=-1e-9F;
     min_vd=-max_vd;
     i=0;

     while (!feof(sample_file))
        {
        if (i>=MAX_SAMPLES)
               {
               printf("Warning: File truncated !!!\n");
               printf("To much samples in file %s\b\n",fname);
               break;
               }
          fscanf(sample_file,"%Lf,%Lf\n", &a, &b);
          vtime[i] = a;
          vd[i] = b;
          if (vd[i]>max_vd) max_vd=vd[i];
          if (vd[i]<min_vd) min_vd=vd[i];
          i++;
          }
     fclose(sample_file);

     /************ Displays Statistics ************************/
```

```
    printf("\n************************************************\n");

    printf("\nStatistics: \n");
    printf(" Filename   : %s\n",fname);
    printf(" Sample count : %d\n",i);
    printf(" Sample rate  : %1.0f MHz\n",1e-6/(vtime[1]-vtime[0]));
    printf(" Max(vd)    : %4.0f mV\n",max_vd*1000);
    printf(" Min(vd)    : %4.0f mV\n",min_vd*1000);
    return 1;
}/*************** End ReadCsv **************/


/*****************************************************************/
/***  DFT : Discrete Fourier Transformation                    ***/
/*****************************************************************/
/***  Description:                                             ***/
/***  This function calculate the Fourier coefficient          ***/
/***                                                           ***/
/***  Input: Number of samples                                 ***/
/***         Carrier divider of the subcarrier                 ***/
/***                                                           ***/
/***  Global Variables:                                        ***/
/***                                                           ***/
/***  Displays Results:                                        ***/
/***                                                           ***/
/***   Carrier coefficient                                     ***/
/***   Upper sideband coefficient                              ***/
/***   Lower sideband coefficient                              ***/
/***                                                           ***/
/*****************************************************************/
void dft(int count, int CarrierDivider)
{
    double c0_real,c0_imag,c0_abs,c0_phase;
    double c1_real,c1_imag,c1_abs,c1_phase;
    double c2_real,c2_imag,c2_abs,c2_phase;
    int N_data,center,start;
    double w0,wu,wl;
    double Wb;          /* Bartlett window coefficient */

    int i,k;

    double fc;          /* add variable for carrier frequency */

    fc=13.56e6;

    w0=(double)(fc*2.0)*pi; /* carrier 13.56 MHz */
    wu=(double)(1.0+1.0/CarrierDivider)*w0; /* upper sideband 14.41 MHz */
    wl=(double)(1.0-1.0/CarrierDivider)*w0; /* lower sideband 12.71 MHz */
    c0_real=0; /* real part of the carrier fourier coefficient */
    c0_imag=0; /* imag part of the carrier fourier coefficient */
    c1_real=0; /* real part of the up. sideband fourier coefficient */
    c1_imag=0; /* imag part of the up. sideband fourier coefficient */
    c2_real=0; /* real part of the lo. sideband fourier coefficient */
    c2_imag=0; /* imag part of the lo. sideband fourier coefficient */

    center=(count+1)/2; /* center address */

    /********** signal selection ****************************/

    /* Number of samples for six subcarrier periods */

    N_data=(int)(0.5+6.0F*CarrierDivider/(vtime[2]-vtime[1])/fc);
```

```
/* Note: (vtime[2]-vtime[1]) is the scope sample rate */

start=center - (int) N_data / 2;

/****************** DFT ******************************/

for( i=0;i<=N_data-1;i++)
{
/* Bartlett window */
if ((N_data & 1) == 0)
{
     /* N_data is even */
     if (i < (int) N_data /2)
     {
     Wb=2.0F*i/(double)(N_data - 1);
     }
     else
     {
     Wb=2.0F*(N_data-i-1)/(double)(N_data - 1);
     }
}
else
{
     /*N_data is odd */
     if (i < (int) N_data /2)
     {
     Wb=2.0F*i/(double)(N_data - 1);
     }
     else
     {
     Wb=2.0F-2.0F*i/(double)(N_data - 1);
     }
}

k=i+start;

c0_real=c0_real+vd[k]*(double)cos(w0*vtime[k])*Wb;
c0_imag=c0_imag+vd[k]*(double)sin(w0*vtime[k])*Wb;
c1_real=c1_real+vd[k]*(double)cos(w1*vtime[k])*Wb;
c1_imag=c1_imag+vd[k]*(double)sin(w1*vtime[k])*Wb;
c2_real=c2_real+vd[k]*(double)cos(w1*vtime[k])*Wb;
c2_imag=c2_imag+vd[k]*(double)sin(w1*vtime[k])*Wb;
}

/****************** DFT scale ***************************/

c0_real=4.0F*c0_real/(double) N_data;
c0_imag=4.0F*c0_imag/(double) N_data;
c1_real=4.0F*c1_real/(double) N_data;
c1_imag=4.0F*c1_imag/(double) N_data;
c2_real=4.0F*c2_real/(double) N_data;
c2_imag=4.0F*c2_imag/(double) N_data;
/* Note: 4.0F includes the correction coef. of the bartlett window */

/***************** absolute fourier coefficient **********/
c0_abs=(double)sqrt(c0_real*c0_real + c0_imag*c0_imag);
c1_abs=(double)sqrt(c1_real*c1_real + c1_imag*c1_imag);
c2_abs=(double)sqrt(c2_real*c2_real+c2_imag*c2_imag);

/************** Phase of fourier coefficient **********/
c0_phase=(double)atan2(c0_imag,c0_real);
c1_phase=(double)atan2(c1_imag,c1_real);
```

```
    c2_phase=(double)atan2(c2_imag,c2_real);

    /************** Result Display *************************/

    printf("\n\nResults: \n");
    printf("Subcarrier frequency = fc/%d\n",CarrierDivider);
    printf("Carrier ");
    printf("Abs: %7.3fmV ",1000*c0_abs);
    printf("Phase: %3.0fdeg\n",c0_phase/pi*180);
    printf("Upper sideband ");
    printf("Abs: %7.3fmV ",1000*c1_abs);
    printf("Phase: %3.0fdeg\n",c1_phase/pi*180);
    printf("Lower sideband ");
    printf("Abs: %7.3fmV ",1000*c2_abs);
    printf("Phase: %3.0fdeg\n\n",c2_phase/pi*180);
    printf("\n*************************************************\n");
    return;
    }/***************** End DFT ***************/


/*************************************************************/
/***  MAIN Program                                         ***/
/*************************************************************/
int main(unsigned short paramCount,char *paramList[])
{
    char fname[256];
    unsigned int sample_count;
    int Dsi; /* Subcarrier frequency code */
    int Ds[4]={2, 4, 8, 16}; /* Carrier divider */

    pi = (double)atan(1.0)*4; /* calculate pi */


    printf("\n*************************************************\n");
    printf("\n**** ISO/IEC 10373-6 PICC Test-Program     ****\n");
    printf("\n**** Version: 3.0 SEPTEMBER 2010           ****\n");
    printf("\n****                       ****\n");
    printf("\n*************************************************\n");
    /************* No Input Parameter           *****************/
    if (paramCount==1)
    {
        printf("\nCSV File name :");
        scanf("%s",fname);
    printf("\nSubcarrier frequency code [1=fc/2, 2=fc/4, 3=fc/8 or 4=fc/16]] :");
    scanf("%d",&Dsi);
    }
    else
    {
        /************* Input Parameter Loop ****************/
        strcpy(fname,paramList[1]);
    if (!strchr(fname, '.')) strcat(fname, ".csv");
    if (paramCount>2)
        {
            Dsi=atoi(paramList[2]);
        }
        else
        {
            Dsi=4; /*** for backward compatibility ***/
        }
    }
    if (!strchr(fname, '.')) strcat(fname, ".csv");
    if (!(sample_count=readcsv(fname))) return 0;
```

```
    if ((Ds1 > 0) & (Ds1 < 5))
    {
        dft(sample_count,Ds[Ds1-1]);
    }
    else
    {
        printf("\nError: Subcarrier frequency code shall be 1,2,3,or 4\n");
    }

    return 0;
}/*********************** End Main ***************************/
```

(Измененная редакция. Изм. А4:2012.)

Приложение G
(обязательное)

Дополнительные методы испытаний PICC

G.1 Оборудование для испытаний PICC и дополнительные приспособления

В настоящем разделе определены испытательное оборудование и испытательные схемы для проверки работы PICC на соответствие требованиям ИСО/МЭК 14443-3:2011. Испытательное оборудование включает в себя:

- калибровочную катушку (см. 5.2);
- испытательную установку PCD (см. 5.3);
- цифровой импульсный осциллограф (см. 5.1.1).

Необходимо проследить, чтобы на результаты не влияло радиочастотное действие испытательной схемы.

G.1.1 Эмуляция протокола на I/O

Оборудование для испытаний PICC должно быть способно эмулировать протоколы типа A и типа B, которые необходимы для испытаний PICC.

G.1.2 Генерация временных соотношений знака на I/O в режиме приема

Оборудование для испытаний PICC должно иметь возможность генерировать поток битов на I/O в соответствии с ИСО/МЭК 14443-3:2011. Временные параметры: длина стартового бита, разграничительный интервал, ширина бита, разграничительный интервал запроса, начало ширины кадра, конец ширины кадра — должны быть с изменяемой конфигурацией.

G.1.3 Измерение и контроль радиочастотного протокола на I/O

Оборудование для испытаний PICC должно иметь возможность измерять и контролировать временные соотношения низкого и высокого логических состояний радиочастотной линии входа/приема, связанных с тактовой частотой. Оборудование для испытаний PICC должно быть способно отслеживать поднесущую PICC.

G.1.4 Анализ протоколов

Оборудование для испытаний PICC должно быть способно анализировать поток битов на линии I/O в соответствии с протоколом типа A и типа B, как определено в ИСО/МЭК 14443-3:2011 и ИСО/МЭК 14443-4:2008, и извлекать поток логических данных для дальнейшего анализа.

G.1.5 RFU-поля

Зарезервированные для использования в будущем поля (RFU-поля) следует постоянно контролировать во время испытания, а также всегда проверять на содержание присвоенного значения по умолчанию. Результаты испытания считают неудовлетворительными, а испытуемая PICC должна быть признана не соответствующей требованиям, если в произвольное время RFU-поле не установлено в состояние значения по умолчанию.

G.1.5.1 RFU-значения

Функциональные поля следует постоянно контролировать во время испытания, а также всегда проверять на содержание только стандартных или проприетарных функциональных значений, установленных в стандартах. Результаты испытания считают неудовлетворительными, а испытуемая PICC должна быть признана не соответствующей требованиям, если в произвольное время функциональное поле не было установлено в состояние вышеуказанных значений (и таким образом было установлено в состояние RFU-значения или значения, ограниченного для применения).

G.1.5.2 Измерения временных соотношений

Оборудование для испытаний PICC должно непрерывно контролировать следующие форматы кадра и временные значения.

Для PICC типа A:

- время задержки кадра от PCD к PICC [см. ИСО/МЭК 14443-3:2011 (подпункт 6.2.1.1)];
- форматы кадра по ИСО/МЭК 14443-3:2011 (пункт 6.2.3);
- время ожидания кадра [см. ИСО/МЭК 14443-4:2008 (подраздел 7.2)].

Для PICC типа B:

- знак, формат кадра и временные соотношения по ИСО/МЭК 14443-3:2011 (подраздел 7.1);
- время ожидания кадра [см. ИСО/МЭК 14443-4:2008 (подраздел 7.2)].

Результаты испытания считают неудовлетворительными, а испытуемая карта должна быть признана не соответствующей требованиям, если хотя бы одно из перечисленных ограничений на временные соотношения не выполняется.

G.1.5.3 Протокол измерения временных соотношений

В протоколе заполняют таблицу G.59 для PICC типа A и/или таблицу G.60 для PICC типа B измеренными временными значениями.

G.2 Взаимосвязь методов испытаний и требований базового стандарта

В таблице G.1 перечислены испытания, применимые для PICC типа A.

В таблице G.2 перечислены испытания, применимые для PICC типа B.

В таблице G.3 перечислены испытания, применимые для PICC типа A и типа B.

PICC по ИСО/МЭК 14443-4:2008 также должны соответствовать ИСО/МЭК 14443-3:2011, их подвергают испытаниям на соответствие требованиям обоих указанных стандартов, в зависимости от их типа.

PICC, соответствующие ИСО/МЭК 14443-3:2011, но не отвечающие требованиям ИСО/МЭК 14443-4:2008, находящиеся в состоянии ACTIVE или ACTIVE* (см. G.3.3.7, G.3.3.12 и G.4.4.6) могут отвечать любым кадром (включая режим Mute) на кадры, не соответствующие требованиям ИСО/МЭК 14443-3:2011.

Таблица G.1 — Методы испытаний для логической операции протокола PICC типа A

Метод испытаний по ИСО/МЭК 10373-6		Соответствующее требование	
Подраздел	Наименование	Базовый стандарт	Структурный(е) элемент(ы)
G.3.2	Опрос	ИСО/МЭК 14443-3:2011	5
G.3.3	Испытания переходов состояний PICC типа A	ИСО/МЭК 14443-3:2011	6.3, 6.4, 6.5
G.3.4	Обработка антиколлизии типа A	ИСО/МЭК 14443-3:2011	6.4.2
G.3.5	Обработка RATS	ИСО/МЭК 14443-4:2008	5.6.1.2
G.3.6	Обработка PPS-запроса	ИСО/МЭК 14443-4:2008	5.6.2.2
G.3.7	Обработка FSD	ИСО/МЭК 14443-4:2008	5.1

Таблица G.2 — Методы испытаний для логической операции протокола PICC типа B

Метод испытаний по ИСО/МЭК 10373-6		Соответствующее требование	
Подраздел	Наименование	Базовый стандарт	Структурный(е) элемент(ы)
G.4.2	Опрос	ИСО/МЭК 14443-3:2011	5
G.4.3	Прием PICC	ИСО/МЭК 14443-3:2011	7.1
G.4.4	Испытания переходов состояний PICC типа B	ИСО/МЭК 14443-3:2011	7.4—7.12
G.4.5	Обработка антиколлизии типа B	ИСО/МЭК 14443-3:2011	7.4—7.12
G.4.6	Обработка ATTRIB	ИСО/МЭК 14443-3:2011	7.10
G.4.7	Обработка максимального размера кадра	ИСО/МЭК 14443-3:2011	7.10.4

Таблица G.3 — Методы испытаний для логической операции PICC типа A или B

Метод испытаний по ИСО/МЭК 10373-6		Соответствующее требование	
Подраздел	Наименование	Базовый стандарт	Структурный(е) элемент(ы)
G.5.2	Реакция PICC на сценарии по ИСО/МЭК 14443-4	ИСО/МЭК 14443-4:2008	7
G.5.3	Обработка операций по обнаружению PICC ошибок	ИСО/МЭК 14443-4:2008	7.5.7[1]
G.5.4	Реакция PICC на CID	ИСО/МЭК 14443-4:2008	7.1.1.2
G.5.5	Реакция PICC на NAD	ИСО/МЭК 14443-4:2008	7.1.1.3

G.3 Метод испытаний для инициализации PICC типа A

G.3.1 Ведение

По результатам испытаний, описанных в данном разделе, определяют, соответствует ли PICC типа A требованиям ИСО/МЭК 14443-3:2011 и требованиям к последовательности активации ИСО/МЭК 14443-4:2008 (раздел 5).

[1] В ИСО/МЭК 10373-6:2011 указан пункт 7.5.6 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

Если соответствие требованиям ИСО/МЭК 14443-4:2008 не нужно, то испытания, содержащие команды ИСО/МЭК 14443-4, не применяют.

G.3.2 Сценарий G.1: Опрос

G.3.2.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа А при получении команды REQA требованиям ИСО/МЭК 14443-3:2011 (раздел 5).

G.3.2.2 Процедура

Выполнить следующие шаги для трех различных значений напряженности рабочего поля: 1,5, 4,5 и 7,5 А/м (среднеквадратичное значение).

a) Поместить PICC в пределах поля и осуществить ее настройку.

b) Переключить радиочастотное рабочее поле в состояние «выключено» на минимальное время для возврата PICC в исходное состояние [см. ИСО/МЭК 14443-3:2011 (подраздел 5.4)].

c) Переключить радиочастотное рабочее поле в состояние «включено».

d) Обеспечить задержку в 5 мс и послать действительный кадр команды REQA.

e) Зафиксировать наличие и содержание ответа PICC.

f) Переключить радиочастотное рабочее поле в состояние «выключено» на минимальное время для возврата PICC в исходное состояние [см. ИСО/МЭК 14443-3:2011 (подраздел 5.4)].

g) Переключить радиочастотное рабочее поле в состояние «включено».

h) Выдержать 5 мс и послать действительный кадр команды REQB (используя модуляцию типа B и кодирование бит).

i) Выдержать 5 мс и послать действительный кадр команды REQA.

j) Зафиксировать наличие и содержание ответа PICC.

G.3.2.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.4.

Таблица G.4 — Критерии оценки результата испытания сценария G.1: Опрос

Объяснение	Результат испытаний
Только если ответ PICC представляет собой действительный ATQA на обоих этапах e) и j)	PASS
Если ответ PICC не является действительным ATQA на любом из этапов e) или j)	FAIL

G.3.3 Испытания переходов состояний PICC типа А

G.3.3.1 Область применения

В данных испытаниях проверяют, соответствует ли реализация диаграммы состояний PICC типа А требованиям ИСО/МЭК 14443-3:2011 (подраздел 6.3).

G.3.3.2 Общий принцип испытаний

Для исчерпывающего испытания конечного автомата PICC типа А должна быть проверена корректность каждого возможного перехода состояний для каждого состояния. Проверку конкретного состояния, использующего конкретный переход состояний, проводят следующим образом.

Сначала необходимо осуществить сброс PICC и поставить ее в исходное состояние испытания (TIS). Это одно из состояний StateSet, в котором следует проверять переходы (T). Далее следует выполнить переход (T) из TransitionSet. После выполнения перехода состояний следует проверить, находится ли PICC в ожидаемом целевом состоянии (TTS). Трудность заключается в том, как выполнить эту проверку, поскольку непосредственно проконтролировать конечный автомат PICC невозможно. Решение этой проблемы — выполнить некоторые дополнительные переходы состояний и проверить ответ PICC. Переходы для этой цели выбирают таким образом, чтобы состояние можно было определить по ответам PICC с максимально возможной точностью.

G.3.3.2.1 Функции для приведения PICC в исходное состояние испытания (TIS)

Приведение PICC в состояние TIS осуществляют с помощью последовательности команд перехода, определенной в следующей таблице. Общий метод заключается в следующем.

Для того чтобы привести PICC в состояние TIS, находят соответствующую последовательность переходов состояний в таблице G.5. Далее последовательно применяют переходы состояний, описанные в графе «Последовательность переходов состояний», находя соответствующие команды в таблице G.6. Следует всегда проверять содержание и целостность ответа PICC.

Таблица G.5 — Таблица последовательности переходов состояний

TIS	Последовательность переходов состояний
POWER-OFF	—
IDLE	POWER-OFF → IDLE
READY(1)	POWER-OFF → IDLE → READY(1)
READY(2)	POWER-OFF → IDLE → READY(1) → READY(2)
READY(3)	POWER-OFF → IDLE → READY(1) → READY(2) → READY(3)
ACTIVE	POWER-OFF → IDLE→ READY(1) →...→ READY(CascadeLevels) → ACTIVE
PROTOCOL	POWER-OFF → IDLE → READY(1) →...→ READY(CascadeLevels) → ACTIVE → PROTOCOL
HALT	POWER-OFF → IDLE → READY(1) →...→ READY(CascadeLevels) → ACTIVE → HALT
READY*(1)	POWER-OFF → IDLE → READY(1) →...→ READY(CascadeLevels) → ACTIVE → HALT → READY*(1)
READY*(2)	POWER-OFF → IDLE → READY(1) →...→ READY(CascadeLevels) → ACTIVE → HALT → READY*(1) → READY*(2)
READY*(3)	POWER-OFF → IDLE → READY(1) →...→ READY(CascadeLevels) → ACTIVE → HALT → READY*(1) → READY*(2) → READY*(3)
ACTIVE*	POWER-OFF → IDLE → READY(1) →...→ READY (CascadeLevels) → ACTIVE → HALT → READY*(1) →...→ READY*(CascadeLevels) → ACTIVE*

Таблица G.6 — Таблица переходов состояний

Состояние » Следующее состояние	Оборудование для испытаний PICC		PICC
POWER-OFF → IDLE	Power On (радиочастотное поле включено)	→	
		←	Mute
IDLE → READY(1)	REQA	→	
		←	ATQA
READY(1) → READY(2)	SELECT(1)[a)]	→	
		←	SAK (cascade)
READY(2) → READY(3)	SELECT(2)[a)]	→	
		←	SAK (cascade)
READY(CascadeLevels) → ACTIVE	SELECT (CascadeLevels)[a)]	→	
		←	SAK (complete)
ACTIVE → PROTOCOL	RATS(0,0)	→	
		←	ATS
ACTIVE → HALT	HLTA	→	
		←	Mute
HALT → READY*(1)	WUPA	→	
		←	ATQA
READY*(1) → READY*(2)	SELECT(1)	→	
		←	SAK (cascade)
READY*(2) → READY*(3)	SELECT(2)	→	
		←	SAK(cascade)
READY*(CascadeLevels) → ACTIVE*	SELECT (CascadeLevels)	→	
		←	SAK (complete)

[a)] Любая команда SELECT может предшествовать команде антиколлизии для получения UID PICC, особенно для произвольного UID.

G.3.3.2.2 Функции для проверки действительности целевого состояния испытания TTS

В таблице G.7 описаны переходы состояний, которые используют для проверки, находится ли PICC в состоянии S. Содержание ответа PICC (т. е. ATQA, SAK ...) должно быть тщательно проверено на соответствие ИСО/МЭК 14443-3:2011 и ИСО/МЭК 14443-4:2008. Следует обратить внимание, что эти проверки могут привести к изменению состояния PICC.

Состояния READY(I)/READY*(I) и ACTIVE/ACTIVE* не могут различаться при однократном выполнении испытания. Для того чтобы отличить состояния с «*» от состояний без «*», необходимо выполнить следующие шаги:

a) повторно выполнить испытание без проверки TTS;

b) послать команду REQA. Ответом PICC должен быть Mute;

c) послать команду REQA;

d) если ответом PICC является Mute, то состоянием PICC было состояние с «*»;

e) в противном случае PICC была в состоянии без «*».

Состояние HALT не отличается от состояний READY*(I) и ACTIVE* при однократном выполнении испытания. Для того чтобы отличить состояние HALT, необходимо выполнить следующие этапы:

f) повторно выполнить испытание без проверки TTS;

g) послать команду WUPA. Ответом PICC должен быть ATQA.

Таблица G.7 — Проверка TTS

Состояние S	Оборудование для испытаний PICC		PICC
IDLE	REQA	→	
		←	ATQA
READY(I), I < CascadeLevels	SELECT(I)[a]	→	
		←	SAK (cascade)
READY(I), I = CascadeLevels	SELECT(I)[a]	→	
		←	SAK (complete)
ACTIVE	RATS (0,0)	→	
		←	ATS
PROTOCOL	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
HALT	REQA	→	
		←	Mute
	WUPA	→	
		←	ATQA
READY*(I), I < CascadeLevels	SELECT (I)	→	
		←	SAK (cascade)
READY*(I), I = CascadeLevels	SELECT (I)	→	
		←	SAK (complete)
ACTIVE*	RATS(0,0)	→	
		←	ATS

[a] Любая команда SELECT может предшествовать команде антиколлизии для получения UID PICC, особенно для произвольного UID.

Примечание — Номер блоков может быть 0 или 1 в зависимости от правил нумерации блоков, определенных в ИСО/МЭК 14443-4:2008 (пункт 7.5.4)[1].

(Введено дополнительно. Изм. А3:2012.)

[1] В ИСО/МЭК 10373-6:2011/Изм. А3:2012 указан пункт 7.5.3 без учета изменения нумерации пунктов, введенного изменением А1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

G.3.3.3 Сценарий G.2: Поведение PICC типа A в состоянии IDLE

G.3.3.3.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа A в состоянии IDLE требованиям ИСО/МЭК 14443-3:2011 (пункт 6.3.2).

G.3.3.3.2 Процедура

Выполнить следующие шаги для каждой строки таблицы G.8.

a) Привести PICC в состояние IDLE.

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Если ответом PICC не является Mute, то проверить, соответствует ли время задержки кадра PICC значению, указанному в графе «FDT».

e) Проверить, находится ли PICC в состоянии TTS.

Таблица G.8 — Переходы из состояния IDLE

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
REQA	REQA	→			READY(1)
		←	ATQA	1172/*fc*	
WUPA	WUPA	→			READY(1)
		←	ATQA	1236/*fc*	
HLTA	HLTA	→			IDLE
		←	Mute		
AC[c)]	('93' NVB $UIDTX_1$[[1..n_1]])[a)]	→			IDLE
		←	Mute		
nAC[d)]	('93' NVB ~$UIDTX_1$[[1.. n_1]])[a)]	→			IDLE
		←	Mute		
SELECT[c)]	SELECT(1)	→			IDLE
		←	Mute		
nSELECT[d)]	('93 70' ~$UIDTX_1$[[1..32]] BCC CRC_A)	→			IDLE
		←	Mute		
RATS	RATS(0,0)	→			IDLE
		←	Mute		
PPS	PPS(0,0,0)	→			IDLE
		←	Mute		
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→			IDLE
		←	Mute		
DESELECT	S(DESELECT)	→			IDLE
		←	Mute		
Состояние ошибки	('26')[b)]	→			IDLE
		←	Mute		

[a)] Пусть $1 \le n_1 \le 32$.

[b)] Значение передается в стандартном кадре и не передается в коротком кадре.

[c)] Данное испытание пропускают для PICC, использующих произвольный UID.

[d)] Для PICC, использующих произвольный UID, данное испытание выполняют с фиксированным произвольно выбранным UID.

G.3.3.3.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.9.

Таблица G.9 — Критерии оценки результата испытания поведения PICC типа A в состоянии IDLE

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.3.3.4 Сценарий G.3: Поведение PICC типа A в состоянии READY(1)

G.3.3.4.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа A в состоянии READY на каскадном уровне 1 требованиям ИСО/МЭК 14443-3:2011 (пункт 6.3.3).

G.3.3.4.2 Процедура

Выполнить следующие шаги для всех PICC и каждой строки таблицы G.10.

a) Привести PICC в состояние READY(1).

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Если ответом PICC не является Mute, то проверить, соответствует ли время задержки кадра PICC значению, указанному в графе «FDT».

e) Проверить, находится ли PICC в состоянии TTS.

Таблица G.10 — Переходы из состояния READY(1)

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
REQA	REQA	→			IDLE
		←	Mute		
WUPA	WUPA	→			IDLE
		←	Mute		
HLTA	HLTA	→			IDLE
		←	Mute		
AC[g] (разделение после (0)b)	('93' NVB $UIDTX_1$[[1..n_1]])[a]	→			READY(1)
		←	Если n_1 = 32, то (BCC), иначе ($UIDTX_1$[[n_1+1..32]] BCC)[a]	1172/*fc*	
AC[g] (разделение после (1)b)	('93' NVB $UIDTX_1$[[1..n_2]])[b]	→			READY(1)
		←	Если n_2 = 32, то (BCC), иначе ($UIDTX_1$[[n_2+1..32]] BCC)[b]	1236/*fc*	
nAC[g] (неправильный UID)	('93' NVB ~$UIDTX_1$[[1..n_3]])[f]	→			IDLE
		←	Mute		
SELECT[g]	SELECT(1)	→			TTS[e]
		←	SAK[d]	FDT[c]	
nSELECT[g] (неправильный UID)	('93 70' ~$UIDTX_1$ BCC CRC_A)	→			IDLE
		←	Mute		
Состояние ошибки	('93 70' $UIDTX_1$ BCC ~CRC_A)	→			IDLE
		←	Mute		

Окончание таблицы G.10

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→			IDLE
		←	Mute		
DESELECT	S(DESELECT)	→			IDLE
		←	Mute		
RATS	RATS(0,0)	→			IDLE
		←	Mute		
PPS	PPS(0,0,0)	→			IDLE
		←	Mute		

[a] Пусть $1 \le n_1 \le 32$, $UIDTX_1[[n_1]] = 0$. Если такого числа нет, то данное испытание может быть пропущено.
[b] Пусть $1 \le n_2 \le 32$, $UIDTX_1[[n_2]] = 1$. Если такого числа нет, то данное испытание может быть пропущено.
[c] FDT принимает значение 1172/*fc* (~86,43 мкс), если последний бит = (0)b, и 1236/*fc* (~91,15 мкс), если последний бит = (1)b [см. предельные значения в ИСО/МЭК 14443-3:2011 (подпункт 6.2.1.1)].
[d] Каскадный бит SAK должен быть (0)b для PICC с одиночным размером UID и (1)b для PICC с двойным и тройным размером UID.
[e] PICC с одиночным размером UID должна быть в состоянии ACTIVE, PICC с двойным и тройным размером UID должны быть в состоянии READY.
[f] Пусть $1 \le n_3 \le 32$.
[g] Любая команда AC или SELECT может предшествовать команде антиколлизии для извлечения UID PICC, особенно для произвольного UID.

G.3.3.4.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.11.

Таблица G.11 — Критерии оценки результата испытания поведения PICC типа A в состоянии READY(1)

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.3.3.5 Сценарий G.4: Поведение PICC типа A в состоянии READY(2)

G.3.3.5.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа A в состоянии READY на каскадном уровне 2 требованиям ИСО/МЭК 14443-3:2011 (пункт 6.3.3). Данное испытание предназначено только для PICC с двойным или тройным размером UID.

G.3.3.5.2 Процедура

Следует выполнить следующие шаги для всех PICC с двойным и тройным размером UID и для каждой строки таблицы G.12.

a) Привести PICC в состояние READY(2).

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Если ответом PICC не является Mute, то проверить, соответствует ли время задержки кадра PICC значению, указанному в графе «FDT».

e) Проверить, находится ли PICC в состоянии TTS.

Таблица G.12 — Переходы из состояния READY(2)

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
REQA	REQA	→			IDLE
		←	Mute		
WUPA	WUPA	→			IDLE
		←	Mute		
HLTA	HLTA	→			IDLE
		←	Mute		
AC[g] (разделение после (0)b)	('95' NVB $UIDTX_2$[[1..n_1]])[a]	→			READY(2)
		←	Если n_1 = 32, то (BCC), иначе ($UIDTX_2$[[n_1+1..32]] BCC)[a]	1172/*fc*	
AC[g] (разделение после (1)b)	('95' NVB $UIDTX_2$[[1..n_2]])[b]	→			READY(2)
		←	Если n_2 = 32, то (BCC), иначе ($UIDTX_2$[[n_2+1..32]] BCC)[b]	1236/*fc*	
nAC[g] (неправильный UID)	('95' NVB ~$UIDTX_2$[[1..n_3]])[f]	→			IDLE
		←	Mute		
SELECT[g]	SELECT(2)	→			TTS[e]
		←	SAK[d]	FDT[c]	
nSELECT[g] (неправильный UID)	('95 70' ~$UIDTX_2$ BCC CRC_A)	→			IDLE
		←	Mute		
Состояние ошибки	('95 70' $UIDTX_2$ BCC ~CRC_A)	→			IDLE
		←	Mute		
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→			IDLE
		←	Mute		
DESELECT	S(DESELECT)	→			IDLE
		←	Mute		
RATS	RATS(0,0)	→			IDLE
		←	Mute		
PPS	PPS(0,0,0)	→			IDLE
		←	Mute		

[a] Пусть 1 ≤ n1 ≤ 32, UIDTX2[[n1]] = 0. Если такого числа нет, то данное испытание может быть пропущено.

[b] Пусть 1 ≤ n2 ≤ 32, UIDTX2[[n2]] = 1. Если такого числа нет, то данное испытание может быть пропущено.

[c] FDT принимает значение 1172/*fc* (~86,43 мкс), если последний бит = (0)b, и 1236/*fc* (~91,15 мкс), если последний бит = (1)b [см. предельные значения в ИСО/МЭК 14443-3:2011 (подпункт 6.2.1.1)].

[d] Каскадный бит SAK должен быть (0)b для PICC с двойным размером UID и (1)b для PICC с тройным размером UID.

[e] PICC с двойным размером UID должна быть в состоянии ACTIVE; PICC с тройным размером UID должна быть в состоянии READY.

[f] Пусть 1 ≤ n3 ≤ 32.

[g] Любая команда AC или SELECT может предшествовать команде антиколлизии для извлечения UID PICC, особенно для произвольного UID.

G.3.3.5.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.13.

Таблица G.13 — Критерии оценки результата испытания поведения PICC типа A в состоянии READY(2)

Объяснение	Результат испытаний
Если PICC имеет одиночный размер UID	Не имеет отношения (N/A)
Если PICC имеет двойной или тройной размер UID и только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.3.3.6 Сценарий G.5: Поведение PICC типа A в состоянии READY(3)

G.3.3.6.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа A в состоянии READY(3) требованиям ИСО/МЭК 14443-3:2011 (пункт 6.3.3). Данное испытание предназначено только для PICC с тройным размером UID.

G.3.3.6.2 Процедура

Следует выполнить следующие шаги для всех PICC с тройным размером UID и для каждой строки таблицы G.14.

a) Привести PICC в состояние READY(3).

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Если ответом PICC не является Mute, то проверить, соответствует ли время задержки кадра PICC значению, указанному в графе «FDT».

e) Проверить, находится ли PICC в состоянии TTS.

Таблица G.14 — Переходы из состояния READY(3)

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
REQA	REQA	→			IDLE
		←	Mute		
WUPA	WUPA	→			IDLE
		←	Mute		
HLTA	HLTA	→			IDLE
		←	Mute		
AC[e] (разделение после (0)b)	('97' NVB $UIDTX_3$[[1..n_1]])[a]	→			READY(3)
		←	Если n_1 = 32, то (BCC), иначе ($UIDTX_3$[[n_1+1..32]] BCC)[a]	1172/*fc*	
AC[e] (разделение после (1)b)	('97' NVB $UIDTX_3$[[1..n_2]])[b]	→			READY(3)
		←	Если n_2 = 32, то (BCC), иначе ($UIDTX_3$[[n_2+1..32]] BCC)[b]	1236/*fc*	
nAC[e] (неправильный UID)	('97' NVB ~$UIDTX_3$[[1..n_3]])[d]	→			IDLE
		←	Mute		
SELECT[e]	SELECT(3)	→		FDT[c]	ACTIVE
		←	SAK(complete)		

Окончание таблицы G.14

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
nSELECT[e] (неправильный UID)	('97 70' ~UIDTX$_3$ BCC CRC_A)	→			IDLE
		←	Mute		
Состояние ошибки	('97 70' UIDTX$_3$ BCC ~CRC_A)	→			IDLE
		←	Mute		
Команда по ИСО/МЭК 14443-4	I(0)$_0$(TEST_COMMAND1(1))	→			IDLE
		←	Mute		
DESELECT	S(DESELECT)	→			IDLE
		←	Mute		
RATS	RATS(0,0)	→			IDLE
		←	Mute		
PPS	PPS(0,0,0)	→			IDLE
		←	Mute		

[a] Пусть $1 \le n_1 \le 32$, UIDTX$_3$[[n_1]] = 0. Если такого числа нет, то данное испытание может быть пропущено.
[b] Пусть $1 \le n_2 \le 32$, UIDTX$_3$[[n_2]] = 1. Если такого числа нет, то данное испытание может быть пропущено.
[c] FDT принимает значение 1172/*fc* (~86,43 мкс), если последний бит = (0)b, и 1236/*fc* (~91,15 мкс), если последний бит = (1)b [см. предельные значения в ИСО/МЭК 14443-3:2011 (подпункт 6.2.1.1)].
[d] Пусть $1 \le n_3 \le 32$.
[e] Любая команда AC или SELECT может предшествовать команде антиколлизии для извлечения UID PICC, особенно для произвольного UID.

G.3.3.6.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.15.

Таблица G.15 — Критерии оценки результата испытания поведения PICC типа A в состоянии READY(3)

Объяснение	Результат испытаний
Если PICC имеет одиночный или двойной размер UID	Не имеет отношения (N/A)
Если PICC имеет тройной размер UID и только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.3.3.7 Сценарий G.6: Поведение PICC типа A в состоянии ACTIVE

G.3.3.7.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа A в состоянии ACTIVE требованиям ИСО/МЭК 14443-3:2011 (пункт 6.3.4).

G.3.3.7.2 Процедура

Следует выполнить следующие шаги для каждой строки таблицы G.16.

a) Привести PICC в состояние ACTIVE.

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Если ответом PICC не является Mute, то проверить, соответствует ли время задержки кадра PICC значению, указанному в графе «FDT».

e) Проверить, находится ли PICC в состоянии TTS.

Таблица G.16 — Переходы из состояния ACTIVE

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
REQA	REQA	→			IDLE
		←	Mute		
WUPA	WUPA	→			IDLE
		←	Mute		
AC	('93' NVB $UIDTX_1[[1..n_1]]$)[a]	→			IDLE
		←	Mute[b]		
nAC	('93' NVB ~$UIDTX_1[[1..n_1]]$)[a]	→			IDLE
		←	Mute[b]		
HLTA	HLTA	→			HALT
		←	Mute		
SELECT	SELECT(1)	→			IDLE
		←	Mute[b]		
nSELECT	('93 70' ~$UIDTX_1$ BCC CRC_A)	→			IDLE
		←	Mute[b]		
RATS	RATS(0,0)	→			PROTOCOL
		←	ATS	< 65536/*fc*	
Состояние ошибки	('E0 00' ~CRC_A)	→			IDLE
		←	Mute[b]		
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→			IDLE
		←	Mute[b]		
DESELECT	S(DESELECT)	→			IDLE
		←	Mute[b]		
PPS	PPS(0,0,0)	→			IDLE
		←	Mute[b]		

[a] Пусть $1 \le n_1 \le 32$.
[b] Mute или проприетарный ответ.

G.3.3.7.3 Протокол испытаний
Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.17.

Таблица G.17 — Критерии оценки результата испытания поведения PICC типа A в состоянии ACTIVE

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.3.3.8 Сценарий G.7: Поведение PICC типа A в состоянии HALT

G.3.3.8.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа A в состоянии HALT требованиям ИСО/МЭК 14443-3:2011 (пункт 6.3.5).

G.3.3.8.2 Процедура

Следует выполнить следующие шаги для каждой строки таблицы G.18.

a) Привести PICC в состояние HALT.

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Если ответом PICC не является Mute, то проверить соответствует ли время задержки кадра PICC значению, указанному в графе «FDT».

e) Проверить, находится ли PICC в состоянии TTS.

Таблица G.18 — Переходы из состояния HALT

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
REQA	REQA	→			HALT
		←	Mute		
WUPA	WUPA	→			READY*(1)
		←	ATQA	1236/*fc*	
HLTA	HLTA	→			HALT
		←	Mute		
AC	('93' NVB $UIDTX_1$[[1..n_1]])[a)]	→			HALT
		←	Mute		
nAC	('93' NVB ~$UIDTX_1$[[1..n_1]])[a)]	→			HALT
		←	Mute		
SELECT	SELECT(1)	→			HALT
		←	Mute		
nSELECT	('93 70' ~$UIDTX_1$ BCC CRC_A)	→			HALT
		←	Mute		
RATS	RATS(0,0)	→			HALT
		←	Mute		
Состояние ошибки	('52') в стандартном кадре	→			HALT
		←	Mute		
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→			HALT
		←	Mute		
DESELECT	S(DESELECT)	→			HALT
		←	Mute		
PPS	PPS(0,0,0)	→			HALT
		←	Mute		

[a)] Пусть $1 \le n1 \le 32$.

G.3.3.8.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.19.

Таблица G.19 — Критерии оценки результата испытания поведения PICC типа A в состоянии HALT

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.3.3.9 Сценарий G.8: Поведение PICC типа A в состоянии READY*(1)

G.3.3.9.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа A в состоянии READY* на каскадном уровне 1 требованиям ИСО/МЭК 14443-3:2011 (пункт 6.3.6).

G.3.3.9.2 Процедура

Выполнить следующие шаги для каждой строки таблицы G.20.

a) Привести PICC в состояние READY*(1).

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Если ответом PICC не является Mute, то проверить, соответствует ли время задержки кадра PICC значению, указанному в графе «FDT».

e) Проверить находится ли PICC в состоянии TTS.

Таблица G.20 — Переходы из состояния READY*(1)

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
REQA	REQA	→			HALT
		←	Mute		
WUPA	WUPA	→			HALT
		←	Mute		
HLTA	HLTA	→			HALT
		←	Mute		
AC (разделение после (0)b)	('93' NVB $UIDTX_1$[[1..n_1]])[a]	→			READY*(1)
		←	Если n_1 = 32, то (BCC), иначе ($UIDTX_1$[[n_1+1..32]] BCC)[a]	1172/*fc*	
AC (разделение после (1)b)	('93' NVB $UIDTX_1$[[1..n_2]])[b]	→			READY*(1)
		←	Если n_2 = 32, то (BCC), иначе ($UIDTX_1$[[n_2+1..32]] BCC)[b]	1236/*fc*	
nAC (неправильный UID)	('93' NVB ~$UIDTX_1$[[1..n_3]])[f]	→			HALT
		←	Mute		
SELECT	SELECT(1)	→			TTS[e]
		←	SAK[d]	FDT[c]	
nSELECT (неправильный UID)	('93 70' ~$UIDTX_1$ BCC CRC_A)	→			HALT
		←	Mute		

Окончание таблицы G.20

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
Состояние ошибки	('93 70' UIDTX$_1$ BCC ~CRC_A)	→			HALT
		←	Mute		
Команда по ИСО/МЭК 14443-4	I(0)$_0$(TEST_COMMAND1(1))	→			HALT
		←	Mute		
DESELECT	S(DESELECT)	→			HALT
		←	Mute		
RATS	RATS(0,0)	→			HALT
		←	Mute		
PPS	PPS(0,0,0)	→			HALT
		←	Mute		

[a] Пусть $1 \le n_1 \le 32$, UIDTX$_1$[[n_1]] = 0. Если такого числа нет, то данное испытание может быть пропущено.
[b] Пусть $1 \le n_2 \le 32$, UIDTX$_1$[[n_2]] = 1. Если такого числа нет, то данное испытание может быть пропущено.
[c] FDT принимает значение 1172/*fc* (~86,43 мкс), если последний бит = (0)b, и 1236/*fc* (~91,15 мкс), если последний бит = (1)b [см. предельные значения в ИСО/МЭК 14443-3:2011 (подпункт 6.2.1.1)].
[d] Каскадный бит SAK должен быть (0)b для PICC с одиночным размером UID и (1)b для PICC с двойным и тройным размером UID.
[e] PICC с одиночным размером UID должна быть в состоянии ACTIVE; PICC с двойным и тройным размером UID должны быть в состоянии READY.
[f] Пусть $1 \le n_3 \le 32$.

G.3.3.9.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.21.

Таблица G.21 — Критерии оценки результата испытания поведения PICC типа A в состоянии READY*(1)

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.3.3.10 Сценарий G.9: Поведение PICC типа A в состоянии READY*(2)

G.3.3.10.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа A в состоянии READY*на каскадной уровне 2 требованиям ИСО/МЭК 14443-3:2011 (пункт 6.3.6). Данное испытание предназначено только для PICC с двойным или тройным размером UID.

G.3.3.10.2 Процедура

Следует выполнить следующие шаги для каждой строки таблицы G.22.

a) Привести PICC в состояние READY*(2).

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Если ответом PICC не является Mute, то проверить, соответствует ли время задержки кадра PICC значению, указанному в графе «FDT».

e) Проверить, находится ли PICC в состоянии TTS.

Таблица G.22 — Переходы из состояния READY*(2)

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
REQA	REQA	→			HALT
		←	Mute		
WUPA	WUPA	→			HALT
		←	Mute		
HLTA	HLTA	→			HALT
		←	Mute		
AC (разделение после (0)b)	('95' NVB $UIDTX_2$[[1..n_1]])[a]	→			READY*(2)
		←	Если n_1 = 32, то (BCC), иначе ($UIDTX_2$[[n_1+1..32]] BCC)[a]	1172/ *fc*	
AC (разделение после (1)b)	('95' NVB $UIDTX_2$[[1..n_2]])[b]	→			READY*(2)
		←	Если n_2 = 32, то (BCC), иначе ($UIDTX_2$[[n_2+1..32]] BCC)[b]	1236/ *fc*	
nAC (неправильный UID)	('95' NVB ~$UIDTX_2$[[1..n_3]])[f]	→			HALT
		←	Mute		
SELECT	SELECT(2)	→			TTS[e]
		←	SAK[d]	FDT[c]	
nSELECT (неправильный UID)	('95 70' ~$UIDTX_2$ BCC CRC_A)	→			HALT
		←	Mute		
Состояние ошибки	('95 70' $UIDTX_2$ BCC ~CRC_A)	→			HALT
		←	Mute		
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→			HALT
		←	Mute		
DESELECT	S(DESELECT)	→			HALT
		←	Mute		
RATS	RATS(0,0)	→			HALT
		←	Mute		
PPS	PPS(0,0,0)	→			HALT
		←	Mute		

[a] Пусть $1 \le n_1 \le 32$, $UIDTX_2$[[n_1]] = 0. Если такого числа нет, то данное испытание может быть пропущено.

[b] Пусть $1 \le n_2 \le 32$, $UIDTX_2$[[n_2]] = 1. Если такого числа нет, то данное испытание может быть пропущено.

[c] FDT принимает значение 1172/*fc* (~86,43 мкс), если последний бит = (0)b, и 1236/*fc* (~91,15 мкс), если последний бит = (1)b [см. предельные значения в ИСО/МЭК 14443-3:2011 (подпункт 6.2.1.1)].

[d] Каскадный бит SAK должен быть (0)b для PICC с двойным размером UID и (1)b для PICC с тройным размером UID.

[e] PICC с двойным размером UID должна быть в состоянии ACTIVE; PICC с тройным размером UID должна быть в состоянии READY.

[f] Пусть $1 \le n_3 \le 32$.

G.3.3.10.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.23.

Таблица G.23 — Критерии оценки результата испытания поведения PICC типа A в состоянии READY*(2)

Объяснение	Результат испытаний
Если PICC имеет одиночный размер UID	Не имеет отношения (N/A)
Если PICC имеет двойной или тройной размер UID и только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.3.3.11 Сценарий G.10: Поведение PICC типа A в состоянии READY*(3)

G.3.3.11.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа A в состоянии READY* на каскадном уровне 3 требованиям ИСО/МЭК 14443-3:2011 (пункт 6.3.6). Данное испытание предназначено только для PICC с тройным размером UID.

G.3.3.11.2 Процедура

Следует выполнить следующие шаги для каждой строки таблицы G.24.

a) Привести PICC в состояние READY*(3).

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Если ответом PICC не является Mute, то проверить, соответствует ли время задержки кадра PICC значению, указанному в графе «FDT».

e) Проверить, находится ли PICC в состоянии TTS.

Таблица G.24 — Переходы из состояния READY*(3)

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
REQA	REQA	→			HALT
		←	Mute		
WUPA	WUPA	→			HALT
		←	Mute		
HLTA	HLTA	→			HALT
		←	Mute		
AC (разделение после (0)b)	('97' NVB $UIDTX_3$[[1..n_1]])[a]	→			READY*(3)
		←	Если n_1 = 32, то (BCC), иначе ($UIDTX_3$[[n_1+1..32]] BCC)[a]	1172/*fc*	
AC (разделение после(1)b)	('97' NVB $UIDTX_3$[[1..n_2]])[b]	→			READY*(3)
		←	Если n_2 = 32, то (BCC), иначе ($UIDTX_3$[[n_2+1..32]] BCC)[b]	1236/*fc*	
nAC (неправильный UID)	('97' NVB ~$UIDTX_3$[[1..n_3]])[d]	→			HALT
		←	Mute		
SELECT	SELECT(3)	→			ACTIVE*
		←	SAK(complete)	FDT[c]	
nSELECT (неправильный UID)	('97 70' ~$UIDTX_3$ BCC CRC_A)	→			HALT
		←	Mute		

Окончание таблицы G.24

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
Состояние ошибки	('97 70' $UIDTX_3$ BCC ~CRC_A)	→			HALT
		←	Mute		
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→			HALT
		←	Mute		
DESELECT	S(DESELECT)	→			HALT
		←	Mute		
RATS	RATS(0,0)	→			HALT
		←	Mute		
PPS	PPS(0,0,0)	→			HALT
		←	Mute		

[a] Пусть $1 \le n_1 \le 32$, $UIDTX_3[[n_1]] = 0$. Если такого числа нет, то данное испытание может быть пропущено.
[b] Пусть $1 \le n_2 \le 32$, $UIDTX_3[[n_2]] = 1$. Если такого числа нет, то данное испытание может быть пропущено.
[c] FDT принимает значение 1172/*fc* (~86,43 мкс), если последний бит = (0)b, и 1236/*fc* (~91,15 мкс), если последний бит = (1)b [см. предельные значения в ИСО/МЭК 14443-3:2011 (подпункт 6.2.1.1)].
[d] Пусть $1 \le n_3 \le 32$.

G.3.3.11.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.25.

Таблица G.25 — Критерии оценки результата испытания поведения PICC типа A в состоянии READY*(3)

Объяснение	Результат испытаний
Если PICC имеет одиночный или двойной размер UID	Не имеет отношения (N/A)
Если PICC имеет тройной размер UID и только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.3.3.12 Сценарий G.11: Поведение PICC типа A в состоянии ACTIVE*

G.3.3.12.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа A в состоянии ACTIVE* требованиям ИСО/МЭК 14443-3:2011 (пункт 6.3.7).

G.3.3.12.2 Процедура

Следует выполнить следующие шаги для каждой строки таблицы G.26.

a) Привести PICC в состояние ACTIVE.

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Если ответом PICC не является Mute, то проверить, соответствует ли время задержки кадра PICC значению, указанному в графе «FDT».

e) Проверить, находится ли PICC в состоянии TTS.

Таблица G.26 — Переходы из состояния ACTIVE*

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
REQA	REQA	→			HALT
		←	Mute		
WUPA	WUPA	→			HALT
		←	Mute		
HLTA	HLTA	→			HALT
		←	Mute		
AC	('93' NVB $UIDTX_1$[[1..n_1]])[a]	→			HALT
		←	Mute[b]		
nAC	('93' NVB ~$UIDTX_1$[[1..n_1]])[a]	→			HALT
		←	Mute[b]		
SELECT	SELECT(1)	→			HALT
		←	Mute[b]		
nSELECT	('93 70' ~$UIDTX_1$ BCC CRC_A)	→			HALT
		←	Mute[b]		
RATS	RATS(0,0)	→			PROTOCOL
		←	ATS	< 65536/*fc*	
Состояние ошибки	('E0 00' ~CRC_A)	→			HALT
		←	Mute[b]		
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→			HALT
		←	Mute[b]		
DESELECT	S(DESELECT)	→			HALT
		←	Mute[b]		
PPS	PPS(0,0,0)	→			HALT
		←	Mute[b]		

[a] Пусть $1 \le n_1 \le 32$.
[b] Mute или проприетарный ответ.

G.3.3.12.3 Протокол испытаний
Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.27.

Таблица G.27 — Критерии оценки результата испытания поведения PICC типа A в состоянии ACTIVE*

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.3.3.13 Сценарий G.12: Поведение PICC типа A в состоянии PROTOCOL

G.3.3.13.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа A в состоянии PROTOCOL требованиям ИСО/МЭК 14443-4:2008. Данное испытание должно подтвердить, что активированная PICC не отвечает на какую-либо команду антиколлизии или инициализации.

G.3.3.13.2 Процедура

Следует выполнить следующие шаги для каждой строки таблицы G.28.

a) Привести PICC в состояние PROTOCOL.

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Если ответом PICC не является Mute, то проверить, соответствует ли время задержки кадра PICC значению, указанному в графе «FDT».

e) Проверить, находится ли PICC в состоянии TTS.

Таблица G.28 — Переходы из состояния PROTOCOL

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
REQA	REQA	→			PROTOCOL
		←	Mute		
WUPA	WUPA	→			PROTOCOL
		←	Mute		
AC	('93' NVB $UIDTX_1$[[1..n_1]])[a]	→			PROTOCOL
		←	Mute		
nAC	('93' NVB ~$UIDTX_1$[[1..n_1]])[a]	→			PROTOCOL
		←	Mute		
HLTA	HLTA	→			PROTOCOL
		←	Mute		
SELECT	SELECT(1)	→			PROTOCOL
		←	Mute		
nSELECT	('93 70' ~$UIDTX_1$ BCC CRC_A)	→			PROTOCOL
		←	Mute		
RATS	RATS(0,0)	→			PROTOCOL
		←	Mute		
Состояние ошибки	S(DESELECT, ~CRC_A)	→			PROTOCOL
		←	Mute		
DESELECT	S(DESELECT)	→			HALT
		←	S(DESELECT)	65536/*fc*, как определено в ИСО/МЭК 14443-4: 2008 (подраздел 8.1)	
PPS	PPS(0,0,0)	→			PROTOCOL
		←	Mute или PPS-ответ[b]		
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→			PROTOCOL
		←	$I(0)_0$(TEST_ RESPONSE1(1))	< FWT	

[a] Пусть $1 \le n_1 \le 32$.

[b] PPS-ответ возвращается, если PICC поддерживает PPS.

G.3.3.13.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.29.

Таблица G.29 — Критерии оценки результата испытания поведения PICC типа A в состоянии PROTOCOL

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.3.4 Сценарий G.13: Обработка антиколлизии типа A

G.3.4.1 Область применения

Данное испытание служит для выполнения полного побитового цикла антиколлизии в соответствии с ИСО/МЭК 14443-3:2011 (пункт 6.5.3).

G.3.4.2 Процедура

a) Поместить PICC в область поля.

b) Привести PICC в состояние READY(1).

c) Выполнить AnticollisionA.

d) Привести PICC в состояние READY*(1).

e) Выполнить AnticollisionA.

Псевдокод: Процедура антиколлизии типа A

```
Procedure AnticollisionA
// TPDUSend and TPDURecv are PCD specific functions
// to send and receive frames

for c = 1 to CascadeLevels do

       // anticollision over UID bits
       for p = 1 to 32 do
            // enter desired cascade level
           if c ≥ 2 then TPDUSend(SELECT(1))
           if c = 3 then TPDUSend(SELECT(2))
          // anticollision with matched bit
           NVB[[1..4]] = (p + 16) mod 8
           NVB[[5..8]] = (p + 16) div 8
           TPDUSend (SEL(c) NVB UIDTXc[[1..p]])
           if TPDURecv() ≠ (UIDTXc[[p+1..32]] BCC) then return FAIL
          // anticollision with unmatched bit
           TPDUSend(SEL(c) NVB UIDTXc[[1..p-1]] ~UIDTXc[[p]])
           if TPDURecv() ≠ Mute then return FAIL
          // re-enter READY(1) (resp. READY*(1)) state
           TPDUSend (WUPA)
      end for
   end for
return PASS
```

G.3.4.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.30.

Таблица G.30 — Критерий оценки результата испытания обработки антиколлизии типа A

Объяснение	Результат испытаний
Только если каждая процедура испытания на антиколлизию выдала PASS	PASS
Если какая-либо процедура испытания на антиколлизию выдала значение FAIL	FAIL

G.3.5 Обработка RATS

Обработку RATS подвергают проверке при испытаниях по G.3.3.7 и G.3.3.12.

Сценарий G.14: RATS после ложного RATS

Сценарий G.14, который был определен в ИСО/МЭК 10373-6:2001/Изм. А1:2007, исключен.

Сценарий G.15: RATS после RATS

Сценарий G.15, который был определен в ИСО/МЭК 10373-6:2001/Изм. А1:2007, исключен.

G.3.6 Обработка PPS-запроса

G.3.6.1 Область применения

Цель данного испытания — определить, соответствует ли выполнение PICC типа А процедуры обработки PPS-запроса требованиям ИСО/МЭК 14443-4:2008 (подпункт 5.6.2.2).

G.3.6.2 Процедура

Следует выполнить следующие шаги для каждого сценария G.16 — G.19, перечисленного в настоящем пункте.

a) Привести PICC в состояние PROTOCOL.

b) Подать команду, указанную в графе «Оборудование для испытаний PICC» в каждом сценарии.

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Проверить, находится ли PICC в состоянии PROTOCOL.

Сценарий G.16: PPS без изменения параметров

Сценарий G.16, который был определен в ИСО/МЭК 10373-6:2001/Изм. А1:2007, исключен.

Сценарий G.17: PPS без PPS1

Оборудование для испытаний PICC		PICC
('D0 01' CRC_A)	→	
	←	Mute или ('D0' CRC_A)[a]
[a] Оба ответа действительны.		

Сценарий G.18: PPS после PPS

Оборудование для испытаний PICC		PICC
PPS(0,0,0)	→	
	←	Mute или ('D0' CRC_A)[a]
PPS(0,0,0)	→	
	←	Mute
[a] Ответ зависит от того, поддерживает ли PICC PPS или нет. Если PICC не поддерживает какие-либо изменяемые параметры, то она может не поддерживать PPS-запрос, т. к. PCD не должно посылать PPS такой PICC [см. ИСО/МЭК 14443-4:2008 (раздел 5, шестое перечисление)]		

Сценарий G.19: PPS после неполученного PPS

Оборудование для испытаний PICC		PICC
('D0 11 00' ~CRC_A)	→	
	←	Mute
PPS(0,0,0)	→	
	←	Mute

G.3.6.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.31.

Таблица G.31 — Критерии оценки результата испытания обработки PPS-запроса

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.3.7 Сценарий G.20: Обработка FSD

G.3.7.1 Область применения

Цель данного испытания — определить, принимает ли PICC типа A значение FSD, предложенное RATS, в соответствии с требованиями ИСО/МЭК 14443-4:2008 (подраздел 5.1).

G.3.7.2 Процедура

Следует выполнить следующие шаги для каждого FSDI, указанного в ИСО/МЭК 14443-4:

(Измененная редакция. Изм. А4:2012.)

a) Привести PICC в состояние ACTIVE.

b) Подать команду RATS(0, fsdi) с параметром fsdi как в отдельном испытании.

c) Проверить, что ответом PICC является действительный ATS и что его размер ≤ FSD.

Примечание — PICC может потребовать дополнительные последовательности, чтобы быть готовой принять TEST_COMMAND2(2).

d) Послать I-блок $I(0)_0$(TEST_COMMAND2(2)).

e) Проверить, что размер ответного I-блока, посланного PICC, ≤ FSD.

G.3.7.3 Протокол испытаний

Используя форму таблицы G.61, заполнить в ней соответствующую строку согласно таблице G.32.

Таблица G.32 — Критерии оценки результата испытания сценария G.20: Обработка FSD

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.4 Метод испытаний для инициализации PICC типа B

G.4.1 Введение

По результатам испытаний, описанных в данном разделе, определяют, соответствует ли PICC типа B требованиям ИСО/МЭК 14443-3:2011. Если соответствие требованиям ИСО/МЭК 14443-4:2008 не нужно, то испытания, содержащие команды ИСО/МЭК 14443-4, не применяют.

Если PICC не поддерживает ни REQB/WUPB с N > 1, ни команду Slot-MARKER [см. ИСО/МЭК 14443-3:2011 (пункт 7.6.1)], то все испытания, содержащие эти команды, не применяют.

G.4.2 Сценарий 21: Опрос

G.4.2.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа B при приеме REQB требованиям ИСО/МЭК 14443-3:2011 (раздел 5).

G.4.2.2 Процедура

Выполнить следующие этапы для трех различных значений напряженности рабочего поля: 1,5, 4,5 и 7,5 А/м (среднеквадратичное значение).

a) Поместить PICC в поле и осуществить ее настройку.

b) Переключить радиочастотное рабочее поле в состояние «выключено» на минимальное время для возврата PICC в исходное состояние в соответствии с ИСО/МЭК 14443-3:2011 (подраздел 5.4).

c) Переключить радиочастотное рабочее поле в состояние «включено».

d) Выдержать 5 мс и послать действительный кадр команды REQB(1).

e) Зафиксировать наличие и содержание ответа PICC.

f) Переключить радиочастотное рабочее поле в состояние «выключено» на минимальное время для возврата PICC в исходное состояние в соответствии с ИСО/МЭК 14443-3:2011 (подраздел 5.4).

g) Переключить радиочастотное рабочее поле в состояние «включено».

h) Выдержать 5 мс и послать действительный кадр команды REQA (используя модуляцию типа A).

i) Выдержать 5 мс и послать действительный кадр команды REQB(1).

j) Зафиксировать наличие и содержание ответа PICC.

G.4.2.3 Протокол испытаний

Используя форму таблицы G.62, заполнить в ней соответствующую строку согласно таблице G.33.

Таблица G.33 — Критерии оценки результата испытания сценария G.21: Опрос

Объяснение	Результат испытаний
Только если ответ PICC представляет собой действительный ATQB на обоих этапах e) и j)	PASS
Если ответ PICC не является действительным ATQB на любом из этапов e) или j)	FAIL

G.4.3 Сценарий G.22: Прием PICC

G.4.3.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа B при приеме сообщений от PCD требованиям ИСО/МЭК 14443-3:2011 (пункты 7.1.1, 7.1.2, 7.1.4 и 7.1.5).

G.4.3.2 Процедура

Выполнить следующие шаги для каждой строки таблицы G.34.

a) Поместить эталонную PICC в область поля.

b) Установить параметры кадра оборудования для испытаний PICC в соответствии с таблицей G.34.

c) Послать команду REQB.

d) Зафиксировать наличие, содержание и временные соотношения ответа PICC.

e) Проверить, что формат кадра ответа PICC соответствует следующему:

- ответом PICC должен быть действительный ATQB;
- временные значения логического нуля SOF должны быть в промежутке от 10 до 11 etu;
- временные значения логической единицы SOF должны быть в промежутке от 2 до 3 etu;
- временные значения логического нуля EOF должны быть в промежутке от 10 до 11 etu;
- временные значения TR0 должны быть в диапазоне $64/fs \le TR0 \le 256/fs$;
- временные значения TR1 должны быть в диапазоне $80/fs \le TR0 \le 200/fs$;
- PICC должна отключить поднесущую в промежутке между 0 и 2 etu после EOF.

Таблица G.34 — Параметры кадра типа B

EGT, мкс	SOF (логический 0), etu	SOF (логическая 1), etu	EOF, etu
0	10	2	10
57	10	2	10
0	11	2	10
0	10	3	10
0	10	2	11

G.4.3.3 Протокол испытаний

Используя форму таблицы G.62, заполнить в ней соответствующую строку согласно таблице G.35.

Таблица G.35 — Критерии оценки результата испытания сценария G.22: Прием PICC

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.4.4 Испытания переходов состояний PICC типа B

Цель данных испытаний — проверить правильность реализации диаграммы состояния PICC типа B, как описано в ИСО/МЭК 14443-3:2011 (подраздел 7.4).

G.4.4.1 Общий принцип испытаний

Процедура аналогична описанной для PICC типа A (см. G.3.3.2).

G.4.4.1.1 Функции для приведения PICC в исходное состояние испытания (TIS)

Приведение PICC в состояние TIS осуществляют с помощью последовательности команд перехода, определенных в таблице G.37. Общий метод заключается в следующем.

Для того чтобы привести PICC в состояние TIS, находят соответствующую последовательность переходов состояний в таблице G.36. Далее последовательно применяют переходы состояний, описанные в графе «Последовательность переходов состояний», находя соответствующие команды в таблице переходов состояний G.37. Следует всегда проверять содержание и целостность ответа PICC.

Таблица G.36 — Последовательность переходов состояний

TIS	Последовательность переходов состояний
POWER-OFF	—
IDLE	POWER-OFF → IDLE
READY-REQUESTED	POWER-OFF → IDLE → READY-REQUESTED
READY-DECLARED	POWER-OFF → IDLE → READY-DECLARED
PROTOCOL	POWER-OFF → IDLE → READY-DECLARED → PROTOCOL
HALT	POWER-OFF → IDLE → READY-DECLARED → HALT

Таблица G.37 — Переходы состояний

Состояние → Следующее состояние	Оборудование для испытаний PICC		PICC
POWER-OFF→ IDLE	Power On (радиочастотное поле включено)	→	
		←	Mute
IDLE → READY-REQUESTED	REQB(16)	→	
		←	Mute[a)]
IDLE → READY-DECLARED	REQB(1)	→	
		←	ATQB
READY-DECLARED → HALT	HLTB	→	
		←	'00' CRC_B
READY-DECLARED → PROTOCOL	ATTRIB(0,0)	→	
		←	ATA(0)
[a)] В случае если PICC выбрала слот 1, то команда REQB должна повторяться до тех пор, пока PICC не ответит командой ATQB.			

G.4.4.1.2 Функции для проверки действительности целевого состояния испытания (TTS)

В таблице G.38 описаны переходы состояний, используемые, чтобы проверить, находится ли PICC в состоянии S. Содержание ответа PICC (т. е. ATQB…) должно быть детально проверено на соответствие ИСО/МЭК 14443-3 и ИСО/МЭК 14443-4.

Примечание — Испытание может вызвать смену состояния PICC.

Таблица G.38 — Проверка TTS

TTS	Оборудование для испытаний PICC		PICC
IDLE	REQB(1)	→	
		←	ATQB
READY-REQUESTED	SLOTMARKER (n)[a)]	→	
		←	ATQB
READY-DECLARED	ATTRIB(0,0)	→	
		←	ATA(0)
PROTOCOL	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
HALT	REQB(1)	→	
		←	Mute
	WUPB(1)	→	
		←	ATQB
[a)] Поскольку выбранный слот PICC неизвестен, то необходимо повторять команду Slot-MARKER с различными значениями слотов, пока не будет получен ATQB.			

Примечание — Номер блоков может быть 0 или 1 в зависимости от правил нумерации блоков, определенных в ИСО/МЭК 14443-4:2008 (пункт 7.5.4)[1].

(Введено дополнительно. Изм. А3:2012.)

G.4.4.2 Сценарий G.23: Поведение PICC типа B в состоянии IDLE

G.4.4.2.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа B в состоянии IDLE требованиям ИСО/МЭК 14443-3:2011 (пункт 7.4.4).

G.4.4.2.2 Процедура

Выполнить следующие шаги для каждой строки таблицы G.39.

[1] В ИСО/МЭК 10373-6:2011/Изм. А3:2012 указан пункт 7.5.3 без учета изменения нумерации пунктов, введенного изменением А1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

a) Привести PICC в состояние IDLE.

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Проверить, находится ли PICC в состоянии TTS.

Таблица G.39 — Переходы из состояния IDLE

Переход	Оборудование для испытаний PICC		PICC	TTS
REQB[e)]	REQB(1)	→		READY-DECLARED
		←	ATQB	
WUPB[e)]	WUPB(1)	→		READY-DECLARED
		←	ATQB	
REQB (неправильный CRC)	('05 00 00' ~CRC_B)	→		IDLE
		←	Mute	
WUPB (неправильный CRC)	('05 00 08' ~CRC_B)	→		IDLE
		←	Mute	
HLTB[b)]	HLTB	→		IDLE
		←	Mute	
ATTRIB[b)]	ATTRIB(0,0)	→		IDLE
		←	Mute	
Slot-MARKER	SLOTMARKER(n)[a)]	→		IDLE
		←	Mute	
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→		IDLE
		←	Mute	
DESELECT	S(DESELECT)	→		IDLE
		←	Mute	
REQB[f)] (несогласованный AFI)	REQB(1,nAFI)	→		IDLE
		←	Mute	
WUPB[f)] (несогласованный AFI)	WUPB(1,nAFI)	→		IDLE
		←	Mute	
HLTB (несогласованный PUPI)	HLTB(~PUPI)	→		IDLE
		←	Mute	
ATTRIB (несогласованный PUPI)	ATTRIB(0, 0, ~PUPI)	→		IDLE
		←	Mute	
REQB[d)]	REQB(16)[c)]	→		READY-REQUESTED
		←	Mute	
WUPB[d)]	WUPB(16)[c)]	→		READY-REQUESTED
		←	Mute	

[a)] n должно принимать все значения $2 \le n \le 16$

[b)] Для PICC, использующих произвольный PUPI, применяют произвольно выбранное значение для этой команды.

[c)] При этом с вероятностью 1/16 PICC даст ответ ATQB и перейдет в состояние READY-DECLARED.

[d)] Если PICC не поддерживает REQB/WUPB с N > 1 [см. ИСО/МЭК 14443-3:2011 (пункт 7.6.1)], то данное испытание не применяют.

[e)] Все согласованные AFI, определенные изготовителем PICC, должны быть проверены.

[f)] Несогласованные AFI, используемые в данном испытании, не должны быть RFU-значениями. Если PICC принимает все AFI, то данное испытание не подходит.

(Введены дополнительно. Изм. А3:2012.)

G.4.4.2.3 Протокол испытаний

Используя форму таблицы G.62, заполнить в ней соответствующую строку согласно таблице G.40.

Таблица G.40 — Критерии оценки результата испытания сценария 23: Поведение PICC типа B в состоянии IDLE

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.4.4.3 Сценарий G.24: Поведение PICC типа B в подсостоянии READY-REQUESTED

Если PICC не поддерживает ни REQB/WUPB с N > 1, ни команду Slot-MARKER [см. ИСО/МЭК 14443-3:2011 (пункт 7.6.1)], то данный сценарий не следует применять.

G.4.4.3.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа B в подсостоянии READY-REQUESTED требованиям ИСО/МЭК 14443-3:2011 (пункт 7.4.5).

G.4.4.3.2 Процедура

Выполнить следующие шаги для каждой строки таблицы G.41.

a) Привести PICC в подсостояние READY-REQUESTED.

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Проверить, находится ли PICC в состоянии TTS.

Таблица G.41 — Переходы из подсостояния READY-REQUESTED

Переход	Оборудование для испытаний PICC		PICC	TTS
REQB[e)]	REQB(1)	→		READY-DECLARED
		←	ATQB	
WUPB[e)]	WUPB(1)	→		READY-DECLARED
		←	ATQB	
REQB (неправильный CRC)	('05 00 00' ~CRC_B)	→		READY-REQUESTED
		←	Mute	
WUPB (неправильный CRC)	('05 00 08' ~CRC_B)	→		READY-REQUESTED
		←	Mute	
HLTB[b)]	HLTB	→		READY-REQUESTED
		←	Mute	
ATTRIB[b)]	ATTRIB(0,0)	→		READY-REQUESTED
		←	Mute	
Slot-MARKER	SLOTMARKER(n)[a)]	→		READY-DECLARED
		←	ATQB или Mute	
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→		READY-REQUESTED
		←	Mute	
DESELECT	S(DESELECT)	→		READY-REQUESTED
		←	Mute	

Окончание таблицы G.41

Переход	Оборудование для испытаний PICC		PICC	TTS
REQB	REQB(16)[c]	→		READY-REQUESTED
		←	Mute	
WUPB	WUPB(16)[c]	→		READY-REQUESTED
		←	Mute	
REQB[f] (несогласованный AFI)	REQB(1,nAFI)	→		IDLE[d]
		←	Mute	
WUPB[f] (несогласованный AFI)	WUPB(1,nAFI)	→		IDLE[d]
		←	Mute	
HLTB (несогласованный PUPI)	HLTB(~PUPI)	→		READY-REQUESTED
		←	Mute	
ATTRIB (несогласованный PUPI)	ATTRIB(0, 0, ~PUPI)	→		READY-REQUESTED
		←	Mute	

[a] n должно принимать все значения $2 \le n \le 16$. PICC должна дать ответ ATQB на ровно одно значение n, в противном случае — Mute.

[b] Для PICC, использующих произвольный PUPI, применяют произвольно выбранное значение для этой команды.

[c] При этом с вероятностью 1/16 PICC даст ответ ATQB и перейдет в состояние READY-DECLARED.

[d] Подать все команды Slot-MARKER и проверить, что ответа нет до проверки состояния IDLE.

[e] Все согласованные AFI, определенные изготовителем PICC, должны быть проверены.

[f] Несогласованные AFI, используемые в данном испытании, не должны быть RFU-значениями. Если PICC принимает все AFI, то данное испытание не подходит.

(Введены дополнительно. Изм. А3:2012.)

G.4.4.3.3 Протокол испытаний

Используя форму таблицы G.62, заполнить в ней соответствующую строку согласно таблице G.42.

Таблица G.42 — Критерии оценки результата испытания сценария G.24: Поведение PICC типа B в подсостоянии READY-REQUESTED

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.4.4.4 Сценарий G.25: Поведение PICC типа B в подсостоянии READY-DECLARED

G.4.4.4.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа B в подсостоянии READY-DECLARED требованиям ИСО/МЭК 14443-3:2011 (пункт 7.4.6).

G.4.4.4.2 Процедура

Выполнить следующие шаги для каждой строки таблицы G.43.

a) Привести PICC в состояние READY-DECLARED.

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Проверить, находится ли PICC в состоянии TTS.

Таблица G.43 — Переходы из подсостояния READY-DECLARED

Переход	Оборудование для испытаний PICC		PICC	TTS
REQB[e)]	REQB(1)	→		READY-DECLARED
		←	ATQB	
WUPB[e)]	WUPB(1)	→		READY-DECLARED
		←	ATQB	
REQB (неправильный CRC)	('05 00 00' ~CRC_B)	→		READY-DECLARED
		←	Mute	
WUPB (неправильный CRC)	('05 00 08' ~CRC_B)	→		READY-DECLARED
		←	Mute	
HLTB	HLTB	→		HALT
		←	('00' CRC_B)	
ATTRIB	ATTRIB(0,0)	→		PROTOCOL
		←	ATA(0)	
Slot-MARKER	SLOTMARKER(n)[a)]	→		READY-DECLARED
		←	Mute	
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→		READY-DECLARED
		←	Mute	
DESELECT	S(DESELECT)	→		READY-DECLARED
		←	Mute	
REQB[b)]	REQB(16)[c)]	→		READY-REQUESTED
		←	Mute	
WUPB[b)]	WUPB(16)[c)]	→		READY-REQUESTED
		←	Mute	
HLTB (несогласованный PUPI)	HLTB(~PUPI)	→		READY-DECLARED
		←	Mute	
ATTRIB (несогласованный PUPI)	ATTRIB(0, 0, ~PUPI)	→		READY-DECLARED
		←	Mute	
REQB[f)] (несогласованный AFI)	REQB(1,nAFI)	→		IDLE[d)]
		←	Mute	
WUPB[f)] (несогласованный AFI)	WUPB(1,nAFI)	→		IDLE[d)]
		←	Mute	

[a)] n должно принимать все значения $2 \le n \le 16$.

[b)] Если PICC не поддерживает REQB/WUPB с N > 1 [см. ИСО/МЭК 14443-3:2011 (пункт 7.6.1)], то данное испытание не применяют.

[c)] При этом с вероятностью 1/16 PICC даст ответ ATQB и перейдет в состояние READY-DECLARED.

[d)] Подать команду ATTRIB и проверить, что ответа нет до проверки состояния IDLE.

[e)] Все согласованные AFI, определенные изготовителем PICC, должны быть проверены.

[f)] Несогласованные AFI, используемые в данном испытании, не должны быть RFU-значениями. Если PICC принимает все AFI, то данное испытание не подходит.

(Введены дополнительно. Изм. А3:2012.)

G.4.4.4.3 Протокол испытаний

Используя форму таблицы G.62, заполнить в ней соответствующую строку согласно таблице G.44.

Таблица G.44 — Критерии оценки результата испытания сценария G.25: Поведение PICC типа B в подсостоянии READY-DECLARED

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.4.4.5 Сценарий G.26: Поведение PICC типа B в состоянии HALT

G.4.4.5.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа B в состоянии HALT требованиям ИСО/МЭК 14443-3:2011 (пункт 7.4.8).

G.4.4.5.2 Процедура

Следует выполнить следующие шаги для каждой строки таблицы G.45.

a) Привести PICC в состояние HALT.

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Проверить, находится ли PICC в состоянии TTS.

Таблица G.45 — Переходы из состояния HALT

Переход	Оборудование для испытаний PICC		PICC	TTS
REQB[d)]	REQB(1)	→		HALT
		←	Mute	
WUPB[d)]	WUPB(1)	→		READY-DECLARED
		←	ATQB	
WUPB (неправильный CRC)	('05 00 08' ~CRC_B)	→		HALT
		←	Mute	
HLTB	HLTB	→		HALT
		←	Mute	
ATTRIB	ATTRIB(0,0)	→		HALT
		←	Mute	
Slot-MARKER	SLOTMARKER(n)[a)]	→		HALT
		←	Mute	
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→		HALT
		←	Mute	
DESELECT	S(DESELECT)	→		HALT
		←	Mute	
WUPB[e)] (несогласованный AFI)	WUPB(1,nAFI)	→		IDLE
		←	Mute	
REQB[e)] (несогласованный AFI)	REQB(1,nAFI)	→		HALT
		←	Mute	

Окончание таблицы G.45

Переход	Оборудование для испытаний PICC		PICC	TTS
HLTB (несогласованный PUPI)	HLTB(~PUPI)	→		HALT
		←	Mute	
ATTRIB (несогласованный PUPI)	ATTRIB(0, 0, ~PUPI)	→		HALT
		←	Mute	
WUPB[b]	WUPB(16)[c]	→		READY-REQUESTED
		←	Mute	

[a] n должно принимать все значения $2 \leq n \leq 16$.
[b] Если PICC не поддерживает REQB/WUPB с N > 1 [см. ИСО/МЭК 14443-3:2011 (пункт 7.6.1)], то данное испытание не применяют.
[c] При этом с вероятностью 1/16 PICC даст ответ ATQB и перейдет в состояние READY-DECLARED.
[d] Все согласованные AFI, определенные изготовителем PICC, должны быть проверены.
[e] Несогласованные AFI, используемые в данном испытании, не должны быть RFU-значениями. Если PICC принимает все AFI, то данное испытание не подходит.

(Введены дополнительно. Изм. А3:2012.)

G.4.4.5.3 Протокол испытаний

Используя форму таблицы G.62, заполнить в ней соответствующую строку согласно таблице G.46.

Таблица G.46 — Критерии оценки результата испытания сценария G.26: Поведение PICC типа B в состоянии HALT

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.4.4.6 Сценарий G.27: Поведение PICC типа B в состоянии PROTOCOL

G.4.4.6.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа B в состоянии PROTOCOL требованиям ИСО/МЭК 14443-4:2008. Данное испытание должно подтвердить, что активированная PICC не реагирует на какую-либо команду инициализации.

G.4.4.6.2 Процедура

Следует выполнить следующие шаги для каждой строки таблицы G.47.

a) Привести PICC в состояние PROTOCOL.

b) Выполнить переход состояния путем подачи команды, указанной в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Проверить, находится ли PICC в состоянии TTS.

Таблица G.47 — Переходы из состояния PROTOCOL

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
REQB[b]	REQB(1)	→			PROTOCOL
		←	Mute		
WUPB[b]	WUPB(1)	→			PROTOCOL
		←	Mute		
REQB (неправильный CRC)	('05 00 00' ~CRC_B)	→			PROTOCOL
		←	Mute		

Окончание таблицы G.47

Переход	Оборудование для испытаний PICC		PICC	FDT	TTS
WUPB (неправильный CRC)	('05 00 08' ~CRC_B)	→			PROTOCOL
		←	Mute		
HLTB	HLTB	→			PROTOCOL
		←	Mute		
ATTRIB	ATTRIB(0,0)	→			PROTOCOL
		←	Mute		
Slot-MARKER	SLOTMARKER(n)[a]	→			PROTOCOL
		←	Mute		
Команда по ИСО/МЭК 14443-4	$I(0)_0$(TEST_COMMAND1(1))	→			PROTOCOL
		←	$I(0)_0$(TEST_ RESPONSE1(1))	< FWT	
DESELECT	S(DESELECT)	→			HALT
		←	S(DESELECT)		
WUPB[c] (несогласованный AFI)	WUPB(1,nAFI)	→			PROTOCOL
		←	Mute		
REQB[c] (несогласованный AFI)	REQB(1,nAFI)	→			PROTOCOL
		←	Mute		
HLTB (несогласованный PUPI)	HLTB(~PUPI)	→			PROTOCOL
		←	Mute		
ATTRIB (несогласованный PUPI)	ATTRIB(0, 0, ~PUPI)	→			PROTOCOL
		←	Mute		

[a] n должно принимать все значения $2 \le n \le 16$.
[b] Все согласованные AFI, определенные изготовителем PICC, должны быть проверены.
[c] Несогласованные AFI, используемые в данном испытании, не должны быть RFU-значениями. Если PICC принимает все AFI, то данное испытание не подходит.

(Введены дополнительно. Изм. А3:2012.)

G.4.4.6.3 Протокол испытаний

Используя форму таблицы G.62, заполнить в ней соответствующую строку согласно таблице G.48.

Таблица G.48 — Критерии оценки результата испытания сценария G.27: Поведение PICC типа B в состоянии PROTOCOL

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.4.5 Сценарий G.28: Обработка антиколлизии типа B

G.4.5.1 Область применения

Цель данного испытания — определить, соответствует ли процедура обработки антиколлизии PICC типа B требованиям ИСО/МЭК 14443-3:2011 (пункт 7.4.1).

Основой этого испытания является процедура AnticollisionB(N, outparam chi2), которая определена в псевдокоде, приведенном ниже. Процедура выполняет 256 команд REQB(N) и последующие команды Slot-MARKER и подсчитывает, сколько раз каждый из N-слотов был выбран PICC. Также процедура проверяет, установила ли PICC соответствие каждого запроса REQB(N) только с одним слотом. Если нет, то испытание возвращает значение FAIL.

Поскольку антиколлизия типа B основана на случайном выборе слотов, то для проверки следует использовать статистические методы. Ввиду свойств всех статистических испытаний данное испытание может показать неудовлетворительный результат даже в том случае, если поведение PICC корректно. Такая ошибка носит название «ошибка I рода» в статистической терминологии. Этой ошибки невозможно полностью избежать. Вместо этого можно контролировать вероятность ее появления с помощью так называемого уровня значимости α. Это означает, что чем меньше α, тем менее вероятна ошибка I рода. Однако это не означает, что следует выбирать α как можно меньше. Причина состоит в том, что чем меньше α, тем больше вероятность того, что испытание пропустит неисправную PICC (т. е. PICC, которая не выбирает слоты с должной вероятностью). В статистической терминологиии это называется «ошибка II рода».

Кроме того, PICC должна выбирать каждый из N-слотов с равной вероятностью (т. е. 1/N). Для того чтобы проверить это, необходимо применить статистический χ^2-тест ко всем слотам. Результатом этого теста является значение chi2, которое необходимо сравнить с квантилью $\chi^2_{\alpha,N-1}$.

G.4.5.2 Процедура

Если один из статистических тестов показывает неудовлетворительный результат на шаге e), то испытательная лаборатория может перезапустить тест для этого параметра N.

Выполнить следующие шаги для каждого значения N = 2, 4, 8, 16.

a) Установить уровень значимости α на 0,005 и найти из таблицы G.49 соответствующее значение квантили $\chi^2_{\alpha,N-1}$. Другие значения α, указанные в таблице G.49, могут быть использованы как дополнительные по усмотрению заказчика испытаний.

b) Осуществить сброс PICC.

c) Выполнить AnticollisionB(N, chi2).

d) Если AnticollisionB возвращает FAIL, то испытание не выдержано.

e) Если chi2 ≤ $\chi^2_{\alpha,N-1}$, то испытание выдержано, в противном случае — не выдержано.

Таблица G.49 — Значения квантили α

α	$\chi^2_{\alpha,N-1}$			
	$\chi^2_{\alpha,1}$	$\chi^2_{\alpha,3}$	$\chi^2_{\alpha,7}$	$\chi^2_{\alpha,15}$
0,1 (опционально)	2,706	6,251	12,017	22,307
0,05 (опционально)	3,841	7,815	14,067	24,996
0,01 (опционально)	6,635	11,345	18,475	30,578
0,005	7,879	12,838	20,278	32,801

Псевдокод: Процедура антиколлизии типа B

```
1   Procedure AnticollisionB(N, chi2)
2
3     // TPDUSend and TPDURecv are PCD specific functions
4     // to send and receive TPDU frames
5
6     // probability for selecting slot
7     p = 1/N
8
9     // clear slot counters
10    for i from 1 to N do
11      Slots[i] = 0
12    Endfor
```

```
13
14        // collect data
15        for i from 1 to 256 do
16          Reset the PICC
17          TPDUSend (REQB(N))
18          if TPDURecv() = ATQB then
19            Slots[1] = Slots[1]+1
20          endif
21          for j from 2 to N do
22            TPDUSend (SLOTMARKER(j))
23            if TPDURecv () = ATQB then
24              Slots[j] = Slots[j]+1
25            endif
26          endfor
27        endfor
28
29        // check that exactly
30        // one slot has been selected at each run
31        cnt = 0
32        for i from 1 to N do
33          cnt = cnt + Slots[i]
34        endfor
35        if cnt ≠ 256 then
36          return FAIL
37        endif
38        Chi2 = 0
39        for i from 1 to N do
40          chi2 = chi2 + Slots[i]*Slots[i]
41        endfor
42        chi2 = chi2*N/256 - 256
43      return PASS
```

Примечание — Продолжить выполнение процедуры с шага e), только если в строке 43 возвращается PASS.

G.4.5.3 Протокол испытаний

Используя форму таблицы G.62, заполнить в ней соответствующую строку согласно таблице G.50.

Таблица G.50 — Критерии оценки результата испытания сценария G.28: Обработка антиколлизии типа B

Объяснение	Результат испытаний
Только если каждая процедура испытания на антиколлизию возвращает PASS	PASS
Если какая-либо процедура испытания на антиколлизию возвращает FAIL	FAIL

G.4.6 Обработка ATTRIB

G.4.6.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC типа B в ответ на команду ATTRIB требованиям ИСО/МЭК 14443-3:2011 (подраздел 7.10).

G.4.6.2 Процедура

Выполнить следующие шаги для каждого сценария G.29 и G.30, перечисленного в настоящем пункте.

a) Привести PICC в подсостояние READY-DECLARED.

b) Послать последовательность команд, указанных в графе «Оборудование для испытаний PICC».

c) Проверить, соответствует ли ответ PICC указанному в графе «PICC».

d) Проверить, находится ли PICC в состоянии PROTOCOL.

Сценарий G.29: ATTRIB с неправильным PUPI

Оборудование для испытаний PICC		PICC
('1D' ~PUPI '00 00 01 00' CRC_B)	→	
	←	Mute
ATTRIB(0,0)	→	
	←	ATA(0)

Сценарий G.30: ATTRIB после неправильного ATTRIB

Оборудование для испытаний PICC		PICC
('1D' PUPI '00 00 01 00' ~CRC_B)	→	
	←	Mute
ATTRIB(0,0)	→	
	←	ATA(0)

G.4.6.3 Протокол испытаний

Используя форму таблицы G.62, заполнить в ней соответствующую строку согласно таблице G.51.

Таблица G.51 — Критерии оценки результата испытания обработки ATTRIB

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.4.7 Сценарий G.31: Обработка максимального размера кадра

G.4.7.1 Область применения

Цель данного испытания — определить, принимает ли PICC размер FSD в соответствии с требованиями ИСО/МЭК 14443-3:2011 (пункт 7.10.4).

G.4.7.2 Процедура

Выполнить следующие шаги для каждого кода максимального размера кадра, определенного в ИСО/МЭК 14443-3.

(Измененная редакция. Изм. А4:2012.)

a) Привести PICC в подсостояние READY-DECLARED, как описано в G.4.4.1.1.

b) Подать команду ATTRIB(0, fsdi) с параметром fsdi, как в отдельном испытании.

c) Проверить, что ответом PICC является ATA(0).

Примечание — PICC может потребовать дополнительные последовательности, чтобы быть готовой принять TEST_COMMAND2(2).

d) Послать I-блок $I(0)_0$(TEST_COMMAND2(2)).
e) Проверить, что размер ответного I-блока, посланного PICC, ≤ FSD.
G.4.7.3 Протокол испытаний
Используя форму таблицы G.62, заполнить в ней соответствующую строку согласно таблице G.52.

Таблица G.52 — Критерии оценки результата испытания сценария G.31: Обработка максимального размера кадра

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.5 Методы испытаний для логических операций PICC типа A или типа B

G.5.1 Введение

Настоящий раздел содержит испытания, в которых проверяют, соответствует ли активированная PICC требованиям ИСО/МЭК 14443-4:2008. Настоящий раздел применим к PICC типа A и типа B.

G.5.1.1 Процесс активации PICC

Активация PICC представляет собой процесс приведения PICC в состояние, в котором возможен обмен блоками протокола, определенными в ИСО/МЭК 14443-4:2008. Данный процесс зависит от типа PICC.

Примечание — PICC может потребовать дополнительные последовательности, чтобы быть готовой принять шаг 1 сценария.

G.5.1.1.1 Активация PICC типа A
a) Привести PICC в состояние ACTIVE, как описано в G.3.3.2.1.
b) Послать RATS(cid, fsdi).
c) Проверить, что ответом PICC является действительный ATS.
G.5.1.1.2 Активация PICC типа B
a) Привести PICC в подсостояние READY-DECLARED, как описано в G.4.4.1.1.
b) Послать ATTRIB(cid, fsdi).
c) Проверить, что ответом PICC является действительный ATA.

G.5.2 Реакция PICC на сценарии по ИСО/МЭК 14443-4

G.5.2.1 Область применения

Цель данного испытания — определить, соответствует ли поведение PICC требованиям ИСО/МЭК 14443-4:2008 (раздел 7). Данное испытание использует реализации сценариев протокола по ИСО/МЭК 14443-4:2008 (приложение B).

G.5.2.2 Процедура

Выполнить следующие шаги для каждого сценария G.32 — G.54, перечисленного в настоящем пункте.
a) Активировать PICC, как описано в G.5.1.1, используя CID = 0 и FSDI = 0.
b) Для каждого шага в сценарии сделать следующее:
1) послать команду, описанную в графе «Оборудование для испытаний PICC»;
2) проверить, что ответ PICC совпадает с указанным в графе «PICC».
c) Завершить действия, указанные в перечислении b).

Сценарий G.32: Обмен I-блоками

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
2	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.33: Запрос на расширение времени ожидания

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND3)	→	
		←	S(WTX) (WTXM)
2	S(WTX) (WTXM)	→	
		←	$I(0)_0$(TEST_RESPONSE3)
3	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.34: DESELECT

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
2	S(DESELECT)	→	
		←	S(DESELECT)
3	REQA или REQB(1)[a)]	→	
		←	Mute
4	WUPA или WUPB(1)[a)]	→	
		←	ATQA или ATQB[a)]

[a)] Для PICC типа A используют левый вариант. Для PICC типа B используют правый вариант.

Сценарий G.35: PCD использует сцепление

Шаг	Оборудование для испытаний PICC		PICC
1	$I(1)_0$(TEST_COMMAND1$(2)_1$)	→	
		←	$R(ACK)_0$
2	$I(0)_1$(TEST_COMMAND1$(2)_2$)	→	
		←	$I(0)_1$(TEST_RESPONSE1(2))
3	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))

Сценарий G.36: PICC использует сцепление

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND2(2))	→	
		←	$I(1)_0$(TEST_RESPONSE2$(2)_1$)
2	$R(ACK)_1$	→	
		←	$I(0)_1$(TEST_RESPONSE2$(2)_2$)
3	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))

Сценарий G.37: Начало протокола

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND1(1), ~CRC)	→	
		←	Mute
2	$R(NAK)_0$	→	
		←	$R(ACK)_1$
3	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
4	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.38: Обмен I-блоками

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
2	$I(0)_1$(TEST_COMMAND1(1), ~CRC)	→	
		←	Mute
3	$R(NAK)_1$	→	
		←	$R(ACK)_0$
4	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))
5	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))

Сценарий G.39: Обмен I-блоками 1

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
2	$R(NAK)_0$	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
3	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.40: Обмен I-блоками 2

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
2	$R(NAK, \sim CRC)_0$	→	
		←	Mute
3	$R(NAK)_0$	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
4	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.41: Запрос на расширение времени ожидания

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND3)	→	
		←	S(WTX)(WTXM)
2	$R(NAK)_0$	→	
		←	S(WTX)(WTXM)
3	S(WTX)(WTXM)	→	
		←	$I(0)_0$(TEST_RESPONSE3)
4	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.42: Запрос на расширение времени ожидания

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND3)	→	
		←	S(WTX)(WTXM)
2	$R(NAK, \sim CRC)_0$	→	
		←	Mute
3	$R(NAK)_0$	→	
		←	S(WTX)(WTXM)
4	S(WTX)(WTXM)	→	
		←	$I(0)_0$(TEST_RESPONSE3)
5	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.43: Запрос на расширение времени ожидания

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND3)	→	
		←	S(WTX)(WTXM)
2	S(WTX)(WTXM, ~CRC)	→	
		←	Mute
3	$R(NAK)_0$	→	
		←	S(WTX)(WTXM)
4	S(WTX)(WTXM)	→	
		←	$I(0)_0$(TEST_RESPONSE3)
5	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.44: Запрос на расширение времени ожидания

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND3)	→	
		←	S(WTX)(WTXM)
2	S(WTX)(WTXM)	→	
		←	$I(0)_0$(TEST_RESPONSE3)
3	$R(NAK)_0$	→	
		←	$I(0)_0$(TEST_RESPONSE3)
4	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.45: Запрос на расширение времени ожидания

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND3)	→	
		←	S(WTX)(WTXM)
2	S(WTX)(WTXM)	→	
		←	$I(0)_0$(TEST_RESPONSE3)
3	$R(NAK, \sim CRC)_0$	→	
		←	Mute
4	$R(NAK)_0$	→	
		←	$I(0)_0$(TEST_RESPONSE3)
5	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.46: DESELECT

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
2	S(DESELECT, ~CRC)	→	
		←	Mute
3	S(DESELECT)	→	
		←	S(DESELECT)
4	REQA или REQB(1)[a]	→	
		←	Mute
5	WUPA или WUPB(1)[a]	→	
		←	ATQA или ATQB[a]

[a] Для PICC типа A используют левый вариант. Для PICC типа B используют правый вариант.

Сценарий G.47: PCD использует сцепление

Шаг	Оборудование для испытаний PICC		PICC
1	$I(1)_0$(TEST_COMMAND1(3)1)	→	
		←	$R(ACK)_0$
2	$R(NAK)_0$	→	
		←	$R(ACK)_0$
3	$I(1)_1$(TEST_COMMAND1$(3)_2$)	→	
		←	$R(ACK)_1$
4	$I(0)_0$(TEST_COMMAND1$(3)_3$)	→	
		←	$I(0)_0$(TEST_RESPONSE1(3))
5	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.48: PCD использует сцепление

Шаг	Оборудование для испытаний PICC		PICC
1	$I(1)_0$(TEST_COMMAND1$(3)_1$)	→	
		←	$R(ACK)_0$
2	$I(1)_1$(TEST_COMMAND1(3)2, ~CRC)	→	
		←	Mute
3	$R(NAK)_1$	→	
		←	$R(ACK)_0$
4	$I(1)_1$(TEST_COMMAND1$(3)_2$)	→	
		←	$R(ACK)_1$
5	$I(0)_0$(TEST_COMMAND1$(3)_3$)	→	
		←	$I(0)_0$(TEST_RESPONSE1(3))
6	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.49: PCD использует сцепление

Шаг	Оборудование для испытаний PICC		PICC
1	$I(1)_0$(TEST_COMMAND1(3)$_1$)	→	
		←	$R(ACK)_0$
2	$R(NAK, \sim CRC)_0$	→	
		←	Mute
3	$R(NAK)_0$	→	
		←	$R(ACK)_0$
4	$I(1)_1$(TEST_COMMAND1(3)$_2$)	→	
		←	$R(ACK)_1$
5	$I(0)_0$(TEST_COMMAND1(3)$_3$)	→	
		←	$I(0)_0$(TEST_RESPONSE1(3))
6	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.50: PICC использует сцепление

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND2(3))	→	
		←	$I(1)_0$(TEST_RESPONSE2(3)$_1$)
2	$R(ACK, \sim CRC)_1$	→	
		←	Mute
3	$R(ACK)_1$	→	
		←	$I(1)_1$(TEST_RESPONSE2(3)$_2$)
4	$R(ACK)_0$	→	
		←	$I(0)_0$(TEST_RESPONSE2(3)$_3$)
5	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.51: PICC использует сцепление

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND2(3))	→	
		←	$I(1)_0$(TEST_RESPONSE2(3)$_1$)
2	$R(ACK)_1$	→	
		←	$I(1)_1$(TEST_RESPONSE2(3)$_2$)
3	$R(ACK)_1$	→	
		←	$I(1)_1$(TEST_RESPONSE2(3)$_2$)
4	$R(ACK)_0$	→	
		←	$I(0)_0$(TEST_RESPONSE2(3)$_3$)
5	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.52: PICC использует сцепление

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND2(2))	→	
		←	$I(1)_0$(TEST_RESPONSE2(2)$_1$)
2	$R(NAK)_0$	→	
		←	$I(1)_0$(TEST_RESPONSE2(2)$_1$)
3	$R(ACK)_1$	→	
		←	$I(0)_1$(TEST_RESPONSE2(2)$_2$)
4	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))

Сценарий G.53: Метод проверки наличия PICC 1

Примечание — Этот сценарий заменяет прежний сценарий, который был исключен.

Шаг	Оборудование для испытаний PICC		PICC
1	$I(empty)_0$	→	
		←	$I()_0$
2	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))
3	$I(empty)_0$	→	
		←	$I()_0$

Сценарий G.54: Метод проверки наличия PICC 2

Примечание — Этот сценарий заменяет прежний сценарий, который был исключен.

Шаг	Оборудование для испытаний PICC		PICC
1	$R(NAK)_0$	→	
		←	$R(ACK)_1$
2	$R(NAK)_0$	→	
		←	$R(ACK)_1$
3	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
4	$R(NAK)_1$		
			$R(ACK)_0$
5	$I(0)_1$(TEST_COMMAND1(1))		
			$I(0)_1$(TEST_RESPONSE1(1))

G.5.2.3 Протокол испытаний

Используя форму таблицы G.63, заполнить в ней соответствующую строку согласно таблице G.53.

Таблица G.53 — Критерии оценки результата испытания реакции PICC на сценарии по ИСО/МЭК 14443-4

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.5.3 Обработка операций по обнаружению PICC ошибок

G.5.3.1 Область применения

Цель данного испытания — определить соответствие механизма обнаружения ошибок у PICC требованиям ИСО/МЭК 14443-4:2008 (пункт 7.5.7)[1].

G.5.3.2 Процедура

Выполнить следующие шаги для каждого сценария G.55 — G.57, перечисленного в настоящем подразделе.

a) Поместить эталонную PICC в область поля.

b) Активировать PICC, как описано в G.5.1.1, используя CID = 0 и FSDI = 0.

c) Для каждого шага в сценарии сделать следующее:

1) послать команду, описанную в графе «Оборудование для испытаний PICC»;

2) проверить, что ответ PICC совпадает с указанным в графе «PICC».

d) Завершить действия, указанные в перечислении c).

Примечание — Графа «Комментарий» следующих сценариев ссылается на правила ИСО/МЭК 14443-4:2008 (пункты 7.5.4—7.5.7)[2].

Сценарий G.55: Неправильный CRC в I-блоке

Шаг	Оборудование для испытаний PICC		PICC	Комментарий
1	$I(0)_0$(TEST_COMMAND1(1), ~CRC)	→		Подпункт 7.5.7.1[3], перечисление a)
		←	Mute	
2	$I(0)_0$(TEST_COMMAND1(1))	→		
		←	$I(0)_0$(TEST_RESPONSE1(1))	

Сценарий G.56: Неправильный CRC в сцепленном I-блоке

Шаг	Оборудование для испытаний PICC		PICC	Комментарий
1	$I(1)_0$(TEST_COMMAND1$(2)_1$)	→		
		←	$R(ACK)_0$	
2	$I(0)_1$(TEST_COMMAND1$(2)_2$, ~CRC)	→		Подпункт 7.5.7.1[3], перечисление a)
		←	Mute	
3	$I(0)_1$(TEST_COMMAND1$(2)_2$)	→		
		←	$I(0)_1$(TEST_RESPONSE1(2))	

Сценарий G.57: Неправильный CRC в S(WTX)-блоке

Шаг	Оборудование для испытаний PICC		PICC	Комментарий
1	$I(0)_0$(TEST_COMMAND3)	→		
		←	S(WTX) (WTXM)	
2	S(WTX)(WTXM, ~CRC)	→		Подпункт 7.5.7.1[3], перечисление a)
		←	Mute	
3	S(WTX)(WTXM)	→		
		←	$I(0)_0$(TEST_RESPONSE3)	

[1] В ИСО/МЭК 10373-6:2011 указан пункт 7.5.6 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

[2] В ИСО/МЭК 10373-6:2011 указаны пункты 7.5.3—7.5.6 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

[3] В ИСО/МЭК 10373-6:2011 указан подпункт 7.5.6.1 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

G.5.3.3 Протокол испытаний

Используя форму таблицы G.63, заполнить в ней соответствующую строку согласно таблице G.54.

Таблица G.54 — Критерии оценки результата испытания обработки операций по обнаружению PICC ошибок

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.5.4 Реакция PICC на CID

G.5.4.1 Область применения

Цель данного испытания — определить соответствие реакции PICC на CID требованиям ИСО/МЭК 14443-4:2008 (подпункт 7.1.1.2).

G.5.4.2 Процедура

Выполнить следующие шаги для каждого сценария G.58 — G.62 и G.66[1], перечисленного в настоящем подразделе. Следует использовать соответствующую таблицу контрольных примеров CID в зависимости от того, поддерживает PICC CID или нет.

Для каждой строки в таблицах контрольных примеров CID (см. таблицу G.55 или таблицу G.56) выполнить следующее.

a) Активировать PICC с cid_{ass}, указанным в графе «Присвоенный CID».

b) Выполнить обмен блоками, как указано в соответствующем сценарии. Использовать cid_{cmd}, указанный в графе «Командный CID» в таблице контрольных примеров CID.

c) Проверить, совпадает ли ответ PICC с ответом в графе «PICC» в сценарии. Если для PICC указаны два варианта ответа, то уникальный ожидаемый ответ определяют из графы «Ожидаемый ответ PICC» в таблице контрольных примеров CID.

Таблица G.55 — Таблица контрольных примеров CID (для PICC, поддерживающих CID)

Номер теста[a)]	Присвоенный CID (cid_{ass})	Командный CID (cid_{cmd})	Ожидаемый ответ PICC
1	1	1	Ответ 1 сценария
2	0	0	Ответ 1 сценария
3	0	CID отсутствует	Ответ 1 сценария
4	1	CID отсутствует	Ответ 2 сценария (Mute)
5	0	1	Ответ 2 сценария (Mute)
6	1	0	Ответ 2 сценария (Mute)
7	2	1	Ответ 2 сценария (Mute)
[a)] Для каждого номера теста должен быть применен каждый из описанных сценариев.			

Таблица G.56 — Таблица контрольных примеров CID (для PICC, не поддерживающих CID)

Номер теста[a)]	Присвоенный CID (cid_{ass})	Командный CID (cid_{cmd})	Ожидаемый ответ PICC
1	0	0	Ответ 2 сценария (Mute)
2	0	CID отсутствует	Ответ 1 сценария
3	0	1	Ответ 2 сценария (Mute)
4[b)]	1	CID отсутствует	Ответ 1 сценария
[a)] Для каждого номера теста должен быть применен каждый из описанных сценариев. [b)] Применять только к PICC типа A.			

[1] Сценарий G.66 введен изменением А3:2012 ИСО/МЭК 10373-6:2011 далее по тексту, но не предусмотрен в данной ссылке на сценарии.

Сценарий G.58: CID в I-блоке

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND1(1), CID = cid_{cmd})	→	
		←	Ответ 1: $I(0)_0$(TEST_RESPONSE1(1), CID = cid_{cmd}) Ответ 2: Mute[a)]

[a)] Ответ 1 или ответ 2 — в соответствии с таблицей G.55 или таблицей G.56.

Сценарий G.59: CID в I-блоке со сцеплением

Шаг	Оборудование для испытаний PICC		PICC
1	$I(1)_0$(TEST_COMMAND1$(2)_1$,CID = cid_{cmd})	→	
		←	Ответ 1: R(ACK, CID = cid_{cmd})$_0$ Ответ 2: Mute[a)]

[a)] Ответ 1 или ответ 2 — в соответствии с таблицей G.55 или таблицей G.56.

Сценарий G.60: CID в R-блоке

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND2(3), CID[a)] = cid_{ass})	→	
		←	$I(1)_0$(TEST_RESPONSE2$(3)_1$, CID = cid_{ass})
2	R(ACK, CID[a)] = cid_{cmd})$_1$	→	
		←	Ответ 1: $I(1)_1$ (TEST_RESPONSE2$(3)_2$, CID = cid_{cmd}) Ответ 2: Mute[b)]

[a)] Для PICC, не поддерживающих CID, не использовать CID.
[b)] Ответ 1 или ответ 2 — в соответствии с таблицей G.55 или таблицей G.56.

Сценарий G.61: CID в S(WTX)-блоке

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND3, CID[a)] = cid_{ass})	→	
		←	S(WTX)(WTXM, CID = cid_{ass})
2	S(WTX)(WTXM, CID[a)] = cid_{cmd})	→	
		←	Ответ 1: $I(0)_0$(TEST_RESPONSE3, CID[a)] = cid_{cmd}) Ответ 2: Mute[b)]

[a)] Для PICC, не поддерживающих CID, не использовать CID.
[b)] Ответ 1 или ответ 2 — в соответствии с таблицей G.55 или таблицей G.56.

Сценарий G.62: CID в S(DESELECT)-блоке

Шаг	Оборудование для испытаний PICC		PICC
1	S(DESELECT, CID = cid_{cmd})	→	
		←	Ответ 1: S(DESELECT, CID = cid_{cmd}) Ответ 2: Mute[a)]

[a)] Ответ 1 или ответ 2 — в соответствии с таблицей G.55 или таблицей G.56.

Сценарий G.66: CID в S(PARAMETERS)-блоке

Шаг	Оборудование для испытаний PICC		PICC
1	S(PARAMETERS, CID = cid_{cmd})	→	
		←	Ответ 1: S(PARAMETERS, CID = cid_{cmd}) Ответ 2: Mute[a)]

[a)] Ответ 1 или ответ 2 — в соответствии с таблицей G.55 или таблицей G.56.

(Введен дополнительно. Изм. А3:2012.)

G.5.4.3 Протокол испытаний

Используя форму таблицы G.63, заполнить в ней соответствующую строку согласно таблице G.57.

Таблица G.57 — Критерии оценки результата испытания реакции PICC на CID

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.5.5 Реакция PICC на NAD

G.5.5.1 Область применения

Цель данного испытания — определить соответствие реакции PICC на кодирование NAD требованиям ИСО/МЭК 14443-4:2008 (подпункт 7.1.1.3).

G.5.5.2 Процедура

Выполнить следующие шаги для каждого сценария G.63 — G.65, перечисленного в настоящем подразделе.

Активировать PICC, как описано в G.5.1.1, используя CID = 0 и FSDI = 0.

Для каждого шага в сценарии выполнить следующее:

a) подать команду, как описано в графе «Оборудование для испытаний PICC»;

b) проверить, совпадает ли ответ PICC с указанным в графе «PICC».

Пусть n — произвольно выбранное значение действительного NAD с b4 и b8, установленными в состояние (0)b.

Сценарий G.63: NAD в I-блоке (для PICC, поддерживающих NAD)

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND1(1), NAD = n)	→	
		←	$I(0)_0$(TEST_RESPONSE1(1), содержащий NAD)

Сценарий G.64: NAD в сцепленном I-блоке (для PICC, поддерживающих NAD)

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND2(3), NAD = n)	→	
		←	$I(1)_0$(TEST_RESPONSE2$(3)_1$, содержащий NAD)
2	$R(ACK)_1$	→	
		←	$I(1)_1$(TEST_RESPONSE2$(3)_2$, не содержащий NAD)

Сценарий G.65: NAD в I-блоке (для PICC, не поддерживающих NAD)

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND1(1), NAD = n)	→	
		←	Mute

G.5.5.3 Протокол испытаний

Используя форму таблицы G.63, заполнить в ней соответствующую строку согласно таблице G.58.

Таблица G.58 — Критерии оценки результата испытания реакции PICC на NAD

Объяснение	Результат испытаний
Если сценарий не применим к PICC	Не имеет отношения (N/A)
Если сценарий применим к PICC и только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

G.5.6 Реакция PICC на блоки S(PARAMETERS)

G.5.6.1 Область применения

Цель данного испытания — определить соответствие поведения PICC требованиям ИСО/МЭК 14443-4:2008/ Изм. А1:2012 (пункт 7.5.1). В данном испытании используют реализации протокола по сценариям ИСО/МЭК 14443-4:2008/ Изм. А1:2012 (пункт В.2.6 приложения В).

G.5.6.2 Процедура

Выполнить следующие шаги для каждого сценария G.67 — G.70, перечисленного в настоящем разделе.

a) Активировать PICC, как описано в G.5.1.1, используя CID = 0 и FSDI = 0.

b) Для каждого шага в сценарии сделать следующее:

1) послать команду, описанную в графе «Оборудование для испытаний PICC»;

2) проверить, что ответ PICC совпадает с указанным в графе «PICC».

c) Завершить действия, указанные в перечислении b).

Сценарий G.67: Обмен дополнительными параметрами

Шаг	Оборудование для испытаний PICC		PICC
1	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))
2	S(PARAMETERS)	→	
		←	S(PARAMETERS)
3	$I(0)_1$(TEST_COMMAND1(1))	→	
		←	$I(0)_1$(TEST_RESPONSE1(1))

Сценарий G.68: Обмен дополнительными параметрами

Шаг	Оборудование для испытаний PICC		PICC
1	S(PARAMETERS)	→	
		←	S(PARAMETERS)
2	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))

Сценарий G.69: Обмен дополнительными параметрами

Шаг	Оборудование для испытаний PICC		PICC
1	S(PARAMETERS)	→	
		←	S(PARAMETERS)
2	S(PARAMETERS)	→	
		←	S(PARAMETERS)
3	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))

Сценарий G.70: Обмен дополнительными параметрами

Шаг	Оборудование для испытаний PICC		PICC
1	S(PARAMETERS, ~CRC)	→	
		←	Mute
2	S(PARAMETERS)	→	
		←	S(PARAMETERS)
3	$I(0)_0$(TEST_COMMAND1(1))	→	
		←	$I(0)_0$(TEST_RESPONSE1(1))

G.5.6.3 Протокол испытаний

Используя форму таблицы G.63, заполнить в ней соответствующую строку согласно таблице Amd.3-1.

Таблица Amd.3-1 — Критерии оценки результата испытания обмена дополнительными параметрами

Объяснение	Результат испытаний
Только если реакция PICC соответствует указанной в процедуре	PASS
Любой другой случай	FAIL

(Введены дополнительно. Изм. А3:2012.)

G.6 Результаты, включаемые в протокол испытаний

Таблица G.59 — Таблица временных соотношений для PICC типа A

№	Параметр	Ссылка на стандарт	Минимальное значение	Максимальное значение	Измеренное(ые) значение(я)
1	Время задержки кадра при передаче от PCD к PICC (для команд REQA,WUPA, ANTICOLLISION, SELECT)	ИСО/МЭК 14443-3:2011 (подпункт 6.2.1.1)	Последний бит (1)b → 1236/*fc* Последний бит (0)b → 1172/*fc*	Последний бит (1) b → 1236/*fc* + 0,4 мкс Последний бит (0) b → 1172/*fc* + 0,4 мкс	
2	Время ожидания кадра деактивации и команды RATS	ИСО/МЭК 14443-4:2008 (подраздел 8.1)	Последний бит (1)b → 1236/*fc* Последний бит (0)b → 1172/*fc*	65536/fc (~4,8 мс)	
3	Время задержки кадра при передаче от PCD к PICC (для кадров, не указанных в предыдущих строках)	ИСО/МЭК 14443-3:2011 (подпункт 6.2.1.1) и ИСО/МЭК 14443-4:2008 (подраздел 8.1)	Последний бит (1)b → 1236/*fc* Последний бит (0)b → 1172/*fc*	(256/*fs*) × 2^{FWI} (~302,06 мкс × 2^{FWI})	FWI = Max FDT =

Примечание — Все временные значения вычисляют для несущей частоты *fc* = 13,56 МГц и скорости передачи = *fc*/128 (~106 кбит/с).

Таблица G.60 — Таблица временных соотношений для PICC типа B

№	Параметр	Ссылка на стандарт	Минимальное значение	Максимальное значение	Измеренное(ые) значение(я)
1	Низкий SOF	ИСО/МЭК 14443-3:2011 (пункт 7.1.4)	10 etu (~94,40 мкс)	11 etu (~103,83 мкс)	
2	Высокий SOF	ИСО/МЭК 14443-3:2011 (пункт 7.1.4)	2 etu (~18,88 мкс)	3 etu (~28,32 мкс)	
3	Низкий EOF	ИСО/МЭК 14443-3:2011 (пункт 7.1.5)	10 etu (~94,40 мкс)	11 etu (~103,83 мкс)	
4	Границы бит	ИСО/МЭК 14443-3:2011 (пункт 7.1.1)	(n — 1/8) etu	(n + 1/8) etu	
5	EGT при передаче от PICC к PCD	ИСО/МЭК 14443-3:2011 (пункт 7.1.2)	0 мкс	19 мкс	
6	TR0 для ATQB	ИСО/МЭК 14443-3:2011 (пункт 7.1.6)	64/*fs* (~75,52 мкс)	256/*fs* (~302,06 мкс)	
7	TR1 для ATQB	ИСО/МЭК 14443-3:2011 (пункт 7.1.6)	80/*fs* (~94,40 мкс)	200/*fs* (~235,99 мкс)	
8	TR0 (не ATQB)	ИСО/МЭК 14443-3:2011 (пункт 7.1.6), ИСО/МЭК 14443-3:2011 (пункт 7.10.3)	Значение по ИСО/МЭК 14443-3:2011 (таблица 30)	$(256/fs) \times 2^{FWI}$ (~302,06 мкс $\times 2^{FWI}$)	FWI = Max TR0 =
9	TR1 (не ATQB)	ИСО/МЭК 14443-3:2011 (пункт 7.1.6), ИСО/МЭК 14443-3:2011 (пункт 7.10.3)	Значение по ИСО/МЭК 14443-3:2011 (таблица 31)	200/*fs* (~235,99 мкс)	
11[1]	Задержка с момента завершения EOF и остановки поднесущей	ИСО/МЭК 14443-3:2011 (пункт 7.1.7)	0	2 etu	
12	Время ожидания кадра деактивации	ИСО/МЭК 14443-4:2008 (подраздел 8.1)	64/*fs* + 80/*fs* (~169,92 мкс)	65536/*fc* (~4,8 мс)	

(Измененная редакция. Изм. А3:2012.)

Примечание — Все временные значения вычисляют для несущей частоты *fc* = 13,56 МГц и скорости передачи = *fc*/128 (~106 кбит/с).

[1] Изменением А3:2012 строка № 10 из таблицы G.60 исключена.

Таблица G.61 — Результаты для методов испытаний PICC типа A

Метод испытаний по ИСО/МЭК 10373-6		Номер сценария ИСО/МЭК 10373-6	Результат испытания: PASS или FAIL или N/A
Подраздел	Параметр		
G.3.2	Опрос	Сценарий G.1	
G.3.3	Испытания переходов состояний PICC типа A	Сценарий G.2	
		Сценарий G.3	
		Сценарий G.4	
		Сценарий G.5	
		Сценарий G.6	
		Сценарий G.7	
		Сценарий G.8	
		Сценарий G.9	
		Сценарий G.10	
		Сценарий G.11	
		Сценарий G.12	
G.3.4	Обработка антиколлизии типа A	Сценарий G.13	
G.3.6	Обработка PPS-запроса	Сценарий G.17	
		Сценарий G.18	
		Сценарий G.19	
G.3.7	Обработка FSD	Сценарий G.20	

Таблица G.62 — Результаты для методов испытаний PICC типа B

Метод испытаний по ИСО/МЭК 10373-6		Номер сценария ИСО/МЭК 10373-6	Результат испытания: PASS или FAIL или N/A
Подраздел	Параметр		
G.4.2	Опрос	Сценарий G.21	
G.4.3	Прием PICC	Сценарий G.22	
G.4.4	Испытания переходов состояний PICC типа B	Сценарий G.23	
		Сценарий G.24	
		Сценарий G.25	
		Сценарий G.26	
		Сценарий G.27	
G.4.5	Обработка антиколлизии типа B	Сценарий G.28	
G.4.6	Обработка ATTRIB	Сценарий G.29	
		Сценарий G.30	
G.4.7	Обработка максимального размера кадра	Сценарий G.31	

Таблица G.63 — Результаты для методов испытаний логической операции PICC типа A или B

Метод испытания по ИСО/МЭК 10373-6		Номер сценария ИСО/МЭК 10373-6	Результат испытаний PASS или FAIL или N/A
Подраздел/ пункт	Параметр		
G.5.2	Реакция PICC на сценарии по ИСО/МЭК 14443-4	Сценарий G.32	
		Сценарий G.33	
		Сценарий G.34	
		Сценарий G.35	
		Сценарий G.36	
		Сценарий G.37	
		Сценарий G.38	
		Сценарий G.39	
		Сценарий G.40	
		Сценарий G.41	
		Сценарий G.42	
		Сценарий G.43	
		Сценарий G.44	
		Сценарий G.45	
		Сценарий G.46	
		Сценарий G.47	
		Сценарий G.48	
		Сценарий G.49	
		Сценарий G.50	
		Сценарий G.51	
		Сценарий G.52	
		Сценарий G.53	
		Сценарий G.54	
G.5.3	Обработка операций по обнаружению PICC ошибок	Сценарий G.55	
		Сценарий G.56	
		Сценарий G.57	
G.5.4	Реакция PICC на CID	Сценарий G.58	
		Сценарий G.59	
		Сценарий G.60	
		Сценарий G.61	
		Сценарий G.62	
		Сценарий G.66	
G.5.5	Реакция PICC на NAD	Сценарий G.63	
		Сценарий G.64	
		Сценарий G.65	
G.5.6	Реакция PICC на блоки S(PARAMETERS)	Сценарий G.67	
		Сценарий G.68	
		Сценарий G.69	
		Сценарий G.70	
G.1.5.1	RFU-значения		

(Введены дополнительно. Изм. А3:2012.)

Приложение Н
(обязательное)

Дополнительные методы испытаний PCD

Н.1 Оборудование для испытаний PCD и дополнительные приспособления

В настоящем разделе определены испытательное оборудование и испытательные схемы для проверки работы PCD на соответствие требованиям ИСО/МЭК 14443-3:2011 и ИСО/МЭК 14443-4:2008.

Н.1.1 Метод испытаний

Для испытаний протокола по ИСО/МЭК 14443 между испытуемым PCD и LT выбирают абстрактную модель по ИСО/МЭК 9646 и используют метод локального тестирования.

Н.1.2 Структура оборудования для испытаний PCD

Оборудование для испытаний PCD состоит из двух частей (см. рисунок Н.1):

- верхний тестер (UT) (может быть персональным компьютером с интерфейсом главного компьютера, подходящим для испытуемого PCD);
- нижний тестер (LT).

Испытуемое PCD рассматривают как тестируемую реализацию (IUT).

Если PCD встроено в изделие, то оно включает в себя UT. В таком случае некоторые испытания могут быть не применимы. Кроме того, в случае если в стандарте не определены конкретные требования, то метод испытания будет заключаться только в протоколировании функциональных возможностей.

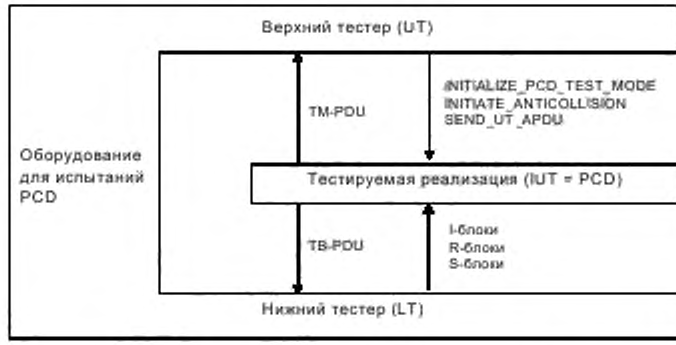


Рисунок Н.1 — Принципиальная архитектура тестера

Компонента LT оборудования для испытаний PCD включает в себя:

- аппаратное обеспечение, эмулирующее PICC, и программные средства, способные эмулировать протоколы типа А и типа В;
- цифровой импульсный осциллограф (см. 5.1.1).

Н.1.3 Интерфейс оборудования для испытаний PCD

UT и IUTобмениваются информацией с помощью TM-PDU (Test Management PDU). Определение TM-PDU зависит от реализации, его предоставляет изготовитель IUT, и он должен инициировать действия, указанные в таблице Н.1.

Таблица Н.1 — Команды логического интерфейса

	Наименование TM-PDU	Требуемое действие IUT
1	INITIALIZE_PCD_TEST_MODE	Возврат в состояние «включено» (IUT должна вступить в цикл антиколлизии). IUT возвращает UT результирующий код его действия
2	INITIATE_ANTICOLLISION	Инициировать последовательность антиколлизии (если IUT запускает последовательность антиколлизии автоматически в результате инициализации, то последовательность может быть пустой). IUT возвращает UT результирующий код его действия
3	SEND_UT_APDU	Передать UT_APDU через радиочастотный интерфейс LT и вернуть UT результирующий код действия IUT. Ответ от IUT должен включать в себя ответ LT на посланный UT_APDU

Оборудование для испытаний PCD должно быть способно инициализировать служебную информацию IUT, предоставляемую изготовителем IUT, посредством интерфейса UT и конфигурировать себя для выполнения необходимых процедур, протоколов и анализа посредством интерфейса LT.

H.1.4 Эмуляция протокола на I/O

Оборудование для испытаний PCD должно быть способно посредством интерфейса LT эмулировать протоколы типа A и типа B и приложения PICC, необходимые для исполнения сценария. LT должен быть способен разбивать передаваемые пакеты на цепочки блоков требуемой длины.

Должна быть предусмотрена возможность конфигурировать LT, для того чтобы симулировать различные опции:

- конфигурации NAD и CID;
- размер кадра, скорость передачи или другой параметр, необходимый для реализации методов испытаний.

H.1.5 Генерирование временных соотношений знаков на I/O в режиме передачи

Оборудование для испытаний должно быть способно посредством интерфейса LT генерировать поток битов на I/O в соответствии с ИСО/МЭК 14443-3:2011. Временные параметры: продолжительность стартового бита, дополнительный разграничительный интервал (только для типа B), продолжительность бита, время задержки кадра, ширина начала кадра и ширина конца кадра — должны быть конфигурируемыми. Для реализации испытаний типа A LT должен быть способен симулировать коллизию бит в выбранной(ых) битовой(ых) позиции(ях).

H.1.6 Измерение и контроль радиочастотного протокола на I/O

Оборудование для испытаний PCD должно обеспечивать посредством интерфейса LT возможность измерять и контролировать временные соотношения низкого и высокого логических состояний, передаваемых PCD.

H.1.7 Анализ протокола

Оборудование для испытаний PCD должно быть способно анализировать поток битов на I/O посредством интерфейса LT в соответствии с протоколами типа A и типа B по ИСО/МЭК 14443-3:2011 и ИСО/МЭК 14443-4:2008 и извлекать поток логических данных для дальнейшего анализа протокола.

H.1.8 Процедура активации протокола

H.1.8.1 Процедура активации для методов испытания антиколлизии

Активировать LT путем выполнения следующей последовательности.

a) Конфигурировать LT для эмуляции протокола типа A или типа B.

b) UT должен послать PCD TM-PDU INITIALIZE_PCD_TEST_MODE.

c) UT должен послать PCD TM-PDU INITIATE_ANTICOLLISION.

H.1.8.2 Процедура активации для методов испытания протокола типа A

Активировать LT путем выполнения следующей последовательности.

a) Конфигурировать LT для эмуляции протокола типа A.

b) UT должен послать PCD TM-PDU INITIALIZE_PCD_TEST_MODE.

c) UT должен послать PCD TM-PDU INITIATE_ANTICOLLISION. PCD должно применить последовательность антиколлизии по ИСО/МЭК 14443-3:2011 (раздел 6) (запрос, цикл антиколлизии и выбор). PCD должно применить последовательность активации протокола по ИСО/МЭК 14443-4:2008 (раздел 5).

d) PCD сообщает UT результат процедуры активации.

H.1.8.3 Процедура активации для методов испытания протокола типа B

Активировать LT путем выполнения следующей последовательности.

a) Конфигурировать LT для эмуляции протокола типа B.

b) UT должен послать PCD TM-PDU INITIALIZE_PCD_TEST_MODE.

c) UT должен послать PCD TM-PDU INITIATE_ANTICOLLISION. PCD должно применить последовательность активации протокола по ИСО/МЭК 14443-3:2011 (раздел 7).

d) PCD сообщает UT результат процедуры активации.

H.1.9 Сценарий

H.1.9.1 Описание

Для испытаний PCD, как установлено в настоящем стандарте, необходимо использовать сценарий. Этот сценарий представляет собой типовую коммуникацию для протокола и приложения, зависящую от их функциональности, обусловленной предполагаемым применением PCD и реализованной в нем.

Типовой сценарий представляет собой набор командных TM-PDU, определенных в H.1.3.

Сценарий должен быть определен исполнителем этих испытаний и отражен в протоколе испытаний наряду с результатами испытаний. Сценарий должен охватывать репрезентативное подмножество или, если целесообразно, полный набор функций PCD, предполагаемых к использованию при нормальном применении PCD.

Примечание — Исполнителю испытаний может потребоваться информация о реализованном протоколе и функциональности.

Посылаемый UT_APDU может представлять собой одну из следующих команд:

- UT_TEST_COMMAND1, выбираемую оборудованием для испытаний PCD и указывающую инструкцию ИСО, используемую как инструкция по умолчанию для сценариев, не требующих сцепления от PCD (в случае если PCD решает все-таки применить сцепление, то сценарий должен быть адаптирован соответствующим образом испытательной лабораторией);

- UT_TEST_COMMAND2, выбираемую оборудованием для испытаний PCD и указывающую инструкцию ИСО, используемую как инструкция по умолчанию для сценариев со сцеплением, обеспечиваемым PCD.

H.1.9.2 Пример сценария

Типовой сценарий:

```
INITIALIZE_PCD_TEST_MODE
INITIATE_ANTICOLLISION
SEND_UT_APDU (UT_TEST_COMMAND1)
SEND_UT_APDU (UT_TEST_COMMAND2)
...
```

H.1.10 Поведение UT, LT и PCD

В следующих перечислениях обобщенно изложено поведение UT, LT и PCD.

a) UT запускает процедуру активации по H.1.8.

b) Если процедура активации прошла с ошибкой, то PCD переходит к обработке исключительной ситуации. Обработка этой исключительной ситуации может включать в себя передачу UT сообщения об ошибке.

c) В случае методов испытания антиколлизии оборудование для испытаний PCD завершает испытание на этом шаге. Для методов испытания протокола UT переходит к следующему шагу.

d) UT посылает PCD первую команду UT_APDU.

e) PCD должно передать LT эту команду (UT_APDU), используя TB-PDU. PCD разделяет текущую UT_APDU на соответствующие TB-PDU (I-блоки), посылает первый I-блок LT и ожидает ответный блок. PCD управляет коммуникационными блоками в соответствии с ИСО/МЭК 14443-4:2008.

f) LT принимает командный UT_APDU. LT посылает PCD ответный UT_APDU. LT управляет коммуникационными блоками (TB-PDU) в соответствии с ИСО/МЭК 14443-4:2008 (LT может использовать механизм сцепления в любое время, даже если это не предписано максимальным размером кадра PCD или PICC). PCD должен передать ответный UT_APDU, полученный от LT, назад UT.

g) Если выполнение команды прервалось на уровне протокола (т. е. ошибка обнаружена PCD), то PCD переходит к обработке исключительной ситуации. Обработка исключительной ситуации может включать в себя передачу UT сообщения об ошибке.

h) Если выполнение команды прошло успешно, то PCD передает сообщение UT о положительном результате. В этом случае, если сценарий определяет посылку LT дополнительного UT_APDU, то UT посылает PCD следующий UT_APDU. Этот цикл продолжается, пока не будет послан последний UT_APDU.

H.1.11 Взаимосвязь методов испытаний и требований базового стандарта

Все испытания, указанные в таблицах H.2, H.3 и H.4, должны быть выполнены, а их результаты отражены в соответствующих таблицах по H.6.

Таблица H.2 — Методы испытаний для типа A

Метод испытаний по ИСО/МЭК 10373-6		Соответствующее требование	
Подраздел	Наименование	Базовый стандарт	Пункт/подпункт
H.2.1	Время задержки кадра при передаче от PICC к PCD	ИСО/МЭК 14443-3:2011	6.2.1.2
H.2.2	Разграничительный интервал запроса	ИСО/МЭК 14443-3:2011	6.2.2
H.2.3	Обработка коллизии бит во время ATQA	ИСО/МЭК 14443-3:2011	6.5.2
H.2.4	Обработка цикла антиколлизии	ИСО/МЭК 14443-3:2011	6.5.3
H.2.5	Обработка RATS и ATS	ИСО/МЭК 14443-4:2008	5.6.1.1
H.2.6	Обработка PPS-ответа	ИСО/МЭК 14443-4:2008	5.6.2.1
H.2.7	Механизм выбора размера кадра	ИСО/МЭК 14443-4:2008	5.2.3
H.2.8	Обработка разграничительного интервала кадра при запуске	ИСО/МЭК 14443-4:2008	5.2.5
H.2.9	Обработка CID во время активации со стороны PCD	ИСО/МЭК 14443-4:2008	5.6.3

Таблица Н.3 — Метод испытаний для типа В

Метод испытаний по ИСО/МЭК 10373-6		Соответствующее требование	
Подраздел	Наименование	Базовый стандарт	Подраздел
Н.3.1	Временные соотношения передачи на I/O	ИСО/МЭК 14443-3:2011	7.1
Н.3.2	Механизм выбора размера кадра	ИСО/МЭК 14443-3:2011	7.9
Н.3.3	Обработка CID во время активации со стороны PCD	ИСО/МЭК 14443-3:2011	7.10

Таблица Н.4 — Метод испытаний для логических операций

Метод испытаний по ИСО/МЭК 10373-6		Соответствующее требование	
Подраздел	Наименование	Базовый стандарт	Структурный элемент
Н.4.1	Обработка цикла опроса	ИСО/МЭК 14443-3:2011	5
Н.4.2	Реакция PCD на запрос расширения времени ожидания	ИСО/МЭК 14443-4:2008	7.3
Н.4.3	Обнаружение и исправление ошибок	ИСО/МЭК 14443-4:2008	7.5.7[1]
Н.4.4	Обработка NAD во время сцепления	ИСО/МЭК 14443-4:2008	7.1.1.3

Н.2 Методы испытаний для типа А

Н.2.1 Время задержки кадра при передаче от PICC к PCD

Цель данного испытания — определить время между кадром PICC и следующим кадром PCD.

Н.2.1.1 Оборудование

См. Н.1.

Н.2.1.2 Процедура

Поместить LT в рабочую область PCD.

Во время следующей процедуры радиочастотные данные входа/приема должны быть непрерывно контролируемы и проверяемы на соответствие ИСО/МЭК 14443-2:2010. Все переходы сигнала (уровень и временные соотношения), а также логическое содержание коммуникации должны быть записаны.

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с Н.1.8.1.

b) LT ожидает, пока PCD пошлет ему действительный кадр команды REQA/WUPA.

c) LT отвечает действительным ATQA.

d) LT ожидает, пока PCD пошлет ему действительную команду антиколлизии в соответствии с ИСО/МЭК 14443-3:2011 (рисунок 7).

e) Измерить время между последней модуляцией, переданной LT, и первой паузой, переданной PCD [см. ИСО/МЭК 14443-3:2011, (подпункт 6.2.1.2)].

Н.2.1.3 Протокол испытаний

В протокол испытаний включить запись сигнала. Используя форму таблицы Н.21, заполнить в ней строку 1 измеренным значением времени задержки кадра. Используя форму таблицы Н.23, заполнить в ней соответствующую строку.

Н.2.2 Разграничительный интервал запроса

Цель данного испытания — определить разграничительный интервал запроса двух последовательных команд REQA/WUPA. Данное испытание относится к PCD, которые посылают последовательно REQA/WUPA.

Н.2.2.1 Оборудование

См. Н.1.

Н.2.2.2 Процедура

Поместить LT в рабочую область PCD.

Во время следующей процедуры радиочастотные данные входа/приема должны быть непрерывно контролируемы и проверяемы на соответствие ИСО/МЭК 14443-2:2010. Все переходы сигнала (уровень и временные соотношения), а также логическое содержание коммуникации должны быть записаны.

Использовать следующую последовательность.

[1] В ИСО/МЭК 10373-6:2011 указан пункт 7.5.6 без учета изменения нумерации пунктов, введенного изменением А1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

a) UT выполняет процедуру активации в соответствии с H.1.8.1.

b) LT ожидает, пока PCD пошлет ему действительный кадр команды REQA/WUPA. LT остается в состоянии Mute.

c) LT ожидает, пока PCD пошлет ему действительный кадр команды REQA/WUPA. LT остается в состоянии Mute.

d) Измерить время между стартовыми битами двух последовательных REQA/WUPA [см. ИСО/МЭК 14443-3:2011 (пункт 6.2.2)].

H.2.2.3 Протокол испытаний

В протокол испытаний включить запись сигнала. Используя форму таблицы H.21, заполнить в ней строку 2 измеренным значением разграничительного интервала запроса. Используя форму таблицы H.23, заполнить в ней соответствующую строку.

H.2.3 Обработка коллизии бит во время ATQA

Цель данного испытания — определить наличие обработки коллизии бит со стороны PCD во время ATQA.

H.2.3.1 Оборудование

См. H.1.

H.2.3.2 Процедура

Поместить LT в рабочую область PCD и записать наличие и содержание команд PCD.

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с H.1.8.1.

b) LT ожидает, пока PCD пошлет ему действительный кадр команды REQA/WUPA.

c) Поддержать LT, чтобы получить ответ ATQA, используя симуляцию коллизии бит на бите N (N от 1 до 16). Коллизия на бите вызывает также коллизию в соответственном бите четности.

H.2.3.3 Протокол испытаний

Используя форму таблицы H.23, заполнить в ней соответствующую строку согласно таблице H.5.

Таблица H.5 — Критерий оценки результата испытания обработки коллизии бит во время ATQA

Объяснение	Результат испытаний
Только если PCD начинает бит-ориентированный цикл антиколлизии	PASS
Любой другой случай	FAIL

H.2.4 Обработка цикла антиколлизии

Цель данного испытания — определить соответствие обработки цикла антиколлизии бит требованиям ИСО/МЭК 14443-3:2011 (пункт 6.5.3).

H.2.4.1 Оборудование

См. H.1.

H.2.4.2 Процедура

Поместить LT в рабочую область PCD и записать наличие и содержание команд PCD.

H.2.4.2.1 Процедура 1 (одиночный размер UID)

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с H.1.8.1.

b) LT ожидает, пока PCD пошлет ему действительный кадр команды REQA/WUPA.

c) LT отвечает ATQA, указывающим на биткадровую антиколлизию и размер UID: одиночный [биты b8 и b7 равны (00)b].

d) PCD должно послать команду ANTICOLLISION со значением '93 20' (каскадный уровень 1).

e) LT отвечает UID CL1(uid0 uid1 uid2 uid3 BCC).

f) PCD должно послать команду SELECT со значением '93 70' uid0 uid1 uid 2 uid3 BCC CRC_A.

g) LT отвечает SAK [каскадный бит освобожден, b3 = (0)b], указывая, что UID полный.

Сценарий H.1 — Обработка цикла антиколлизии для PICC с одиночным размером UID (процедура 1)

Испытание	PCD		LT	Фаза
REQA/WUPA	REQA/WUPA	→		
		←	ATQA (одиночный размер UID)	1
ANTICOLLISION Уровень 1	Команда ANTICOLLISION Уровень 1 ('93 20')	→		
		←	UID CL1 (uid0 uid1 uid2 uid3 BCC)	2
SELECT	Команда SELECT ('93 70' uid0 uid1 uid2 uid3 BCC CRC_A)	→		
		←	SAK(complete)	3

H.2.4.2.2 Процедура 2 (двойной размер UID)

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с H.1.8.1.

b) LT ожидает, пока PCD пошлет ему действительный кадр команды REQA/WUPA.

c) LT отвечает ATQA, указывающим на биткадровую антиколлизию и размер UID: двойной [биты b8 и b7 равны (01)b].

d) PCD должно послать команду ANTICOLLISION со значением '93 20' (каскадный уровень 1).

e) LT отвечает UID CL1 ('88' uid0 uid1 uid2 BCC).

f) PCD должно послать команду SELECT со значением '93 70 88' uid0 uid1 uid2 BCC CRC_A.

g) LT отвечает SAK [каскадный бит установлен, b3 = (1)b].

h) PCD должно увеличить каскадный уровень и послать команду ANTICOLLISION со значением '95 20' (каскадный уровень 2).

i) LT отвечает UID CL2 (uid3 uid4 uid5 uid6 BCC).

j) PCD должно послать команду SELECT со значением '95 70' uid3 uid4 uid5 uid6 BCC CRC_A.

k) LT отвечает SAK [каскадный бит освобожден, b3 = (0)b], указывая, что UID полный.

Сценарий H.2 — Обработка цикла антиколлизии для PICC с двойным размером UID (процедура 2)

Испытание	PCD		LT	Фаза
REQA/WUPA	REQA/WUPA	→		
		←	ATQA (двойной размер UID)	1
ANTICOLLISION Уровень 1	Команда ANTICOLLISION Уровень 1 ('93 20')	→		
		←	UID CL1 ('88' uid0 uid1 uid2 BCC)	2
SELECT	Команда SELECT ('93 70 88' uid0 uid1 uid2 BCC CRC_A)	→		
		←	SAK(cascade)	3
ANTICOLLISION Уровень 2	Команда ANTICOLLISION Уровень 2 ('95 20')	→		
		←	UID CL2 (uid3 uid4 uid5 uid6 BCC)	4
SELECT	Команда SELECT ('95 70' uid3 uid4 uid5 uid6 BCC CRC_A)	→		
		←	SAK(complete)	5

H.2.4.2.3 Процедура 3 (тройной размер UID)

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с H.1.8.1.

b) LT ожидает, пока PCD пошлет ему действительный кадр команды REQA/WUPA.

c) LT отвечает ATQA, указывающим на биткадровую антиколлизию и размер UID: тройной [биты b8 и b7 равны (10)b].

d) PCD должно послать команду ANTICOLLISION со значением '93 20' (каскадный уровень 1).

e) LT отвечает UID CL1 ('88' uid0 uid1 uid2 BCC).

f) PCD должно послать команду SELECT со значением '93 70 88' uid0 uid1 uid2 BCC CRC_A.

g) LT отвечает SAK [каскадный бит установлен, b3 = (1)b].

h) PCD должно увеличить каскадный уровень и послать команду ANTICOLLISION со значением '95 20' (каскадный уровень 2).

i) LT отвечает UID CL2 ('88' uid3 uid4 uid5 BCC).

j) PCD должно послать команду SELECT со значением '95 70 88' uid3 uid4 uid5 BCC CRC_A.

k) LT отвечает SAK [каскадный бит установлен, b3 = (1)b].

l) PCD должно увеличить каскадный уровень и послать команду ANTICOLLISION со значением '97 20' (каскадный уровень 3).

m) LT отвечает UID CL3 (uid6 uid7 uid8 uid9 BCC).

n) PCD должно послать команду SELECT со значением '97 70' uid6 uid7 uid8 uid9 BCC CRC_A.

o) LT отвечает SAK [каскадный бит освобожден, b3 = (0)b], указывая, что UID полный.

Сценарий H.3 — Обработка цикла антиколлизии для PICC с тройным размером UID (процедура 3)

Испытание	PCD		LT	Фаза
REQA/WUPA	REQA/WUPA	→		
		←	ATQA (тройной размер UID)	1
ANTICOLLISION Уровень 1	Команда ANTICOLLISION Уровень 1 ('93 20')	→		
		←	UID CL1 ('88' uid0 uid1 uid2 BCC)	2
SELECT	Команда SELECT ('93 70 88' uid0 uid1 uid2 BCC CRC_A)	→		
		←	SAK(cascade)	3
ANTICOLLISION Уровень 2	Команда ANTICOLLISION Уровень 2 ('95 20')	→		
		←	UID CL2 ('88' uid3 uid4 uid5 BCC)	4
SELECT	Команда SELECT ('95 70 88' uid3 uid4 uid5 BCC CRC_A)	→		
		←	SAK(cascade)	5
ANTICOLLISION Уровень 3	Команда ANTICOLLISION Уровень 3 ('97 20')	→		
		←	UID CL3 (uid6 uid7 uid8 uid9 BCC)	6
SELECT	Команда SELECT ('97 70' uid6 uid7 uid8 uid9 BCC CRC_A)	→		
		←	SAK(complete)	7

H.2.4.2.4 Процедура 4 (полная побитовая антиколлизия, одиночный размер UID)

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с H.1.8.1.

b) LT ожидает, пока PCD пошлет ему действительный кадр команды REQA или WUPA.

c) LT отвечает ATQA, указывающим на биткадровую антиколлизию и размер UID: одиночный [биты b8 и b7 равны (00)b].

d) PCD должно послать команду ANTICOLLISION со значением '93 20'.

e) LT отвечает потоком из 40 бит путем эмуляции коллизии на каждом бите, включая биты четности.

f) Повторить шаги с g) по h) для значений k от 1 до 31.

g) PCD должно послать команду ANTICOLLISION со значением '93' NVB $UIDTX_1$[[1..k]], где $UIDTX_1$[[1..k-1]] либо пустое (т. е. k = 1), либо значение уже известно PCD и $UIDTX_1$[[k]] представляет собой произвольный бит, выбранный PCD.

h) LT отвечает потоком 40 минус k бит путем эмуляции коллизии на каждом бите, включая биты четности.

i) PCD может дополнительно использовать команду ANTICOLLISION: '93 60' $UIDTX_1$[[1..32]]. В этом случае LT отвечает BCC.

Примечание — Дополнительная команда ANTICOLLISION не изменяет номер фазы.

j) PCD должно послать команду SELECT со значением '93 70' $UIDTX_1$[[1..32]] BCC CRC_A с BCC, вычисленным PCD, если оно не использовало дополнительный шаг i).

k) LT отвечает SAK [каскадный бит освобожден, b3 = (0)b], указывая, что UID полный.

Сценарий Н.4 — Обработка полного побитового цикла антиколлизии для PICC (процедура 4)

Испытание	PCD		LT	Фаза
REQA/WUPA	REQA/WUPA	→		
		←	ATQA (одиночный размер UID)	1
ANTICOLLISION	Команда ANTICOLLISION ('93 20')	→		
		←	Кадр полной коллизии из 40 бит	2
ANTICOLLISION (k бит $UID_{PARTIAL}$) $1 \le k \le 31$	Команда ANTICOLLISION ('93' NVB $UIDTX_1$[[1..k]])	→		
		←	Кадр коллизии из 40 минус k бит	k+2
OPTIONAL ANTICOLLISION (32 бит $UID_{PARTIAL}$)	Команда ANTICOLLISION ('93 60' $UIDTX_1$[[1..32]])	→		
		←	(BCC)	
SELECT	КомандаSELECT ('93 70' $UIDTX_1$[[1..32]] BCC CRC_A)	→		
		←	SAK(complete)	34

Н.2.4.3 Протокол испытаний

Используя форму таблицы Н.23, заполнить в ней соответствующую строку согласно таблице Н.6.

Таблица Н.6 — Критерии оценки результата испытания обработки цикла антиколлизии

Объяснение	Результат испытаний
Только если поведение PCD совпадает с ожидаемым по сценарию для каждой процедуры	PASS
Любой другой случай	FAIL

Н.2.5 Обработка RATS и ATS

Цель данного испытания — определить соответствие обработки RATS и ATS со стороны PCD требованиям ИСО/МЭК 14443-4:2008 (подпункт 5.6.1.1).

Н.2.5.1 Оборудование

См. Н.1.

Н.2.5.2 Процедура

Поместить LT в рабочую область PCD и записать наличие и содержание команд PCD.

Н.2.5.2.1 Процедура 1

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с Н.1.8.1.

b) LT отвечает сообщениями, относящимися к антиколлизии, и ожидает, пока PCD пошлет действительный кадр команды RATS.

c) LT не реагирует на RATS (Mute).

d) PCD может послать действительный кадр команды RATS.

e) Если PCD посылает вторую RATS, то LT не реагирует на RATS (Mute).

f) PCD должно начать последовательность деактивации, определенную в ИСО/МЭК 14443-4:2008 (раздел 8).

g) Повторить процедуру с ошибочным кадром ATS (использовать ошибочный CRC_A) вместо Mute.

Сценарий Н.5 — Обработка RATS и ATS (процедура 1)

Испытание	PCD		LT
	Кадр команды RATS (например, 'E0 01' CRC_A)	→	
Mute		←	Mute
	Кадр команды RATS (например, 'E0 01' CRC_A)	→	
		←	Mute
Начало деактивации	DESELECT	→	
	Кадр команды RATS (например, 'E0 01' CRC_A)	→	
Ошибочный кадр ATS		←	Ошибочный кадр ATS[a)]
	Кадр команды RATS (например, 'E0 01' CRC_A)	→	
		←	Ошибочный кадр ATS[a)]
Начало деактивации	DESELECT	→	
[a)] Определен на шаге g).			

Н.2.5.2.2 Процедура 2

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с Н.1.8.1.

b) LT отвечает сообщениями, относящимися к антиколлизии, и ожидает, пока PCD пошлет действительный кадр команды RATS.

c) LT отвечает действительным ATS без байта TA.

d) PCD должно вернуть UT результирующий код его действия (в соответствии с таблицей Н.1).

e) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

f) PCD должно послать LT какой-нибудь I-блок (включая пустой), возможно после контрольных последовательностей наличия PICC.

Сценарий Н.6 — Обработка RATS и ATS (процедура 2)

Испытание	PCD		LT
Правильный ATS	Кадр команды RATS (например, 'E0 01' CRC_A)	→	
		←	ATS
Продолжение операции	Какой-нибудь I-блок (включая пустой)	→	

Н.2.5.2.3 Процедура 3

В случае если PCD не использует необязательную повторную передачу RATS в соответствии в ИСО/МЭК 14443-4:2008 (подпункт 5.6.1.1), то данную процедуру пропустить.

В случае если PCD использует необязательную повторную передачу RATS в соответствии в ИСО/МЭК 14443-4:2008 (подпункт 5.6.1.1), то использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с Н.1.8.1.

b) LT отвечает сообщениями, относящимися к антиколлизии, и ожидает, пока PCD пошлет действительный кадр команды RATS.

c) Если PCD передает RATS, то LT не реагирует на RATS (Mute).

d) Если PCD повторно передает RATS, то LT отвечает действительным ATS.

e) PCD должно послать LT какой-либо I-блок (включая пустой), возможно после контрольных последовательностей наличия PICC или PPS-запроса.

f) Повторить процедуру с ошибочным кадром ATS (использовать ошибочный CRC_A) вместо Mute.

Сценарий Н.7 — Обработка RATS и ATS (процедура 3)

Испытание	PCD		LT
Mute или ошибочный ATS	Кадр команды RATS (например, 'E0 01' CRC_A)	→	
		←	Mute/ошибочный кадр ATS[a)]
PCD передает RATS	Кадр команды RATS (например, 'E0 01' CRC_A)	→	
		←	ATS
Продолжение операции	Какой-нибудь I-блок (включая пустой) или PPS-запрос	→	
[a)] Определен на шаге f).			

Н.2.5.3 Протокол испытаний

Используя форму таблицы Н.23, заполнить в ней соответствующую строку согласно таблице Н.7.

Таблица Н.7 — Критерии оценки результата испытания обработки RATS и ATS

Объяснение	Результат испытаний
Только если поведение PCD совпадает с ожидаемым по сценарию для каждой процедуры	PASS
Любой другой случай	FAIL

Н.2.6 Обработка PPS-ответа

Цель данного испытания — определить соответствие обработки PPS-запроса требованиям ИСО/МЭК 14443-4:2008 (подпункт 5.6.2.1). Данное испытание применимо только к PCD, которое использует механизм выбора протокола и параметров как часть последовательности активации PICC. В случае если PCD не использует механизм PPS, то в графе протокола испытаний указывают N/A.

Н.2.6.1 Оборудование

См. Н.1.

Н.2.6.2 Предварительная процедура

Использовать следующую последовательность для приведения PCD в состояние, необходимое для данного испытания.

a) Поместить LT в рабочую область PCD и записать наличие и содержание команд PCD.

b) UT выполняет процедуру активации в соответствии с Н.1.8.1.

c) LT отвечает сообщениями, относящимися к антиколлизии, и ожидает, пока PCD пошлет RATS.

d) LT отвечает ATS (с действительным TA<> '00', указывающим, что поддерживаются более высокие скорости передачи, и поэтому PICC поддерживает PPS, и таким образом PCD может выполнять PPS-последовательность).

Н.2.6.2.1 Процедура 1

Использовать следующую последовательность сразу после процедуры по Н.2.6.2.

a) LT ожидает, пока PCD пошлет ему действительный PPS-запрос. Следует убедиться, что стартовый байт PPSS, Parameter 0 и Parameter 1 не содержат RFU-значения.

Примечание — Необязательно посылать PPS-запрос.

b) LT отвечает PPS-ответом.

c) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

d) PCD должно передать I-блок, используя выбранные параметры, возможно после контрольных последовательностей наличия PICC.

Сценарий Н.8 — Обработка PPS-запроса и PPS-ответа (процедура 1)

Испытание	PCD		LT
Правильный PPS-ответ	PPS-запрос	→	
		←	PPS-ответ
Установление параметров и продолжение операции	I-блок, полученный от UT	→	

H.2.6.2.2 Процедура 2

Использовать следующую последовательность сразу после процедуры по H.2.6.2.

a) LT ожидает, пока PCD пошлет ему действительный PPS-запрос.

b) LT отвечает ошибочным PPS-ответом (использовать ошибочный CRC_A).

c) PCD может повторно передать действительный PPS-запрос или продолжить операцию (например, послать I-блок), используя в обоих случаях скорость передачи по умолчанию.

d) Повторить процедуру без ответа на PPS-запрос (Mute).

Сценарий H.9 — Обработка PPS-запроса и PPS-ответа (процедура 2)

Испытание	PCD		LT
Ошибочный PPS-ответ или Mute	PPS-запрос	→	
		←	Ошибочный PPS-ответ или Mute[a)]
	Дополнительный PPS-запрос или I-блок	→	
[a)] Определено на шаге d).			

H.2.6.3 Протокол испытаний

Используя форму таблицы H.23, заполнить в ней соответствующую строку согласно таблице H.8.

Таблица H.8 — Критерии оценки результата испытания обработки PPS-запроса и PPS-ответа

Объяснение	Результат испытаний
Только если поведение PCD совпадает с ожидаемым по сценарию для каждой процедуры	PASS
Любой другой случай	FAIL

H.2.7 Механизм выбора размера кадра

Цель данного испытания — проверить правильность обработки передаваемого размера кадра. Передаваемые кадры не должны быть длиннее, чем указывает FSCI. Данное испытание выполняют как минимум при FSCI, установленном в состояние значений '0', '1', '8' и 'C'.

(Измененная редакция. Изм. A4:2012.)

H.2.7.1 Оборудование

См. H.1.

H.2.7.2 Процедура

Поместить LT в рабочую область PCD и записать наличие и содержание команд PCD.

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с H.1.8.1.

b) LT отвечает сообщениями, относящимися к антиколлизии, и ожидает, пока PCD пошлет действительный кадр команды RATS.

c) LT отвечает действительным ATS. Для цели данного испытания LT возвращает байт формата T0, равный '70' [см. ИСО/МЭК 14443-4:2008 (пункт 5.2.3)]. В случае PPS-запроса LT ответит на него прежде, чем начнется следующий шаг: максимальный размер кадра, допускаемый LT, — в соответствии с FSCI.

d) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND2), где длина данных должна быть больше максимального размера кадра, допускаемого LT.

e) PCD должно послать следующий $I(1)_0$ блок максимальной длиной в соответствии с FSCI.

Сценарий Н.10 — Механизм выбора размера кадра

PCD		LT
Кадр команды RATS (например, 'E0 01' CRC_A)	→	
	←	ATS, T0 = '70'
Возможен необязательный PPS-запрос, который будет обработан LT перед I-блоком	→	
	←	PPS-ответ в случае PPS-запроса
$I(1)_0$ (максимальная длина — в соответствии с FSCI)	→	

Н.2.7.3 Протокол испытаний

Используя форму таблицы Н.23, заполнить в ней соответствующую строку согласно таблице Н.9.

Таблица Н.9 — Критерии оценки результата испытания механизма выбора размера кадра

Объяснение	Результат испытаний
Только если поведение PCD совпадает с ожидаемым по сценарию для всех испытуемых значений FSCI	PASS
Любой другой случай	FAIL

Н.2.8 Обработка разграничительного интервала кадра при запуске

Цель данного испытания — определить соответствие временных соотношений передачи PCD требованиям ИСО/МЭК 14443-4:2008 (пункт 5.2.5).

Данное испытание выполняют как минимум при SFGI, установленном в состояние значений 0, 1 и 14.

Н.2.8.1 Оборудование

См. Н.1.

Н.2.8.2 Процедура

Поместить LT в рабочую область PCD и записать наличие и содержание команд PCD.

Во время следующей процедуры радиочастотные данные входа/приема должны быть непрерывно контролируемы и проверяемы на соответствие ИСО/МЭК 14443-2:2010. Все переходы сигнала (уровень и временные соотношения), а также логическое содержание коммуникации должны быть записаны.

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с Н.1.8.1.

b) LT отвечает сообщениями, относящимися к антиколлизии, и ожидает, пока PCD пошлет действительный кадр команды RATS.

c) LT отвечает действительным ATS. Для цели данного испытания LT возвращает байт интерфейса TB(1), равный '0E' [см. ИСО/МЭК 14443-4:2008 (пункт 5.2.5)]. В случае PPS-запроса LT ответит на него прежде, чем начнется следующий шаг.

Значение '0E'= (00001110)b означает: минимальное значение задержки кадра, допускаемое LT, составляет $(256 \times 16/fc) \times 2^{14}$ (~4949 мс).

d) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

e) PCD должно послать следующий $I(0 \text{ или } 1)_0$ блок после минимальной задержки 4949 мс.

Сценарий Н.11 — Механизм разграничительного интервала кадра при запуске

PCD		LT
Кадр команды RATS (например, 'E0 01' CRC_A)	→	
	←	ATS, TB(1) = '0E'
Возможен необязательный PPS-запрос, который будет обработан LT перед I-блоком	→	
	←	PPS-ответ в случае PPS-запроса
$I(0 \text{ или } 1)_0$ [послан после SFGT = $(256 \times 16/fc) \times 2^{14}$ (~4949 мс)]	→	

H.2.8.3 Протокол испытаний

Используя форму таблицы H.23, заполнить в ней соответствующую строку согласно таблице H.10.

Таблица H.10 — Критерии оценки результата испытания обработки разграничительного интервала кадра при запуске

Объяснение	Результат испытаний
Только если поведение PCD совпадает с ожидаемым по сценарию для всех испытуемых значений SFGI	PASS
Любой другой случай	FAIL

H.2.9 Обработка CID во время активации со стороны PCD

Цель данного испытания — определить соответствие обработки CID требованиям ИСО/МЭК 14443-4:2008 (пункт 5.6.3). Данное испытание проводят для CID, как минимум установленного в состояние значений 0, 1 и 14, если CID может быть выбран UT. В противном случае должен быть использован только CID, выбранный PCD.

H.2.9.1 Оборудование

См. H.1.

H.2.9.2 Процедура

Использовать следующие шаги a) — c) для приведения PCD в состояние, необходимое для данного испытания.

a) Поместить LT в рабочую область PCD и записать наличие и содержание команд PCD.

b) UT выполняет процедуру активации в соответствии с H.1.8.1.

c) LT отвечает сообщениями, относящимися к антиколлизии, и ожидает, пока PCD пошлет RATS. LT отвечает ATS.

Для каждого испытания из сценария H.12, если его поддерживает PCD, использовать следующую последовательность шагов d) — h).

d) Привести PCD в состояние, необходимое для данного испытания.

e) LT ожидает, пока PCD применит команду, описанную в графе «PCD».

f) LT отвечает, как описано в графе «LT».

g) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

h) PCD должно послать LT I-блок, применяя условие, описанное в графе «PCD».

Сценарий H.12 — Обработка CID

Испытание	PCD		LT
CID = n, не равному 0, и прием CID поддерживается	RATS (CID не равен 0)	→	
		←	ATS (CID поддерживается)
	Любая действительная команда, использующая CID	→	
CID = n, не равному 0, и прием CID не поддерживается	RATS (CID не равен 0)	→	
		←	ATS (CID не поддерживается)
	Любая действительная команда без CID	→	
CID = n, равному 0, и прием CID поддерживается	RATS (CID равен 0)	→	
		←	ATS (CID поддерживается)
	Любая действительная команда, использующая CID = 0, или без CID	→	
CID = n, равному 0, и прием CID не поддерживается	RATS (CID равен 0)	→	
		←	ATS (CID не поддерживается)
	Любая действительная команда без CID	→	

H.2.9.3 Протокол испытаний

Используя форму таблицы H.23, заполнить в ней соответствующую строку согласно таблице H.11.

Таблица Н.11 — Критерии оценки результата испытания обработки CID во время активации со стороны PCD

Объяснение	Результат испытаний
Только если поведение PCD совпадает с ожидаемым по сценарию	PASS
Любой другой случай	FAIL

Н.3 Методы испытаний для типа B

Н.3.1 Временные соотношения передачи на I/O

Цель данного испытания — определить соответствие временных соотношений передачи PCD требованиям ИСО/МЭК 14443-3:2011 (подраздел 7.1).

Н.3.1.1 Оборудование

См. Н.1.

Н.3.1.2 Процедура

Поместить LT в область рабочего поля PCD.

Во время следующей процедуры радиочастотные данные входа/приема должны быть непрерывно контролируемы и проверяемы на соответствие ИСО/МЭК 14443-2:2010. Все переходы сигнала (уровень и временные соотношения), а также логическое содержание коммуникации должны быть записаны.

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с Н.1.8.1.

b) Анализировать временные границы бит в пределах знака, посылаемого PCD [см. ИСО/МЭК 14443-3:2011 (пункт 7.1.1)].

c) Анализировать дополнительный разграничительный интервал (EGT) между двумя последовательными знаками, посланными PCD [см. ИСО/МЭК 14443-3:2011 (пункт 7.1.2)].

d) Анализировать временные соотношения SOF, посылаемого PCD [см. ИСО/МЭК 14443-3:2011 (пункт 7.1.4)].

e) Анализировать временные соотношения EOF, посылаемого PCD [см. ИСО/МЭК 14443-3:2011, (пункт 7.1.5)].

f) Анализировать временные соотношения перед SOF, посылаемым PCD [см. ИСО/МЭК 14443-3:2011 (пункт 7.1.7)].

Н.3.1.3 Протокол испытаний

В протокол испытаний включить запись сигнала. Используя форму таблицы Н.22, заполнить ее значениями. Измеренными на шагах b) — f). Используя форму таблицы Н.24, заполнить в ней соответствующую строку.

Н.3.2 Механизм выбора размера кадра

Цель данного испытания — анализ механизма выбора размера кадра на соответствие требованиям ИСО/МЭК 14443-3:2011 (подраздел 7.9).

Данное испытание выполняют при коде максимального размера кадра, как минимум установленном в состояние значений '0', '1', '8' и 'C'.

(Измененная редакция. Изм. А4:2012.)

Н.3.2.1 Оборудование

См. Н.1.

Н.3.2.2 Процедура

Поместить LT в рабочую область PCD и записать наличие и содержание команд PCD.

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с Н.1.8.1.

b) LT ожидает, пока PCD пошлет ему действительный кадр команды REQB/WUPB.

c) LT отвечает ATQB. Принимают, что PUPI для LT равен '12 23 34 45' и LT поддерживает CID. Для цели данного испытания LT возвращает второй информационный байт протокола, равный '01' [см. ИСО/МЭК 14443-3:2011 (пункт 7.9.4)], который означает, что максимальный размер кадра, поддерживаемый LT, — 16 байтов, и LT соответствует ИСО/МЭК 14443-4:2008.

d) PCD должно послать действительный кадр команды ATTRIB.

e) LT посылает ответ на команду ATTRIB.

f) PCD должно вернуть UT результирующий код его действия в соответствии с таблицей Н.1.

g) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND2).

h) PCD должно послать $I(1)_0$ блок максимальной длиной 16 байтов, как показано в графе «PCD» сценария Н.13.

Сценарий Н.13 — Механизм выбора размера кадра

PCD		LT
REQB/WUPB	→	
	←	ATQB
Кадр команды ATTRIB	→	
	←	Ответ на команду ATTRIB
$I(1)_0$ (INF = первая цепочка включает в себя первый блок UT_TEST_COMMAND2) максимальной длиной 16 байтов	→	

Н.3.2.3 Протокол испытаний

Используя форму таблицы Н.24, заполнить в ней соответствующую строку согласно таблице Н.12.

Таблица Н.12 — Критерии оценки результата испытания механизма выбора размера кадра

Объяснение	Результат испытаний
Только если поведение PCD совпадает с ожидаемым по сценарию для кода максимального размера кадра, как минимум установленного в состояние значений '0', '1', '8' и 'C'[1]	PASS
Любой другой случай	FAIL

Н.3.3 Обработка CID во время активации со стороны PCD

Цель данного испытания — определить соответствие обработки CID требованиям ИСО/МЭК 14443-3:2011.

Данное испытание выполняют для CID, как минимум установленного в состояние значений 0, 1 и 14, если CID может быть выбран UT. В противном случае используют только CID, выбранный PCD.

Н.3.3.1 Оборудование

См. Н.1.

Н.3.3.2 Процедура

Поместить LT в рабочую область PCD и записать наличие и содержание команд PCD.

Н.3.3.2.1 Процедура 1

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с Н.1.8.1.

b) LT ожидает, пока PCD пошлет ему команду REQB/WUPB.

c) LT посылает ATQB с битами (b2, b1) Frame Option, равными (00)b. Это означает: CID и NAD не поддерживаются.

d) LT ожидает, пока PCD пошлет ему команду ATTRIB. PCD должно послать действительный кадр команды ATTRIB с байтом Param 4, равным 0.

e) LT посылает ответ на команду ATTRIB со значением CID, равным 0.

f) PCD должно вернуть UT результирующий код его действия в соответствии с таблицей Н.1.

g) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

h) PCD должно послать следующий $I(0 \text{ или } 1)_0$ блок без NAD и CID.

Сценарий Н.14 — Обработка CID (процедура 1)

PCD		LT
REQB/WUPB	→	
	←	ATQB Биты Frame Option (b2, b1) = (00)b
Кадр команды ATTRIB (например, '1D 12 23 34 45 00 05 01 00' CRC_B)	→	
	←	Ответ на команду ATTRIB
$I(0 \text{ или } 1)_0$ без NAD и CID	→	

[1] В ИСО/МЭК 10373-6:2011 значение 'C' не указано. Здесь указано для согласования с измененной редакцией второго абзаца подраздела Н.3.2.

H.3.3.2.2 Процедура 2

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с H.1.8.1.

b) LT ожидает, пока PCD пошлет ему команду REQB/WUPB.

c) LT посылает ATQB с битами (b2, b1) Frame Option, равными (01)b. Это означает: CID поддерживается, а NAD не поддерживается.

d) LT ожидает, пока PCD пошлет ему команду ATTRIB. PCD должно послать действительный кадр команды ATTRIB, используя байт Param 4, равный '0X' (CID = X в диапазоне от 0 до 14).

e) LT посылает ответ на команду ATTRIB со значением CID, присвоенным PCD на шаге d), в байте Param 4.

f) PCD должно вернуть UT результирующий код его действия в соответствии с таблицей H.1.

g) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

h) PCD должно послать следующий $I(0 \text{ или } 1)_0$ блок, используя значение CID, присвоенное PCD на шаге d), или, на выбор, не используя CID, если CID = 0. PCD не должно использовать NAD в этом $I(0 \text{ или } 1)_0$ блоке.

Сценарий H.15 — Обработка CID (процедура 2)

PCD		LT
REQB/WUPB	→	
	←	ATQB Биты Frame Option (b2, b1) = (01)b
Кадр команды ATTRIB с Param 4 = '0X' (например, '1D 12 23 34 45 00 05 01 '0X' CRC_B), с X в диапазоне от 0 до 14	→	
	←	Ответ на команду ATTRIB
$I(0 \text{ или } 1)_0$ со значением CID, равным '0X', и без NAD	→	

H.3.3.3 Протокол испытаний

Используя форму таблицы H.24, заполнить в ней соответствующую строку согласно таблице H.13.

Таблица H.13 — Критерии оценки результата испытания обработки CID во время активации со стороны PCD

Объяснение	Результат испытаний
Только если поведение PCD совпадает с ожидаемым по сценарию для каждой процедуры	PASS
Любой другой случай	FAIL

H.4 Метод испытания для логических операций PCD

Все методы, описанные в настоящем разделе, за исключением H.4.1, должны быть применены дважды — один раз для сигнального интерфейса типа A и один раз для сигнального интерфейса типа B.

H.4.1 Обработка цикла опроса

Цель данного испытания — определить поведение PCD во время опроса. Применяют условия, определенные в ИСО/МЭК 14443-3:2011 (подраздел 5.1).

H.4.1.1 Оборудование

См. H.1.

H.4.1.2 Процедура

Поместить LT в область рабочего поля PCD и записать наличие и содержание команд PCD.

Во время следующей процедуры радиочастотные данные входа/приема должны быть непрерывно контролируемы и проверяемы на соответствие ИСО/МЭК 14443-2:2010. Все переходы сигнала (уровень и временные соотношения), а также логическое содержание коммуникации должны быть записаны.

Использовать следующую последовательность.

a) UT выполняет процедуру активации в соответствии с H.1.8.1.

b) LT ожидает, пока PCD пошлет ему действительный кадр команды REQA/WUPA и действительный кадр команды REQB/WUPB в любом порядке и повторяемости.

H.4.1.3 Протокол испытаний

Используя форму таблицы H.25, заполнить в ней соответствующую строку, учитывая результаты для обоих типов (типа A и типа B) согласно таблице H.14.

Таблица Н.14 — Критерии оценки результата испытания обработки цикла опроса

Объяснение	Результат испытания
Только если PCD посылает как минимум один раз кадр команды REQA/WUPA и как минимум один раз кадр команды REQB/WUPB (не менее 5 мс между каждым типом). Применяют условия по ИСО/МЭК 14443-3:2011 (подраздел 5.1)	PASS
Любой другой случай	FAIL

Н.4.2 Реакция PCD на запрос расширения времени ожидания

Цель данного испытания — определить поведение PCD, когда PICC использует запрос расширения времени ожидания [см. ИСО/МЭК 14443-4:2008, (подраздел 7.3)]. Механизм поддержки WTX со стороны PCD также испытывают.

Данное испытание выполняют для FWI, как минимум установленного в состояние значений 0, 1 и 14, с TR0 и TR1, установленными, чтобы указывать значение по умолчанию в LT, если он эмулирует PICC типа B.

Данную тестовую комбинацию выполняют для WTXM, как минимум установленного в состояние значений 1, 3 и 59 в соответствии с таблицей Н.15.

Таблица Н.15 — Минимальные комбинации

Комбинация	FWI	WTXM
1	0	1
2	0	3
3	0	59
4	1	1
5	1	3
6	1	59
7	14	1
8	14	3
9	14	59

Н.4.2.1 Оборудование

См. Н.1.

Н.4.2.2 Процедура

Поместить LT в рабочую область PCD и записать наличие и содержание команд PCD.

Во время следующей процедуры радиочастотные данные входа/приема должны быть непрерывно контролируемы и проверяемы на соответствие ИСО/МЭК 14443-2:2010. Все переходы сигнала (уровень и временные соотношения), а также логическое содержание коммуникации должны быть записаны.

UT выполняет процедуру активации протокола в соответствии с Н.1.8.2 для типа A или Н.1.8.3 для типа B.

Н.4.2.2.1 Процедура 1 [ИСО/МЭК 14443-4:2008 (подраздел 7.3)]

Использовать следующую последовательность сразу после процедуры по Н.4.2.2.

a) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

b) LT ожидает, пока PCD пошлет ему блок $I(0)_0$ с полем INF, содержащим UT_TEST_COMMAND1.

c) LT посылает запрос S(WTX) на расширение времени ожидания.

d) PCD должно послать ответ S(WTX) с INF(с b6 по b1) = используемый WTXM.

Сценарий Н.16 — Реакция PCD на запрос LT расширения времени ожидания, процедура 1

PCD		LT
I(0)0 (INF = UT_TEST_COMMAND1)	→	
	←	Запрос S(WTX)
Ответ S(WTX)	→	

H.4.2.2.1.1 Ожидаемый результат

Поведение PCD должно соответствовать ожидаемому по сценарию H.16 для всех значений FWI и WTXM, определенных в H.4.2 во всех тестовых комбинациях. Результаты данного испытания считают удовлетворительными, только если все выполненные тесты удовлетворяют предъявляемым требованиям.

H.4.2.2.1.2 Протокол испытаний

Используя форму таблицы H.25, заполнить в ней соответствующую строку, учитывая результаты для обоих типов (типа A и типа B) согласно таблице H.16.

Таблица H.16 — Критерии оценки результата испытания реакции PCD на запрос расширения времени ожидания (процедура 1)

Объяснение	Результат испытаний
Только если PCD посылает запрос S(WTX) с INF(от b6 до b1) = WTXM для всех значений FWI и WTXM, определенных в H.4.2 (минимальные комбинации для испытания приведены в таблице H.15)	PASS
Любой другой случай	FAIL

H.4.2.2.2 Процедура 2 [ИСО/МЭК 14443-4:2008 (подраздел 7.3)]

Использовать следующую последовательность сразу после процедуры по H.4.2.2.

a) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

b) LT ожидает, пока PCD пошлет ему I-блок с полем INF, содержащим UT_TEST_COMMAND1.

c) LT посылает запрос S(WTX).

d) PCD должно послать ответ S(WTX) с INF(от b6 до b1) = WTXM. Если он не соответствует ожидаемому ответу, то закончить испытание на данном шаге.

e) Установить следующие параметры временных соотношений бит на LT:

Параметр	Значение	Ссылка
Вспомогательное время ожидания кадра FWT_{TEMP}	$WTXM \times (256 \times 16/fc) \times 2^{FWI}$	ИСО/МЭК 14443-3:2011 (подпункт 7.9.4.3) ИСО/МЭК 14443-4:2008 (подразделы 7.2 и 7.3)
EGT по ИСО/МЭК 14443-3:2011	Максимальное (19 мкс)	ИСО/МЭК 14443-3:2011 (пункт 7.1.2)
Примечание — Время ответа кадра определяют как промежуток времени между задним фронтом EOF полученного кадра и передним фронтом SOF последующего посланного кадра.		

f) LT посылает ответ на команду UT_TEST_COMMAND1, посланную на шаге b).

g) PCD должно передать ответный UT_APDU (ответ на команду UT_TEST_COMMAND1) обратно UT.

h) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

i) LT ожидает, пока PCD пошлет ему I-блок с полем INF, содержащим UT_TEST_COMMAND1. PCD должно осуществить сброс FWT на этом шаге.

j) LT остается в состоянии Mute как минимум в течение ожидаемого FWT. Это означает тайм-аут FWT для PCD.

k) Записать наличие, содержание и временные соотношения ответа PCD. PCD должно послать блок R(NAK) только после того, как закончится FWT (ИСО/МЭК 14443-4:2008, 7.5.5.2[1]).

l) Измерить и записать промежуток времени между концом кадра PCD на шаге i) и началом кадра PCD на шаге k).

Сценарий H.17 — Реакция PCD на запрос LT расширения времени ожидания, процедура 2

PCD		LT
$I(0)_0$ (INF = UT_TEST_COMMAND1)	→	
	←	Запрос S(WTX) (WTXM принадлежит тестовому множеству)
Ответ S(WTX)	→	
	←	$I(0)_0$ (INF = ответ на UT_TEST_COMMAND1)
$I(0)_1$ (INF = UT_TEST_COMMAND1)	→	
	←	Mute
R(NAK)	→	

[1] В ИСО/МЭК 10373-6:2011 указан подпункт 7.5.4.2 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

H.4.2.2.2.1 Ожидаемый результат

Поведение PCD должно соответствовать ожидаемому по сценарию H.17 для всех значений FWI и WTXM, определенных в H.4.2 во всех тестовых комбинациях. Результаты данного испытания считают удовлетворительными, только если все выполненные тесты удовлетворяют предъявляемым требованиям.

H.4.2.2.2.2 Протокол испытаний

Используя форму таблицы H.25, заполнить в ней соответствующую строку для обоих типов (типа A и типа B) согласно таблице H.17.

Таблица H.17 — Критерии оценки результата испытания реакции PCD на запрос расширения времени ожидания (процедура 2)

Объяснение	Результат испытаний
Только если ответ на TEST_COMMAND_1 не был послан обратно UT или если PCD посылает R(NAK) до окончания FWT как минимум для одного значения FWI и одного значения WTXM, определенных в H.4.2 (минимальные комбинации для тестирования приведены в таблице H.15)	FAIL
Любой другой случай	PASS

H.4.3 Обнаружение и исправление ошибок

Цель данного испытания — определить соответствие поведения PCD, если происходят ошибки, требованиям ИСО/МЭК 14443-4:2008 (пункт 7.5.7[1]).

Примечание — В данном разделе ошибочной блок — это кадр с ошибочным CRC.

H.4.3.1 Оборудование

См. H.1.

H.4.3.2 Процедура

Поместить LT в область рабочего поля PCD и записать наличие и содержание команд PCD.

Во время следующей процедуры радиочастотные данные входа/приема должны быть непрерывно контролируемы и проверяемы на соответствие ИСО/МЭК 14443-2:2010. Все переходы сигнала (уровень и временные соотношения), а также логическое содержание коммуникации должны быть записаны.

UT выполняет процедуру активации протокола в соответствии с H.1.8.2 для типа A или H.1.8.3 для типа B.

H.4.3.2.1 Процедура 1 [ИСО/МЭК 14443-4:2008 (сценарий 12 приложения B)]

Использовать следующую последовательность сразу после процедуры по H.4.3.2.

a) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

b) LT ожидает, пока PCD пошлет ему I-блок с полем INF, содержащим UT_TEST_COMMAND1.

c) LT посылает PCD ошибочный I-блок.

d) PCD должно послать $R(NAK)_0$.

e) LT посылает PCD I-блок (содержащий какой-либо ответный UT_APDU с ответом на UT_TEST_COMMAND1).

f) PCD должно передать ответный UT_APDU обратно UT. Проверить на UT, что блок с этим ответным UT_APDU принят корректно.

Сценарий H.18 — Обнаружение и исправление PCD ошибок передачи [ИСО/МЭК 14443-4:2008 (сценарий 12 приложения B)] (процедура 1)

PCD		LT
$I(0)_0$ (INF = UT_TEST_COMMAND1)	→	
	←	$I(0)_0$ (INF = ответ на UT_TEST_COMMAND1, ошибочный CRC)
$R(NAK)_0$ ('BA' CID CRC или 'B2' CRC)	→	
	←	$I(0)_0$ (INF = ответ на UT_TEST_COMMAND1)

H.4.3.2.2 Процедура 2 [ИСО/МЭК 14443-4:2008 (подпункт 7.5.5.2[2], правило 4)]

Использовать следующую последовательность сразу после процедуры по H.4.3.2.

[1] В ИСО/МЭК 10373-6:2011 указан пункт 7.5.6 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

[2] В ИСО/МЭК 10373-6:2011 указан подпункт 7.5.4.2 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

a) UTпосылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).
b) LT ожидает, пока PCD пошлет ему I-блок с полем INF, содержащим UT_TEST_COMMAND1.
c) LT посылает PCD ошибочный блок.
d) PCD должно послать $R(NAK)_0$.
e) LT посылает PCD второй ошибочный блок.
f) PCD должно послать $R(NAK)_0$ или запрос S(DESELECT).

Сценарий Н.19 — Обнаружение и исправление PCD ошибок передачи (процедура 2)

PCD		LT
$I(0)_0$ (INF = UT_TEST_COMMAND1)	→	
	←	I(0)0 (INF = ответ на UT_TEST_COMMAND1, ошибочный CRC)
$R(NAK)_0$ ('BA' CID CRC или 'B2' CRC)	→	
	←	$I(0)_0$ (INF = ответ на UT_TEST_COMMAND1, ошибочный CRC)
$R(NAK)_0$ ('BA' CID CRC или 'B2' CRC) или S(DESELECT)	→	

Н.4.3.2.3 Процедура 3 [ИСО/МЭК 14443-4:2008 (подпункт 7.5.5.2[1]), правило 4)]
Использовать следующую последовательность сразу после процедуры по Н.4.3.2.
a) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).
b) LT ожидает, пока PCD пошлет ему I-блок с полем INF, содержащим UT_TEST_COMMAND1.
c) Поддерживать LT в состоянии Mute.
d) Записать все запросы от PCD. PCD должно послать $R(NAK)_0$ как минимум один раз.

Сценарий Н.20 — Обнаружение и исправление PCD ошибок передачи (процедура 3)

PCD		LT
$I(0)_0$ (INF = UT_TEST_COMMAND1)	→	
	←	Mute
$R(NAK)_0$ ('BA' CID CRC или 'B2' CRC)	→	
	←	Mute
$R(NAK)_0$ ('BA' CID CRC или 'B2' CRC) или S(DESELECT)	→	

Н.4.3.2.4 Процедура 4 [ИСО/МЭК 14443-4:2008 (подпункт 7.5.5.2[1], правило 7)]
Использовать следующую последовательность сразу после процедуры по Н.4.3.2.
a) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).
b) LT ожидает, пока PCD пошлет ему I-блок с полем INF, содержащим UT_TEST_COMMAND1.
c) LT посылает $R(ACK)_0$ (из-за нарушения LT правил протокола).
d) PCD должно послать запрос S(DESELECT).

Сценарий Н.21 — Обнаружение и исправление PCD ошибок передачи (процедура 4)

PCD		LT
$I(0)_0$ (INF = UT_TEST_COMMAND1)	→	
	←	$R(ACK)_0$ ('AA' CID CRC или 'A2' CRC)[a)]
Запрос S(DESELECT)	→	

[a)] Если блок PCD содержит CID, то используют левый вариант, в противном случае — правый вариант.

[1] В ИСО/МЭК 10373-6:2011 указан подпункт 7.5.4.2 без учета изменения нумерации пунктов, введенного изменением А1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

H.4.3.2.5 Процедура 5 (со сцеплением) [ИСО/МЭК 14443-4:2008 (подпункт 7.5.5.2[1], правило 5), ИСО/МЭК 14443-4:2008 (сценарий 23 приложения B)]

Использовать следующую последовательность сразу после процедуры по H.4.3.2.

a) UTпосылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

b) LT ожидает, пока PCD пошлет ему I-блок с полем INF, содержащим UT_TEST_COMMAND1.

c) LT посылает первый блок $I(1)_0$ цепочки и ожидает ответа PCD.

d) PCD должно послать $R(ACK)_1$.

e) LT посылает PCD ошибочный блок $I(0)_1$.

f) PCD должно послать $R(ACK)_1$.

Сценарий H.22 — Обнаружение и исправление PCD ошибок передачи (процедура 5) [ИСО/МЭК 14443-4:2008 (сценарий 19 приложения B)]

PCD		LT
$I(0)_0$ (INF = UT_TEST_COMMAND1)	→	
	←	$I(1)_0$ (INF = первая цепочка ответа на UT_TEST_COMMAND1)
$R(ACK)_1$ ('AB' CID CRC или 'A3' CRC)	→	
	←	$I(0)_1$ (INF = последняя цепочка ответа на UT_TEST_COMMAND1, ошибочный CRC)
$R(ACK)_1$	→	

H.4.3.2.6 Процедура 6 [ИСО/МЭК 14443-4:2008 (подпункт 7.5.5.2[1])]

Использовать следующую последовательность сразу после процедуры по H.4.3.2.

a) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

b) LT ожидает, пока PCD пошлет ему I-блок с полем INF, содержащим UT_TEST_COMMAND1.

c) LT посылает PCD ошибочный блок.

d) PCD должно послать $R(NAK)_0$.

e) LT остается в состоянии Mute.

f) Записать все ответы от PCD. PCD должно послать либо $R(NAK)_0$, либо запрос S(DESELECT).

Сценарий H.23 — Обнаружение и исправление PCD ошибок передачи (процедура 6)

PCD		LT
$I(0)_0$ (INF = UT_TEST_COMMAND1)	→	
	←	$I(0)_0$ (INF = ответ на UT_TEST_COMMAND1, ошибочный CRC)
$R(NAK)_0$ ('BA' CID CRC или 'B2' CRC)	→	
	←	Mute
$R(NAK)_0$ или запрос S(DESELECT)	→	

H.4.3.2.7 Процедура 7 [ИСО/МЭК 14443-4:2008 (пункт 7.5.7[2]; сценарий 14 приложения B)]

Использовать следующую последовательность сразу после процедуры по H.4.3.2.

a) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).

b) LT ожидает, пока PCD пошлет ему I-блок с полем INF, содержащим UT_TEST_COMMAND1.

c) LT посылает ошибочный запрос S(WTX).

d) PCD должно послать $R(NAK)_0$.

e) LT посылает действительный запрос S(WTX).

f) PCD должно послать ответ S(WTX).

g) LT посылает PCD I-блок (содержащий какой-либо ответный UT_APDU с ответом на UT_TEST_COMMAND1).

[1] В ИСО/МЭК 10373-6:2011 указан подпункт 7.5.4.2 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

[2] В ИСО/МЭК 10373-6:2011 указан пункт 7.5.6 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

h) PCD должно передать этот ответный UT_APDU обратно UT. Проверить на UT, что блок с этим ответным UT_APDU принят корректно.

Сценарий Н.24 — Обнаружение и исправление PCD ошибок передачи (процедура 7) [ИСО/МЭК 14443-4:2008 (сценарий 14 приложения В)]

PCD		LT
$I(0)_0$ (INF = UT_TEST_COMMAND1)	→	
	←	Запрос S(WTX), ошибочный CRC
$R(NAK)_0$ ('BA' CID CRC или 'B2' CRC)	→	
	←	Запрос S(WTX)
Ответ S(WTX)	→	
	←	$I(0)_0$ (INF = ответ на UT_TEST_COMMAND1)

Н.4.3.2.8 Процедура 8 [ИСО/МЭК 14443-4:2008 (пункт 7.5.7[1], сценарий 17 приложения В)]
Использовать следующую последовательность сразу после процедуры по Н.4.3.2.
a) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).
b) LT ожидает, пока PCD пошлет ему I-блок с полем INF, содержащим UT_TEST_COMMAND1.
c) LT посылает запрос S(WTX).
d) PCD должно отправить ответ S(WTX).
e) LT посылает ошибочный $I(0)_0$.
f) PCD должно послать $R(NAK)_0$.
g) LT посылает действительный $I(0)_0$ с максимальным временным интервалом между $R(NAK)_0$ и $I(0)_0$, чтобы проверить, что FWT PCD все еще расширенный.
h) PCD должно передать этот ответный UT_APDU обратно UT. Проверить на UT, что блок с этим ответным UT_APDU принят корректно.

Сценарий Н.25 — Обнаружение и исправление PCD ошибок передачи (процедура 8) [ИСО/МЭК 14443-4:2008 (сценарий 17 приложения В)]

PCD		LT
$I(0)_0$ (INF = UT_TEST_COMMAND1)	→	
	←	Запрос S(WTX)
Ответ S(WTX)	→	
	←	$I(0)_0$ (INF = ответ на UT_TEST_COMMAND1, ошибочный CRC)
$R(NAK)_0$ ('BA' CID CRC или 'B2' CRC)	→	
	←	$I(0)_0$ (INF = ответ на UT_TEST_COMMAND1)

Н.4.3.2.9 Процедура 9 (со сцеплением) [см. ИСО/МЭК 14443-4:2008 (сценарий 20 приложения В)]
Использовать следующую последовательность сразу после процедуры по Н.4.3.2.
a) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND2 (вызывающую три цепочки)).
b) LT ожидает, пока PCD пошлет ему I-блок $I(1)_0$ с первой цепочкой UT_TEST_COMMAND2 в поле INF.
c) LT посылает ошибочный $R(ACK)_0$.
d) PCD должно послать $R(NAK)_0$.
e) LT посылает $R(ACK)_0$.
f) PCD должно послать следующий блок $I(1)_1$ цепочки.
g) LT посылает $R(ACK)_1$.
h) PCD должно послать последний блок $I(0)_0$ цепочки.
i) LT посылает I-блок (содержащий какой-либо ответный UT_APDU с ответом на UT_TEST_COMMAND2).
j) PCD должно передать этот ответный UT_APDU обратно UT. Проверить на UT, что блок с этим ответным UT_APDU принят корректно.

Примечание — В случае если UT не может принудить PCD использовать три цепочки для посылки TEST_COMMAND2, необходимо изменить ожидаемую процедуру, чтобы отразить цель испытания, которая заключается в том, чтобы убедиться, что нумерация блоков и образование цепочек происходят правильно после ошибки передачи на шаге c).

[1] В ИСО/МЭК 10373-6:2011 указан пункт 7.5.6 без учета изменения нумерации пунктов, введенного изменением А1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

Сценарий Н.26 — Обнаружение и исправление PCD ошибок передачи (процедура 9) (со сцеплением) [ИСО/МЭК 14443-4:2008 (сценарий 20 приложения В)]

PCD		LT
$I(1)_0$ (INF = первая цепочка UT_TEST_COMMAND2)	→	
	←	$R(ACK)_0$, ошибочный CRC ('AA' CID '00 00' или 'A2 00 00')[a]
$R(NAK)_0$ ('BA' CID CRC или 'B2' CRC)	→	
	←	$R(ACK)_0$ ('AA' CID CRC или 'A2' CRC)[a]
$I(1)_1$ (INF = вторая цепочка UT_TEST_COMMAND2)	→	
	←	$R(ACK)_1$ ('AB' CID CRC или 'A3' CRC)[a]
$I(0)_0$ (INF = последняя цепочка UT_TEST_COMMAND2)	→	
	←	$I(0)_0$ (INF = ответ на UT_TEST_COMMAND2)
[a] Если кадр PCD содержит CID, то используют левый вариант, в противном случае — правый вариант.		

Н.4.3.2.10 Процедура 10 (со сцеплением) [см. ИСО/МЭК 14443-4:2008 (сценарий 21 приложения В)]
Использовать следующую последовательность сразу после процедуры по Н.4.3.2.
a) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND2 (вызывающую три цепочки)).
b) LT ожидает, пока PCD пошлет ему I-блок $I(1)_0$ с первой цепочкой UT_TEST_COMMAND2 в поле INF.
c) LT остается в состоянии Mute.
d) PCD должно послать $R(NAK)_0$.
e) LT посылает R-блок с несинхронизированным последовательным номером блока $R(ACK)_1$.
f) PCD должно повторить для LT предыдущий I-блок $I(1)_0$.
g) LT посылает $R(ACK)_0$.
h) PCD должно послать следующий блок $I(1)_1$ цепочки.
i) LT посылает $R(ACK)_1$.
j) PCD должно послать последний блок $I(0)_0$ цепочки.
k) LT посылает PCD I-блок (содержащий какой-либо ответный UT_APDU с ответом на UT_TEST_COMMAND2).
l) PCD должно передать этот ответный UT_APDU обратно UT. Проверить на UT, что блок с этим ответным UT_APDU принят корректно.

Примечание — В случае если UT не может принудить PCD использовать три цепочки для посылки TEST_COMMAND2, необходимо изменить ожидаемую процедуру, чтобы отразить цель испытания, которая заключается в том, чтобы убедиться, что нумерация блоков и сцепление происходят правильно после ошибки передачи на шаге c).

Сценарий Н.27 — Обнаружение и исправление PCD ошибок передачи (процедура 10) [см. ИСО/МЭК 14443-4:2008 (сценарий 21 приложения В)]

PCD		LT
$I(1)_0$(INF = первая цепочка UT_TEST_COMMAND2)	→	
	←	$R(ACK)_0$ ('AA' CID CRC или 'A2' CRC)[a]
$I(1)_1$ (INF = вторая цепочка UT_TEST_COMMAND2)	→	
	←	Mute
$R(NAK)_1$ ('BB' CID CRC или 'B3' CRC)	→	
	←	$R(ACK)_0$ ('AA' CID CRC или 'A2' CRC)[a]
$I(1)_1$ (INF = вторая цепочка UT_TEST_COMMAND2)	→	
	←	$R(ACK)_1$ ('AB' CID CRC или 'A3' CRC)[a]
$I(0)_0$ (INF = последняя цепочка UT_TEST_COMMAND2)	→	
	←	$I(0)_0$ (INF = ответ на UT_TEST_COMMAND2)
[a] Если кадр PCD содержит CID, то используют левый вариант, в противном случае — правый вариант.		

H.4.3.2.11 Процедура 11 (со сцеплением) [см. ИСО/МЭК 14443-4:2008 (сценарий 24 приложения B)]
Использовать следующую последовательность сразу после процедуры по H.4.3.2.
a) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).
b) LT ожидает, пока PCD пошлет ему I-блок $I(0)_0$ с полем INF, содержащим UT_TEST_COMMAND1.
c) LT посылает I-блок $I(1)_0$, указывающий сцепление.
d) PCD должно послать $R(ACK)_1$.
e) LT посылает PCD ошибочный I-блок $I(1)_1$.
f) PCD должно послать $R(ACK)_1$.
g) LT повторно посылает I-блок $I(1)_1$ без ошибки.
h) PCD должно послать $R(ACK)_0$.
i) LT посылает PCD последний блок цепочки в I-блоке $I(0)_0$ (его ответного UT_APDU с ответом на UT_TEST_COMMAND1).
j) PCD должно передать ответный UT_APDU, содержащий все сцепленные сегменты, обратно UT. Проверить на UT, что блок с этим ответным UT_APDU принят корректно.

Сценарий H.28 — Обнаружение и исправление PCD ошибок передачи (процедура 11) [ИСО/МЭК 14443-4:2008 (сценарий 24 приложения B)]

PCD		LT
$I(0)_0$ (INF = UT_TEST_COMMAND1)	→	
	←	$I(1)_0$ (INF = первая цепочка ответа на UT_TEST_COMMAND1)
$R(ACK)_1$ ('AB' CID CRC или 'A3' CRC)	→	
	←	$I(1)_1$ (INF = вторая цепочка ответа на UT_TEST_COMMAND1, ошибочный CRC)
$R(ACK)_1$ ('AB' CID CRC или 'A3' CRC)	→	
	←	$I(1)_1$ (INF = вторая цепочка ответа на UT_TEST_COMMAND1)
$R(ACK)_0$ ('AA' CID CRC или 'A2' CRC)	→	
	←	$I(0)_0$ (INF = последняя цепочка ответа на UT_TEST_COMMAND1)

H.4.3.2.12 Процедура 12 [ИСО/МЭК 14443-4:2008 (подпункт 7.5.5.2[1], правило 8)]
Использовать следующую последовательность сразу после процедуры по H.4.3.2.
a) UT посылает PCD SEND_UT_APDU(UT_TEST_COMMAND1).
b) LT ожидает, пока PCD пошлет ему I-блок с полем INF, содержащим UT_TEST_COMMAND1.
c) LT остается в состоянии Mute.
d) LT ожидает, пока PCD пошлет ему $R(NAK)_0$ (R-блок может быть послан более одного раза).
e) LT остается в состоянии Mute.
f) LT ожидает, пока PCD пошлет ему запрос S(DESELECT).
g) LT остается в состоянии Mute.
h) Записать ответ от PCD. PCD должно либо повторно передать запрос S(DESELECT), либо игнорировать LT.

Примечание — В рамках этого сценария новое правило 8 проверяют только для блоков S(DESELECT).

(Введено дополнительно. Изм. A3:2012.)

Сценарий H.29 — Обнаружение и исправление PCD ошибок передачи (процедура 12)

PCD		LT
$I(0)_0$(INF = UT_TEST_COMMAND1)	→	
	←	Mute
$R(NAK)_0$ ('BA' CID CRC или 'B2' CRC)	→	
	←	Mute
Запрос S(DESELECT) ('CA' CID CRC или 'C2' CRC)	→	
	←	Mute
Ответ PCD	→	

[1] В ИСО/МЭК 10373-6:2011 указан подпункт 7.5.4.2 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

H.4.3.3 Протокол испытаний

Используя форму таблицы H.25, заполнить в ней соответствующую строку для обоих типов (типа A и типа B) согласно таблице H.18.

Таблица H.18 — Критерии оценки результата испытания обнаружения и исправления ошибок

Объяснение	Результат испытаний
Только если поведение PCD совпадает с ожидаемым по сценарию для каждой процедуры	PASS
Любой другой случай	FAIL

H.4.4 Обработка NAD во время сцепления

Цель данного испытания — подтвердить, что PCD поддерживает NAD надлежащим образом.

H.4.4.1 Оборудование

См. H.1.

H.4.4.2 Процедура

Поместить LT в рабочую область PCD и записать наличие и содержание команд PCD.

Использовать следующую последовательность.

a) Конфигурировать LT как поддерживающую NAD.

b) Повторить процедуру из сценария H.26.

H.4.4.3 Протокол испытаний

Используя форму таблицы H.25, заполнить в ней соответствующую строку для обоих типов (типа A и типа B) согласно таблице H.19.

Таблица H.19 — Критерии оценки результата испытания обработки NAD во время сцепления

Объяснение	Результат испытаний
Только если PCD использует NAD только в первом пакете сцепления или не использует NAD	PASS
Любой другой случай	FAIL

H.5 Непрерывный мониторинг пакетов, посылаемых PCD

Цель данного испытания — подтвердить, что PCD не устанавливает никакие RFU-биты в любом посылаемом кадре в состояние значения, отличного от значения по умолчанию, предписанного для RFU-бит официальными документами. Более того, испытание должно гарантировать, что поле никогда не принимает RFU-значения. Также испытание должно подтвердить, что R-блок и S-блок соответствуют определениям протокола, и что правила, касающиеся первого байта пакета, не нарушены.

H.5.1 RFU-поля

RFU-поля следует непрерывно контролировать во время испытания и всегда проверять на содержание присвоенного значения по умолчанию. Результаты испытания следует считать неудовлетворительными, а испытуемое PCD признавать несоответствующим предъявляемым требованиям, если в произвольное время RFU-поле не установлено в состояние соответствующего значения по умолчанию.

H.5.2 RFU-значения

Функциональные поля следует непрерывно контролировать во время испытания и всегда проверять на содержание только стандартных или проприетарных функциональных значений, установленных в стандартах. Результаты испытания считают неудовлетворительными, а испытуемое PCD должно быть признано не соответствующим требованиям, если в произвольное время функциональное поле не установлено в состояние вышеуказанных значений.

H.5.3 R-блок

R-блок никогда не должен содержать поле INF [см. ИСО/МЭК 14443-4:2008 (подпункт 7.1.1.1)].

H.5.4 S-блок

S-блок должен иметь:

- поле INF только из одного байта, если он представляет собой блок WTX, либо
- поле INF из n байтов ($n \geq 0$), если он представляет собой блок PARAMRTERS, либо
- он не должен иметь поле INF в остальных случаях.

[См. ИСО/МЭК 14443-4:2008 (подпункт 7.1.1.1) и ИСО/МЭК 14443-4:2008/Изм. A1]

(Измененная редакция. Изм. A3:2012.)

H.5.5 PCB

Байт PCB должен содержать допустимые значения [см. ИСО/МЭК 14443-4:2008 (подпункт 7.1.1.1) и ИСО/МЭК 14443-4:2008 (приложение C)].

H.5.6 Кадры инициализации типа A

Кадры инициализации типа A должны содержать допустимые значения [см. ИСО/МЭК 14443-3:2011 (пункт 6.4.1 и подпункт 6.5.3.2)].

H.5.7 Оборудование

См. H.1.

H.5.8 Процедура

Во время испытательных процедур и сценариев всегда следует записывать логическое содержание коммуникации.

H.5.9 Протокол испытаний

Используя формы таблиц H.25 и H.27, заполнить в них соответствующие строки для типа A и типа B согласно таблице H.20.

Таблица H.20 — Критерии оценки результатов непрерывного мониторинга пакетов, посылаемых PCD

Объяснение	Результат испытаний
Если PCD удовлетворяет всем следующим условиям: - PCD устанавливает значения по умолчанию во RFU-битах во всех посылаемых кадрах; - PCD не устанавливает RFU-значение для какого-либо поля; - PCD не нарушает правила длины для R-блока и S-блока; - PCD не нарушает правила кодирования первого байта блока и кадра [см. ИСО/МЭК 14443-4:2008 (приложение C)]	PASS
Если PCD удовлетворяет как минимум одному из следующих условий: - PCD устанавливает значение, отличное от значения по умолчанию, как минимум в одном RFU-бите в каком-либо посылаемом кадре; - PCD устанавливает какое-либо поле в состояние RFU-значения; - PCD нарушает правила длины для R-блока и S-блока; - PCD нарушает правила кодирования первого байта блока и кадра [см. ИСО/МЭК 14443-4:2008 (приложение C)]	FAIL

H.6 Результаты, включаемые в протокол испытаний

Таблица H.21 — Таблица временных соотношений для типа A

№	Параметр	Ссылка на стандарт	Эталонное значение	Измеренное значение
1	Время задержки кадра при передаче от PICC к PCD	ИСО/МЭК 14443-3:2011 (подпункт 6.2.1.2)	Не менее 1175/*fc* (~86 мкс)	
2	Разграничительный интервал запроса	ИСО/МЭК 14443-3:2011 (пункт 6.2.2)	Не менее 7000/*fc* (~512 мкс)	

Примечание — Все временные значения вычисляют для несущей частоты *fc* = 13,56 МГц и скорости передачи *fc*/128 (~106 кбит/с).

Таблица H.22 — Таблица временных соотношений для типа B

№	Параметр	Ссылка на стандарт	Минимальное значение	Максимальное значение	Измеренное значение
1	Низкий SOF	ИСО/МЭК 14443-3:2011 (пункт 7.1.4)	10 etu (~94,40 мкс)	11 etu (~103,83 мкс)	
2	Низкий EOF	ИСО/МЭК 14443-3:2011 (пункт 7.1.5)	10 etu (~94,40 мкс)	11 etu (~103,83 мкс)	
3	Границы бит	ИСО/МЭК 14443-3:2011 (пункт 7.1.1)	(n – 0,125) etu	(n + 0,125) etu	
4	EGT от PCD к PICC	ИСО/МЭК 14443-3:2011 (пункт 7.1.2)	0 мкс	57 мкс	
5	Минимальная задержка между стартом EOF PICC и стартом SOF PCD	ИСО/МЭК 14443-3:2011 (пункт 7.1.7)	10 etu + 32/*fs*	Отсутствует	

Примечание — Все временные значения вычисляют для несущей частоты *fc* = 13,56 МГц и скорости передачи *fc*/128 (~106 кбит/с).

Таблица H.23 — Результаты для методов испытаний, применяемых для типа A

Метод испытаний по ИСО/МЭК 10373-6		Номер сценария ИСО/МЭК 10373-6	Результат испытания: PASS или FAIL или N/A[a)]
Подраздел	Параметр		
H.2.1	Время задержки кадра при передаче от PICC к PCD		
H.2.2	Разграничительный интервал запроса		
H.2.3	Обработка коллизии бит во время ATQA		
H.2.4	Обработка цикла антиколлизии	Сценарий H.1	
		Сценарий H.2	
		Сценарий H.3	
		Сценарий H.4	
H.2.5	Обработка RATS и ATS	Сценарий H.5	
		Сценарий H.6	
		Сценарий H.7	
H.2.6	Обработка PPS-ответа	Сценарий H.8	
		Сценарий H.9	
H.2.7	Механизм выбора размера кадра	Сценарий H.10	
H.2.8	Обработка разграничительного интервала кадра при запуске	Сценарий H.11	
H.2.9	Обработка CID во время активации со стороны PCD	Сценарий H.12	

[a)] Если испытание состоит из несколько процедур, то PASS указывают только в том случае, когда каждая отдельная процедура выдает PASS.

Таблица H.24 — Результаты для методов испытаний, применяемых для типа B

Метод испытаний по ИСО/МЭК 10373-6		Номера сценария ИСО/МЭК 10373-6	Результат испытания: PASS или FAIL или N/A[a)]
Подраздел	Параметр		
H.3.1	Временные соотношения передачи на I/O		
H.3.2	Механизм выбора размера кадра	Сценарий H.13	
H.3.3	Обработка CID во время активации со стороны PCD	Сценарий H.14	
		Сценарий H.15	

[a)] Если испытание состоит из несколько процедур, то PASS указывают только в том случае, когда каждая отдельная процедура выдает PASS.

Таблица H.25 — Результаты для методов испытаний логических операций PCD

Метод испытания по ИСО/МЭК 10373-6		Номер сценария		Результат испытания	
Раздел/ подраздел	Параметр	ИСО/МЭК 10373-6	ИСО/МЭК 14443-4:2008 (приложение B)	Тип A[a)]	Тип B[a)]
H.4.1	Обработка цикла опроса				
H.4.2	Реакция PCD на запрос расширения времени ожидания	Сценарий H.16			
		Сценарий H.17			
H.4.3	Обнаружение и исправление ошибок	Сценарий H.18	Сценарий 8 «Обмен I-блоками»		
		Сценарий H.19			
		Сценарий H.20			
		Сценарий H.21			
		Сценарий H.22	Сценарий 19 «PICC использует сцепление»		
		Сценарий H.23			
		Сценарий H.24	Сценарий 14 «Запрос расширения времени ожидания»		
		Сценарий H.25	Сценарий 17 «Запрос расширения времени ожидания»		
		Сценарий H.26	Сценарий 20 «PCD использует сцепление»		
		Сценарий H.27	Сценарий 21 «PCD использует сцепление»		
		Сценарий H.28	Сценарий 24 «PICC использует сцепление»		
		Сценарий H.29			
H.4.4	Обработка NAD во время сцепления				
H.5	Непрерывный мониторинг пакетов, посылаемых PCD				

[a)] Если испытание состоит из несколько процедур, то PASS указывают только в том случае, когда каждая отдельная процедура выдает PASS.

Таблица H.26 — Отчет по тестовому покрытию

№	Параметр	Описание	Информация
1	Сцепление	Испытывают, только если есть команда, поддерживающая более 16 байтов	
2	Обработка NAD		

Таблица Н.27 — Таблица отчета по RFU-полям/RFU-значениям PCD

Наименование	Команда PCD	RFU-поле/ RFU-значение	Значение		Результат испытания: PASS или FAIL или Not Done
			По умолчанию	Неразрешенное	
Короткий кадр типа А	REQA/WUPA	RFU-значения		Все значения, определенные как RFU в ИСО/МЭК 14443-3:2011 (таблица 3)[1]	
Кодирование SEL	SEL	RFU-значения		Все значения, определенные как RFU в ИСО/МЭК 14443-3:2011 (таблица 7)[1]	
AFI	REQB/WUPB	RFU-значения		Все значения, определенные как RFU в ИСО/МЭК 14443-3:2011 (таблица 22)[1]	
PARAM	REQB/WUPB	RFU-поле (от b8 до b6)	(000)b	Все остальные значения	
		RFU-значения в номере слотов (от b3 до b1)		(101)b (110)b (111)b	
Param 1	ATTRIB	RFU-поле (от b2 до b1)	(00)b	Все остальные значения	
Минимальный TR0	ATTRIB	RFU-значения (от b8 до b7)		(11)b	
Минимальный TR1	ATTRIB	RFU-значения (от b6 до b5)		(11)b	
Param 2	ATTRIB	RFU-значения (от b4 до b1)		Все значения от '9'((1001) b) до 'F'((1111)b)	
Param 3	ATTRIB	RFU-поле (от b8 дob4)	(00000)b	Все остальные значения	
Param 4	ATTRIB	RFU-поле (от b8 дob5)	(0000)b	Все остальные значения	
		RFU-значения (от b4 до b1)		'F'((1111)b)	

[1] В ИСО/МЭК 10373-6:2011 в этом месте ошибочно указан ИСО/МЭК 14443-3:2010.

Приложение I
(обязательное)

Методы испытания PCD для выбора высокой скорости передачи

I.1 Оборудование

В данном испытании испытательное оборудование для PCD должно быть конфигурируемым, чтобы можно было изменять скорость передачи во время выполнения процедуры испытания. Тестер должен быть способен измерять скорость передачи, используемую PCD, на каждой стадии процедуры испытания.

I.2 Процедура

Поместить испытательное оборудование в область поля PCD.

I.2.1 Процедура для типа A

Следующую процедуру необходимо повторить для всех значений байта интерфейса TA(1), определенных в таблице I.1.

a) Выполнить последовательность активации, как определено в ИСО/МЭК 14443-3:2011.

b) PCD должно послать команду RATS, как определено в ИСО/МЭК 14443-4:2008.

c) Испытательное оборудование отвечает действительным ATS, включающим в себя TA(1) в соответствии с таблицей I.1.

d) PCD может дополнительно послать PPS с байтом PPS1, имеющим допустимое значение параметра в соответствии с таблицей I.1.

e) Если PCD послало PPS, то испытательное оборудование подтверждает прием полученного PPS действительным ответом PPS.

f) PCD должно послать блок $I(0)_0$, используя выбранную скорость передачи.

Примечание 1 — Этим блоком может быть также $I(1)_0$ или R(NAK) в случае использования метода 2a для проверки наличия PICC, как определено в ИСО/МЭК 14443-4:2008 (подпункт 7.5.6.2[1]).

g) Испытательное оборудование посылает действительный ответ, используя выбранную скорость передачи. Следует проверить, приняло ли PCD ответ испытательного оборудования.

Примечание 2 — Следующие шаги могут быть не применимы, если PCD встроенно в изделие.

h) PCD должно послать запрос S(DESELECT), используя выбранную скорость передачи.

i) Испытательное оборудование посылает действительный ответ S(DESELECT), используя выбранную скорость передачи. Следует проверить, приняло ли PCD ответ испытательного оборудования.

j) PCD должно послать действительный кадр команды REQA, используя скорость передачи *fc*/128.

k) Испытательное оборудование отвечает действительным ATQA.

Таблица I.1 — Корректное поведение PCD после ATS с TA(1)

TA(1)	Допустимые значения параметра для PPS1
(10000000)b	(00000000)b[a)]
(10010001)b	(00000101)b, (00000000)b
(10100010)b	(00001010)b, (00000000)b
(10110011)b	(00000101)b, (00001010)b, (00000000)b
(11000100)b	(00001111)b, (00000000)b
(11010101)b	(00000101)b, (00001111)b, (00000000)b
(11100110)b	(00001010)b, (00001111)b, (00000000)b
(11110111)b	(00000101)b, (00001010)b, (00001111)b, (00000000)b
(00000000)b	(00000000)b[a)]

[1] В ИСО/МЭК 10373-6:2011 указан подпункт 7.5.5.2 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

Продолжение таблицы I.1

TA(1)	Допустимые значения параметра для PPS1
(00000001)b	(00000001)b, (00000000)b
(00000010)b	(00000010)b, (00000000)b
(00000011)b	(00000001)b, (00000010)b, (00000000)b
(00000100)b	(00000011)b, (00000000)b
(00000101)b	(00000001)b, (00000011)b, (00000000)b
(00000110)b	(00000010)b, (00000011)b, (00000000)b
(00000111)b	(00000001)b, (00000010)b, (00000011)b, (00000000)b
(00010000)b	(00000000)b (00000100)b
(00010001)b	(00000001)b, (00000000)b (00000101)b, (00000100)b
(00010010)b	(00000010)b, (00000000)b (00000110)b, (00000100)b
(00010011)b	(00000001)b, (00000010)b, (00000000)b (00000101)b, (00000110)b, (00000100)b
(00010100)b	(00000011)b, (00000000)b (00000111)b, (00000100)b
(00010101)b	(00000001)b, (00000011)b, (00000000)b (00000101)b, (00000111)b, (00000100)b
(00010110)b	(00000010)b, (00000011)b, (00000000)b (00000110)b, (00000111)b, (00000100)b
(00010111)b	(00000001)b, (00000010)b, (00000011)b, (00000000)b (00000101)b, (00000110)b, (00000111)b, (00000100)b
(00100000)b	(00000000)b (00001000)b
(00100001)b	(00000001)b, (00000000)b (00001001)b, (00001000)b
(00100010)b	(00000010)b, (00000000)b (00001010)b, (00001000)b
(00100011)b	(00000001)b, (00000010)b, (00000000)b (00001001)b, (00001010)b, (00001000)b
(00100100)b	(00000011)b, (00000000)b (00001011)b, (00001000)b
(00100101)b	(00000001)b, (00000011)b, (00000000)b (00001001)b, (00001011)b, (00001000)b
(00100110)b	(00000010)b, (00000011)b, (00000000)b (00001010)b, (00001011)b, (00001000)b
(00100111)b	(00000001)b, (00000010)b, (00000011)b, (00000000)b (00001001)b, (00001010)b, (00001011)b, (00001000)b
(00110000)b	(00000000)b (00000100)b (00001000)b
(00110001)b	(00000001)b, (00000000)b (00000101)b, (00000100)b (00001001)b, (00001000)b

Продолжение таблицы I.1

TA(1)	Допустимые значения параметра для PPS1
(00110010)b	(00000010)b, (00000000)b (00000110)b, (00000100)b (00001010)b, (00001000)b
(00110011)b	(00000001)b, (00000010)b, (00000000)b (00000101)b, (00000110)b, (00000100)b (00001001)b, (00001010)b, (00001000)b
(00110100)b	(00000011)b, (00000000)b (00000111)b, (00000100)b (00001011)b, (00001000)b
(00110101)b	(00000001)b, (00000011)b, (00000000)b (00000101)b, (00000111)b, (00000100)b (00001001)b, (00001011)b, (00001000)b
(00110110)b	(00000010)b, (00000011)b, (00000000)b (00000110)b, (00000111)b, (00000100)b (00001010)b, (00001011)b, (00001000)b
(00110111)b	(00000001)b, (00000010)b, (00000011)b, (00000000)b (00000101)b, (00000110)b, (00000111)b, (00000100)b (00001001)b, (00001010)b, (00001011)b, (00001000)b
(01000000)b	(00000000)b (00001100)b
(01000001)b	(00000001)b, (00000000)b (00001101)b, (00001100)b
(01000010)b	(00000010)b, (00000000)b (00001110)b, (00001100)b
(01000011)b	(00000001)b, (00000010)b, (00000000)b (00001101)b, (00001110)b, (00001100)b
(01000100)b	(00000011)b, (00000000)b (00001111)b, (00001100)b
(01000101)b	(00000001)b, (00000011)b, (00000000)b (00001101)b, (00001111)b, (00001100)b
(01000110)b	(00000010)b, (00000011)b, (00000000)b (00001110)b, (00001111)b, (00001100)b
(01000111)b	(00000001)b, (00000010)b, (00000011)b, (00000000)b (00001101)b, (00001110)b, (00001111)b, (00001100)b
(01010000)b	(00000000)b (00000100)b (00001100)b
(01010001)b	(00000001)b, (00000000)b (00000101)b, (00000100)b (00001101)b, (00001100)b
(01010010)b	(00000010)b, (00000000)b (00000110)b, (00000100)b (00001110)b, (00001100)b
(01010011)b	(00000001)b, (00000010)b, (00000000)b (00000101)b, (00000110)b, (00000100)b (00001101)b, (00001110)b, (00001100)b

Продолжение таблицы I.1

TA(1)	Допустимые значения параметра для PPS1
(01010100)b	(00000011)b, (00000000)b (00000111)b, (00000100)b (00001111)b, (00001100)b
(01010101)b	(00000001)b, (00000011)b, (00000000)b (00000101)b, (00000111)b, (00000100)b (00001101)b, (00001111)b, (00001100)b
(01010110)b	(00000010)b, (00000011)b, (00000000)b (00000110)b, (00000111)b, (00000100)b (00001110)b, (00001111)b, (00001100)b
(01010111)b	(00000001)b, (00000010)b, (00000011)b, (00000000)b (00000101)b, (00000110)b, (00000111)b, (00000100)b (00001101)b, (00001110)b, (00001111)b, (00001100)b
(01100000)b	(00000000)b (00001000)b (00001100)b
(01100001)b	(00000001)b, (00000000)b (00001001)b, (00001000)b (00001101)b, (00001100)b
(01100010)b	(00000010)b, (00000000)b (00001010)b, (00001000)b (00001110)b, (00001100)b
(01100011)b	(00000001)b, (00000010)b, (00000000)b (00001001)b, (00001010)b, (00001000)b (00001101)b, (00001110)b, (00001100)b
(01100100)b	(00000011)b, (00000000)b (00001011)b, (00001000)b (00001111)b, (00001100)b
(01100101)b	(00000001)b, (00000011)b, (00000000)b (00001001)b, (00001011)b, (00001000)b (00001101)b, (00001111)b, (00001100)b
(01100110)b	(00000010)b, (00000011)b, (00000000)b (00001010)b, (00001011)b, (00001000)b (00001110)b, (00001111)b, (00001100)b
(01100111)b	(00000001)b, (00000010)b, (00000011)b, (00000000)b (00001001)b, (00001010)b, (00001011)b, (00001000)b (00001101)b, (00001110)b, (00001111)b, (00001100)b
(01110000)b	(00000000)b (00000100)b (00001000)b (00001100)b
(01110001)b	(00000001)b, (00000000)b (00000101)b, (00000100)b (00001001)b, (00001000)b (00001101)b, (00001100)b
(01110010)b	(00000010)b, (00000000)b (00000110)b, (00000100)b (00001010)b, (00001000)b (00001110)b, (00001100)b

Окончание таблицы I.1

TA(1)	Допустимые значения параметра для PPS1
(01110011)b	(00000001)b, (00000010)b, (00000000)b (00000101)b, (00000110)b, (00000100)b (00001001)b, (00001010)b, (00001000)b (00001101)b, (00001110)b, (00001100)b
(01110100)b	(00000011)b, (00000000)b (00000111)b, (00000100)b (00001011)b, (00001000)b (00001111)b, (00001100)b
(01110101)b	(00000001)b, (00000011)b, (00000000)b (00000101)b, (00000111)b, (00000100)b (00001001)b, (00001011)b, (00001000)b (00001101)b, (00001111)b, (00001100)b
(01110110)b	(00000010)b, (00000011)b, (00000000)b (00000110)b, (00000111)b, (00000100)b (00001010)b, (00001011)b, (00001000)b (00001110)b, (00001111)b, (00001100)b
(01110111)b	(00000001)b, (00000010)b, (00000011)b, (00000000)b (00000101)b, (00000110)b, (00000111)b, (00000100)b (00001001)b, (00001010)b, (00001011)b, (00001000)b (00001101)b, (00001110)b, (00001111)b, (00001100)b
[a] Команда PPS в этом случае бесполезна и может быть не поддерживаема PICC.	

Сценарий I.1 — Выбор высокой скорости передачи (тип A, процедура 1)

PCD		Оборудование для испытаний PCD
Кадр команды RATS ('E0 01' CRC_A)	→	
	←	ATS, TA(1) в соответствии с таблицей I.1
Дополнительный PPS-запрос в соответствии с таблицей I.1 «Корректное поведение PCD после ATS с TA(1)»	→	
	←	PPS-ответ (с использованием скорости передачи *fc*/128)
$I(0)_0$ (с использованием выбранной скорости передачи)	→	
	←	$I(0)_0$ (с использованием выбранной скорости передачи)
Запрос S(DESELECT)	→	
	←	Ответ S(DESELECT) (с использованием выбранной скорости передачи)
WUPA (с использованием скорости передачи *fc*/128)	→	
	←	ATQA (с использованием скорости передачи *fc*/128)

I.2.1.1 Ожидаемый результат

PCD должно вести себя, как описано в сценарии I.1, в каждом из 72 случаев испытания.

I.2.1.2 Протокол испытаний

Если PCD ведет себя допустимым образом в соответствии со сценарием I.1 в каждом из 72 случаев испытания, то результаты данного испытания считают удовлетворительными. В протоколе испытаний должны быть указаны скорости передачи, выбранные PCD в каждом из 72 случаев испытания.

I.2.2 Процедура для типа B

Следующую процедуру необходимо повторить для всех значений информационного байта протокола Bit_Rate_capability, определенных в таблице I.2.

a) PCD должно послать действительный кадр команды REQB.

b) Испытательное оборудование отвечает действительным ATQB, включающим в себя байт Bit_Rate_capability в соответствии с таблицей I.2.

c) PCD должно послать команду ATTRIB с допустимым значением параметра для байта Param 2 в соответствии в таблицей I.2.

d) Испытательное оборудование подтверждает прием полученного ATTRIB действительным ответом на команду ATTRIB.

e) PCD должно послать блок $I(0)_0$, используя выбранную согласно Param 2 скорость передачи.

Примечание 1 — Эти блоком может быть также $I(1)_0$ или R(NAK) в случае использования метода 2a для проверки наличия PICC, как описано в ИСО/МЭК 14443-4:2008 (подпункт 7.5.6.2[1]).

f) Испытательное оборудование посылает действительный ответ, используя выбранную по Param 2 скорость передачи. Следует проверить, приняло ли PCD ответ испытательного оборудования.

Примечание 2 — Следующие шаги могут быть не применимы, если PCD встроено в изделие.

g) PCD должно послать запрос S(DESELECT), используя выбранную скорость передачи.

h) Испытательное оборудование посылает действительный ответ S(DESELECT), используя выбранную скорость передачи. Следует проверить, приняло ли PCD ответ испытательного оборудования.

i) PCD должно послать действительный кадр команды REQB, используя скорость передачи *fc*/128.

j) Испытательное оборудование отвечает действительным ATQB, включающим в себя байт Bit_Rate_capability в соответствии с таблицей I.2.

Таблица I.2 — Корректное поведение PCD после ATQB

Bit_Rate_capability	Допустимые значения параметра для Param 2[a)]
(10000000)b	(0000xxxx)b
(10010001)b	(0101xxxx)b, (0000xxxx)b
(10100010)b	(1010xxxx)b, (0000xxxx)b
(10110011)b	(0101xxxx)b, (1010xxxx)b, (0000xxxx)b
(11000100)b	(1111xxxx)b, (0000xxxx)b
(11010101)b	(0101xxxx)b, (1111xxxx)b, (0000xxxx)b
(11100110)b	(1010xxxx)b, (1111xxxx)b, (0000xxxx)b
(11110111)b	(0101xxxx)b, (1010xxxx)b, (1111xxxx)b, (0000xxxx)b
(00000000)b	(0000xxxx)b
(00000001)b	(0001xxxx)b, (0000xxxx)b
(00000010)b	(0010xxxx)b, (0000xxxx)b
(00000011)b	(0001xxxx)b, (0010xxxx)b, (0000xxxx)b
(00000100)b	(0011xxxx)b, (0000xxxx)b
(00000101)b	(0001xxxx)b, (0011xxxx)b, (0000xxxx)b
(00000110)b	(0010xxxx)b, (0011xxxx)b, (0000xxxx)b
(00000111)b	(0001xxxx)b, (0010xxxx)b, (0011xxxx)b, (0000xxxx)b
(00010000)b	(0000xxxx)b (0100xxxx)b

[1] В ИСО/МЭК 10373-6:2011 указан подпункт 7.5.5.2 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

Продолжение таблицы I.2

Bit_Rate_capability	Допустимые значения параметра для Param 2[a)]
(00010001)b	(0001xxxx)b, (0000xxxx)b (0101xxxx)b, (0100xxxx)b
(00010010)b	(0010xxxx)b, (0000xxxx)b (0110xxxx)b, (0100xxxx)b
(00010011)b	(0001xxxx)b, (0010xxxx)b, (0000xxxx)b (0101xxxx)b, (0110xxxx)b, (0100xxxx)b
(00010100)b	(0011xxxx)b, (0000xxxx)b (0111xxxx)b, (0100xxxx)b
(00010101)b	(0001xxxx)b, (0011xxxx)b, (0000xxxx)b (0101xxxx)b, (0111xxxx)b, (0100xxxx)b
(00010110)b	(0010xxxx)b, (0011xxxx)b, (0000xxxx)b (0110xxxx)b, (0111xxxx)b, (0100xxxx)b
(00010111)b	(0001xxxx)b, (0010xxxx)b, (0011xxxx)b, (0000xxxx)b (0101xxxx)b, (0110xxxx)b, (0111xxxx)b, (0100xxxx)b
(00100000)b	(0000xxxx)b (1000xxxx)b
(00100001)b	(0001xxxx)b, (0000xxxx)b (1001xxxx)b, (1000xxxx)b
(00100010)b	(0010xxxx)b, (0000xxxx)b (1010xxxx)b, (1000xxxx)b
(00100011)b	(0001xxxx)b, (0010xxxx)b, (0000xxxx)b (1001xxxx)b, (1010xxxx)b, (1000xxxx)b
(00100100)b	(0011xxxx)b, (0000xxxx)b (1011xxxx)b, (1000xxxx)b
(00100101)b	(0001xxxx)b, (0011xxxx)b, (0000xxxx)b (1001xxxx)b, (1011xxxx)b, (1000xxxx)b
(00100110)b	(0010xxxx)b, (0011xxxx)b, (0000xxxx)b (1010xxxx)b, (1011xxxx)b, (1000xxxx)b
(00100111)b	(0001xxxx)b, (0010xxxx)b, (0011xxxx)b, (0000xxxx)b (1001xxxx)b, (1010xxxx)b, (1011xxxx)b, (1000xxxx)b
(00110000)b	(0000xxxx)b (0100xxxx)b (1000xxxx)b
(00110001)b	(0001xxxx)b, (0000xxxx)b (0101xxxx)b, (0100xxxx)b (1001xxxx)b, (1000xxxx)b
(00110010)b	(0010xxxx)b, (0000xxxx)b (0110xxxx)b, (0100xxxx)b (1010xxxx)b, (1000xxxx)b
(00110011)b	(0001xxxx)b, (0010xxxx)b, (0000xxxx)b (0101xxxx)b, (0110xxxx)b, (0100xxxx)b (1001xxxx)b, (1010xxxx)b, (1000xxxx)b
(00110100)b	(0011xxxx)b, (0000xxxx)b (0111xxxx)b, (0100xxxx)b (1011xxxx)b, (1000xxxx)b

Продолжение таблицы I.2

Bit_Rate_capability	Допустимые значения параметра для Param 2[a)]
(00110101)b	(0001xxxx)b, (0011xxxx)b, (0000xxxx)b (0101xxxx)b, (0111xxxx)b, (0100xxxx)b (1001xxxx)b, (1011xxxx)b, (1000xxxx)b
(00110110)b	(0010xxxx)b, (0011xxxx)b, (0000xxxx)b (0110xxxx)b, (0111xxxx)b, (0100xxxx)b (1010xxxx)b, (1011xxxx)b, (1000xxxx)b
(00110111)b	(0001xxxx)b, (0010xxxx)b, (0011xxxx)b, (0000xxxx)b (0101xxxx)b, (0110xxxx)b, (0111xxxx)b, (0100xxxx)b (1001xxxx)b, (1010xxxx)b, (1011xxxx)b, (1000xxxx)b
(01000000)b	(0000xxxx)b (1100xxxx)b
(01000001)b	(0001xxxx)b, (0000xxxx)b (1101xxxx)b, (1100xxxx)b
(01000010)b	(0010xxxx)b, (0000xxxx)b (1110xxxx)b, (1100xxxx)b
(01000011)b	(0001xxxx)b, (0010xxxx)b, (0000xxxx)b (1101xxxx)b, (1110xxxx)b, (1100xxxx)b
(01000100)b	(0011xxxx)b, (0000xxxx)b (1111xxxx)b, (1100xxxx)b
(01000101)b	(0001xxxx)b, (0011xxxx)b, (0000xxxx)b (1101xxxx)b, (1111xxxx)b, (1100xxxx)b
(01000110)b	(0010xxxx)b, (0011xxxx)b, (0000xxxx)b (1110xxxx)b, (1111xxxx)b, (1100xxxx)b
(01000111)b	(0001xxxx)b, (0010xxxx)b, (0011xxxx)b, (0000xxxx)b (1101xxxx)b, (1110xxxx)b, (1111xxxx)b, (1100xxxx)b
(01010000)b	(0000xxxx)b (0100xxxx)b (1100xxxx)b
(01010001)b	(0001xxxx)b, (0000xxxx)b (0101xxxx)b, (0100xxxx)b (1101xxxx)b, (1100xxxx)b
(01010010)b	(0010xxxx)b, (0000xxxx)b (0110xxxx)b, (0100xxxx)b (1110xxxx)b, (1100xxxx)b
(01010011)b	(0001xxxx)b, (0010xxxx)b, (0000xxxx)b (0101xxxx)b, (0110xxxx)b, (0100xxxx)b (1101xxxx)b, (1110xxxx)b, (1100xxxx)b
(01010100)b	(0011xxxx)b, (0000xxxx)b (0111xxxx)b, (0100xxxx)b (1111xxxx)b, (1100xxxx)b
(01010101)b	(0001xxxx)b, (0011xxxx)b, (0000xxxx)b (0101xxxx)b, (0111xxxx)b, (0100xxxx)b (1101xxxx)b, (1111xxxx)b, (1100xxxx)b
(01010110)b	(0010xxxx)b, (0011xxxx)b, (0000xxxx)b (0110xxxx)b, (0111xxxx)b, (0100xxxx)b (1110xxxx)b, (1111xxxx)b, (1100xxxx)b

Продолжение таблицы I.2

Bit_Rate_capability	Допустимые значения параметра для Param 2[a)]
(01010111)b	(0001xxxx)b, (0010xxxx)b, (0011xxxx)b, (0000xxxx)b (0101xxxx)b, (0110xxxx)b, (0111xxxx)b, (0100xxxx)b (1101xxxx)b, (1110xxxx)b, (1111xxxx)b, (1100xxxx)b
(01100000)b	(0000xxxx)b (1000xxxx)b (1100xxxx)b
(01100001)b	(0001xxxx)b, (0000xxxx)b (1001xxxx)b, (1000xxxx)b (1101xxxx)b, (1100xxxx)b
(01100010)b	(0010xxxx)b, (0000xxxx)b (1010xxxx)b, (1000xxxx)b (1110xxxx)b, (1100xxxx)b
(01100011)b	(0001xxxx)b, (0010xxxx)b, (0000xxxx)b (1001xxxx)b, (1010xxxx)b, (1000xxxx)b (1101xxxx)b, (1110xxxx)b, (1100xxxx)b
(01100100)b	(0011xxxx)b, (0000xxxx)b (1011xxxx)b, (1000xxxx)b (1111xxxx)b, (1100xxxx)b
(01100101)b	(0001xxxx)b, (0011xxxx)b, (0000xxxx)b (1001xxxx)b, (1011xxxx)b, (1000xxxx)b (1101xxxx)b, (1111xxxx)b, (1100xxxx)b
(01100110)b	(0010xxxx)b, (0011xxxx)b, (0000xxxx)b (1010xxxx)b, (1011xxxx)b, (1000xxxx)b (1110xxxx)b, (1111xxxx)b, (1100xxxx)b
(01100111)b	(0001xxxx)b, (0010xxxx)b, (0011xxxx)b, (0000xxxx)b (1001xxxx)b, (1010xxxx)b, (1011xxxx)b, (1000xxxx)b (1101xxxx)b, (1110xxxx)b, (1111xxxx)b, (1100xxxx)b
(01110000)b	(0000xxxx)b (0100xxxx)b (1000xxxx)b (1100xxxx)b
(01110001)b	(0001xxxx)b, (0000xxxx)b (0101xxxx)b, (0100xxxx)b (1001xxxx)b, (1000xxxx)b (1101xxxx)b, (1100xxxx)b
(01110010)b	(0010xxxx)b, (0000xxxx)b (0110xxxx)b, (0100xxxx)b (1010xxxx)b, (1000xxxx)b (1110xxxx)b, (1100xxxx)b
(01110011)b	(0001xxxx)b, (0010xxxx)b, (0000xxxx)b (0101xxxx)b, (0110xxxx)b, (0100xxxx)b (1001xxxx)b, (1010xxxx)b, (1000xxxx)b (1101xxxx)b, (1110xxxx)b, (1100xxxx)b
(01110100)b	(0011xxxx)b, (0000xxxx)b (0111xxxx)b, (0100xxxx)b (1011xxxx)b, (1000xxxx)b (1111xxxx)b, (1100xxxx)b
(01110101)b	(0001xxxx)b, (0011xxxx)b, (0000xxxx)b (0101xxxx)b, (0111xxxx)b, (0100xxxx)b (1001xxxx)b, (1011xxxx)b, (1000xxxx)b (1101xxxx)b, (1111xxxx)b, (1100xxxx)b

Окончание таблицы I.2

Bit_Rate_capability	Допустимые значения параметра для Param 2[a)]
(01110110)b	(0010xxxx)b, (0011xxxx)b, (0000xxxx)b (0110xxxx)b, (0111xxxx)b, (0100xxxx)b (1010xxxx)b, (1011xxxx)b, (1000xxxx)b (1110xxxx)b, (1111xxxx)b, (1100xxxx)b
(01110111)b	(0001xxxx)b, (0010xxxx)b, (0011xxxx)b, (0000xxxx)b (0101xxxx)b, (0110xxxx)b, (0111xxxx)b, (0100xxxx)b (1001xxxx)b, (1010xxxx)b, (1011xxxx)b, (1000xxxx)b (1101xxxx)b, (1110xxxx)b, (1111xxxx)b, (1100xxxx)b
[a)] Младший значащий полубайт байта Param 2 используют для кодирования максимального размера кадра, который может быть принят PCD.	

Сценарий I.2 — Выбор высокой скорости передачи (тип B, процедура 2)

PCD		Оборудование для испытаний PCD
REQB ('05 00 00' CRC_B)	→	
	←	ATQB, Bit_Rate_capability в соответствии с таблицей I.2
Команда ATTRIB, Param 2 в соответствии с таблицей I.2	→	
	←	Ответ на команду ATTRIB (с использованием скорости передачи *fc*/128)
$I(0)_0$ (с использованием выбранной скорости передачи)	→	
	←	$I(0)_0$ (с использование выбранной скорости передачи)
Запрос S(DESELECT)	→	
	←	Ответ S(DESELECT) (с использованием выбранной скорости передачи)
WUPB (с использованием скорости передачи *fc*/128)	→	
	←	ATQB (с использованием скорости передачи *fc*/128)

I.2.2.1 Ожидаемый результат

PCD должно вести себя, как описано в сценарии I.2, в каждом из 72 случаев испытания.

I.2.2.2 Протокол испытаний

Если PCD ведет себя допустимым образом в соответствии со сценарием I.2 в каждом из 72 случаев испытания, то результаты данного испытания считают удовлетворительными. В протоколе испытаний должны быть указаны скорости передачи, выбранные PCD в каждом из 72 случаев испытания.

I.2.3 Процедура для выбора скорости передачи с использованием блоков S(PARAMETERS)

Следующую процедуру необходимо повторить для всех значений информационных байтов скорости передачи, определенных в таблице I.3 для PCD, использующих механизм S(PARAMETERS) для выбора скорости передачи, как определено в ИСО/МЭК 14443-4 (раздел 9).

a) UT выполняет процедуру активации протокола в соответствии с H.1.8.2 для типа A или H.1.8.3 для типа B.

Примечание 1 — Если PCD встроено или согласование не начинается сразу после процедуры активации протокола, то метод активации механизма должен предоставить изготовитель PCD.

b) PCD должно послать блок S(PARAMETERS) для запроса параметров скорости передачи.

c) Испытательное оборудование отвечает действительным блоком S(PARAMETERS), включающим в себя информационные байты скорости передачи в соответствии с таблицей I.3.

Примечание 2 — Тестируют только специальный набор всех возможных скоростей передачи. Второй байт всегда установлен в состояние значения '00' для обоих направлений передачи.

d) PCD должно послать блок S(PARAMETERS) с действительным полем INF, содержащим байты выбора скорости передачи с ровно одним набором бит для скорости передачи от PCD к PICC и ровно одним набором бит для скорости передачи от PICC к PCD, указанными испытательным оборудованием на шаге c).

e) Испытательное оборудование подтверждает прием полученного блока S(PARAMETERS) ответом с действительным блоком S(PARAMETERS).

f) PCD должно послать блок $I(0)_0$, используя выбранную скорость передачи.

Примечание 3 — Этим блоком может быть также I(1)0 или R(NAK) в случае использования метода 2a для проверки наличия PICC, как описано в ИСО/МЭК 14443-4:2008 (подпункт 7.5.6.2[1]).

g) Испытательное оборудование посылает действительный ответ, используя выбранную скорость передачи. Следует проверить, приняло ли PCD ответ испытательного оборудования.

Примечание 4 — Следующие шаги могут быть не применимы, если PCD встроено в изделие.

h) PCD должно послать запрос S(DESELECT), используя выбранную скорость передачи.

i) Испытательное оборудование посылает действительный ответ S(DESELECT), используя выбранную скорость передачи. Следует проверить, приняло ли PCD ответ испытательного оборудования.

j) PCD должно послать действительный WUPA для типа A или WUPB для типа B, используя скорость передачи *fc*/128.

k) Испытательное оборудование отвечает действительным ATQA для типа A или ATQB для типа B.

Таблица I.3 — Поддерживаемые скорости передачи

Первый байт при передаче от PCD к PICC								Первый байт при передаче от PICC к PCD							
b8	b7	b6	b5	b4	b3	b2	b1	b8	b7	b6	b5	b4	b3	b2	b1
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1
0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1
0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	1
0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1

[1] В ИСО/МЭК 10373-6:2011 указан подпункт 7.5.5.2 без учета изменения нумерации пунктов, введенного изменением A1:2012 ИСО/МЭК 14443-4:2008 в связи с внесением дополнительного пункта 7.5.1.

Окончание таблицы I.3

Первый байт при передаче от PCD к PICC								Первый байт при передаче от PICC к PCD							
b8	b7	b6	b5	b4	b3	b2	b1	b8	b7	b6	b5	b4	b3	b2	b1
0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1
0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1
0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1
0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1

Примечание — Кодирование этих байтов определено в ИСО/МЭК 14443-4:2008/Изм. А2:2012 (рисунок 23).

Сценарий I.3 — Выбор скорости передачи с использованием блоков S(PARAMETERS) (процедура 3)

PCD		Оборудование для испытаний PCD
S(PARAMETERS)	→	
	←	S(PARAMETERS), информация о скорости передачи в соответствии в таблицей I.3
S(PARAMETERS), выбор скорости передачи только с одним набором бит для каждого направления передачи из полученных байтов с возможными скоростями передачи	→	
	←	S(PARAMETERS), (с использованием скорости передачи *fc*/128)
$I(0)_0$ (с использованием выбранной скорости передачи)	→	
	←	$I(0)_0$ (с использование выбранной скорости передачи)
Запрос S(DESELECT) (с использование выбранной скорости передачи)	→	
	←	Ответ S(DESELECT) (с использованием выбранной скорости передачи)
WUPA для типа A, WUPB для типа B (с использованием скорости передачи *fc*/128)	→	
	←	ATQA для типа A, ATQB для типа B (с использованием скорости передачи *fc*/128)

I.2.3.1 Ожидаемый результат

PCD должно вести себя, как описано в сценарии I.3, в каждом из 34 случаев испытания.

I.2.3.2 Протокол испытаний

Если PCD ведет себя допустимым образом в соответствии со сценарием I.3 в каждом из 34 случаев испытания, то результаты данного испытания считают удовлетворительными. В протоколе испытаний должны быть указаны скорости передачи, выбранные PCD в каждом из 34 случаев испытания.

(Введены дополнительно. Изм. А4:2012.)

Приложение J
(справочное)

Программа для измерения уровня EMD

Следующая программа на языке С может быть использована для выполнения измерений уровня EMD для скоростей передачи от PICC к PCD *fc*/128, *fc*/64, *fc*/32 и *fc*/16.

Примечание 1 — Для измерений уровня EMD с поднесущей частотой PICC выше, чем *fc*/16, программу необходимо соответствующим образом перенастроить.

Примечание 2 — Выходная информация (время, USB, LSB) может зависить от опций компилятора и архитектуры используемой рабочей системы.

```
/*****************************************************************************/
/*** This program calculates the upper side band (USB) and                 ***/
/*** lower side band (LSB) load modulation amplitudes                      ***/
/*** versus time of a PICC for the evaluation of EMD levels      ***/
/*** according to ISO/IEC 10373-6/Amd.2                                    ***/
/*****************************************************************************/
/*** Input:                                                                ***/
/*** File in CSV format containing a table of two                          ***/
/*** columns (time and sense coils' voltage)                               ***/
/*** data format of input-file:                                            ***/
/*** - one data-point per line:                                            ***/
/*** time[seconds], sense-coil-voltage[volts])                             ***/
/*** - contents in ASCII, no headers                                       ***/
/*** - data-points shall be equidistant time                               ***/
/*** - minimum sampling rate: 100 MSamples/second                          ***/
/*** - At least 200 microsecond before start of PICC                       ***/
/*** sub-carrier generation                                                ***/
/*** - At least 50 microsecond after start of PICC                         ***/
/*** sub-carrier generation                                                ***/
/***                                                                       ***/
/*** example for spreadsheet file (start in next line):                    ***/
/*** (time) (voltage)                                                      ***/
/*** 3.00000e-06,1.00                                                      ***/
/*** 3.00200e-06,1.01                                                      ***/
/*** .....                                                                 ***/
/***                                                                       ***/
/*** Ouput:                                                                ***/
/*** File in CSV format containing the results                             ***/
/*** in a table of three columns (time, USB, LSB)                          ***/
/***                                                                       ***/
/***                                                                       ***/
/*****************************************************************************/
/*** RUN:                                                                  ***/
/*** "exefilename" filename[.csv]                                          ***/
/*****************************************************************************/
/*** ISO/IEC 10373-6 EMD levels Calculation                                ***/
/*** (according to clause 5.5.2)                                           ***/
/*** Program Version 1.0 Release January 2010                              ***/
/*****************************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

#define MAX_SAMPLES 5000000
#define MAX_LMA 100000
```

```
/*** Declare function prototype ***/
int File_info(char * , double *);
int readcsv(char * ,double * ,double * );
int writecsv(char* ,int ,double * ,double * ,double * );
int Sliding_LMA(double ,int ,double *,double *,int ,double *,double *,double *);
void Noise_STD(int ,double *,double *, double *,double *);

/*****************************************************************************************/
/*** File_info function                                                                ***/
/***                                                                                   ***/
/*** Description:                                                                      ***/
/*** This function parse a file in CSV format                                          ***/
/*** to determine the number of lines and sampling rate                                ***/
/***                                                                                   ***/
/*** Input: filename                                                                   ***/
/***                                                                                   ***/
/*** Return: Number of samples (sample Count)                                          ***/
/***         0 if an error occurred                                                    ***/
/***                                                                                   ***/
/*****************************************************************************************/
int File_info(char * fname , double * pdt)
{
     int i, c;
     double t1,v1,t2,v2;
     FILE     *sample_file;

     /************ Open File ***********************/
     if (!strchr(fname, '.')) strcat(fname, ".csv");

     if ((sample_file = fopen(fname, "r"))== NULL)
     {
          printf("Cannot open input file %s.\n",fname);
          return 0;
     }
     /*** Read two first lines to retrieve sampling rate ***/
     fscanf(sample_file,"%Lf,%Lf\n", &t1, &v1);
     if (feof(sample_file))
     {
          fclose(sample_file);
          return 0;
     }
     fscanf(sample_file,"%Lf,%Lf\n", &t2, &v2);
     if (feof(sample_file))
     {
          fclose(sample_file);
          return 0;
     }
     *pdt=t2-t1;

     i=2;

     while (!feof(sample_file))
     {
          c = fgetc(sample_file);
          if (c == '\n')
          {
               i++;
               if (i>=MAX_SAMPLES)
               {
                    printf("Too many samples in input file: only %d samples retained\n",i);
                    break;
```

```
                    }
                }
            }
            fclose(sample_file);
            return i;
        }
/*****************************************************************************************/

/*****************************************************************************************/
/*** Read CSV File Function                                                            ***/
/***                                                                                   ***/
/*** Description:                                                                      ***/
/*** This function reads the table of time and sense coil                              ***/
/*** voltage from a File in CSV Format                                                 ***/
/***                                                                                   ***/
/*** Input: filename                                                                   ***/
/***                                                                                   ***/
/*** Return: Number of samples (sample Count)                                          ***/
/***         0 if an error occurred                                                    ***/
/***                                                                                   ***/
/*** Displays Statistics:                                                              ***/
/***                                                                                   ***/
/*** Filename, SampleCount, Sampling rate, Max/Min Voltage                             ***/
/*****************************************************************************************/

int readcsv(char* fname,double vtime[],double vd[])
{
      double max_vd,min_vd;
      int i;

      FILE     *sample_file;

      /************ Open File ***********************/
      if (!strchr(fname, '.')) strcat(fname, ".csv");

      if ((sample_file = fopen(fname, "r"))== NULL)
      {
           printf("Cannot open input file %s.\n",fname);
           return 0;
      }


/*****************************************************************************************/
/* Read CSV File                                                                       ***/
/*****************************************************************************************/
      max_vd=-1e-9F;
      min_vd=-max_vd;
      i=0;

      while (!feof(sample_file))
          {
             if (i>=MAX_SAMPLES)
             {
             printf("Too many samples in input file: only %d samples read\n",i);
             break;
             }
          fscanf(sample_file,"%lf,%lf\n", &vtime[i], &vd[i]);
          if (vd[i]>max_vd) max_vd=vd[i];
          if (vd[i]<min_vd) min_vd=vd[i];
          i++;
          }
```

```
    fclose(sample_file);

    /************ Displays Statistics ********************************************/
    printf("\n*****************************************************************\n");

    printf("Statistics: \n");
    printf(" Filename                                                    : %s\n",fname);
    printf(" Sample count                                                : %d\n",i);
    printf(" Sampling rate : %1.0f MHz\n",1e-6/(vtime[1]-vtime[0]));
    printf(" Max(vd)                                                     : %4.0f mV\n",max_vd*1000);
    printf(" Min(vd)                                                     : %4.0f mV\n",min_vd*1000);
    return i;
}
/****** End ReadCsv ***********************************************************/

/*****************************************************************************/
/*** Write CSV File Function                                                 ***/
/***                                                                         ***/
/*** Description:                                                            ***/
/*** This function writes in a CSV format file                               ***/
/*** the result of LMA computation:                                          ***/
/*** Fist column = time(s)                                                   ***/
/*** Second column = Upper side band amplitude (V)                           ***/
/*** Third column = Lower side band amplitude (V)                            ***/
/***                                                                         ***/
/*** Return: Number of written samples                                       ***/
/***         0 if an error occurred                                          ***/
/*****************************************************************************/

int writecsv(char* fname,int n_LMA,double LMA_time[],double USB[],double LSB[])
{
    int i;
    FILE *out_file;
    if ((out_file = fopen(fname, "w"))== NULL)
        {
            printf("Cannot open output file %s.\n",fname);
            return 0;
        }
        for (i=0;i<n_LMA;i++)
        {
            fprintf(out_file,"%7.4E,%7.4E,%7.4E\n",LMA_time[i],USB[i],LSB[i]);
        }
        fclose(out_file);
        return i;
}

/*****************************************************************************/
/*** Sliding_LMA : Load Modulation Amplitude versus Time                      ***/
/*****************************************************************************/
/*** Description:                                                            ***/
/*** This function calculates Upper side band and                            ***/
/*** Lower side band amplitudes as a function of time                        ***/
/***                                                                         ***/
/*** Arguments:                                                              ***/
/*** fc = carrier frequency (Hz)                                             ***/
/*** count = number of input signal samples                                  ***/
/*** vtime[] = input signal time array                                       ***/
/*** vd[] = input signal voltage array                                       ***/
/*** lout = max. size of following arrays                                    ***/
/*** LMA_time[] = Times to which LMAs are computed                           ***/
/*** USB[] = load modulation amplitude at fc+fs                              ***/
```

```
/*** LSB[] - load modulation amplitude at fc-fs                                    ***/
/***                                                                              ***/
/*** return value: number of computed LMA                                         ***/
/**********************************************************************************/
int Sliding_LMA(double fc,int count,double vtime[],double vd[],int lout,double LMA_
time[],double USB[],double LSB[])
{
     double c1_real,c1_imag;
     double c2_real,c2_imag;
     double w0,wu,wl,dt;
     double Wb;                    /* Bartlett window coefficient */
     int i,j,k=0;
     int N_data;                   /* Time window size*/
     int N_over;                   /* Overlap */
     int N_middle;                 /* Half window size */
     double *Yuc,*Yus,*Ylc,*Yls; /* Phase factors */
     double pi;                    /* pi=3.14.... */
     double sum_Wb=0;              /* Sum of Bartlett coeff. */
     double cf;                    /* correction factor of the Bartlett window */

     pi = (double)atan(1.0)*4; /* calculate pi */

     w0=(double)(fc*2.0)*pi; /* carrier 13.56 MHz */
     wu=(double)(1.0+1.0/16.0)*w0; /* upper sideband 14.41 MHz */
     wl=(double)(1.0-1.0/16.0)*w0; /* lower sideband 12.71 MHz */


     /********** Time window ***********************************************************/
     dt=vtime[2]-vtime[1]; /* Note: (vtime[2]-vtime[1]) is the scope sampling rate */
     N_data=(int)(0.5+2*16.0F/dt/fc); /* Number of samples for two subcarrier periods */
     N_middle=(int) (0.5+N_data/2);
     N_over=(int) (0.5+1.0/dt/fc); /* Overlap of 1/fc */

     /******* Allocate memory ***************************************************/
     Yuc=(double *) malloc(N_data * sizeof(double));
     if (Yuc == NULL)
     {
          printf("Cannot allocate memory");
          return 0;
     }
     Yus=(double *) malloc(N_data * sizeof(double));
     if (Yus == NULL)
     {
          printf("Cannot allocate memory");
          free(Yuc);
          return 0;
     }
     Ylc=(double *) malloc(N_data * sizeof(double));
     if (Ylc == NULL)
     {
          printf("Cannot allocate memory");
          free(Yuc); free(Yus);
          return 0;
     }
     Yls=(double *) malloc(N_data * sizeof(double));
     if (Yls == NULL)
     {
          printf("Cannot allocate memory");
          free(Yuc); free(Yus); free(Ylc);
          return 0;
     }
```

```
/************ Calculate apodization window and phase factors ****/
for( i=0;i<N_data;i++)
    {
         /* Bartlett window */
         if ((N_data & 1) == 0)
         {
              /* N_data is even */
              if (i < (int) (N_data /2))
              {
                   Wb=2.0F*i/(double)(N_data - 1);
              }
              else
              {
                   Wb=2.0F*(N_data-i-1)/(double)(N_data - 1);
              }
         }
         else
         {
              /*N_data is odd */
              if (i <= (int) (0.001+(N_data-1) /2))
              {
                   Wb=2.0F*i/(double)(N_data - 1);
              }
              else
              {
                   Wb=2.0F-2.0F*i/(double)(N_data - 1);
              }
         }
         Yuc[i]=(double)cos(wu*i*dt)*Wb;
         Yus[i]=(double)sin(wu*i*dt)*Wb;
         Ylc[i]=(double)cos(wl*i*dt)*Wb;
         Yls[i]=(double)sin(wl*i*dt)*Wb;
         sum_Wb += Wb;
    }
    cf=N_data/sum_Wb;

/****************** DFT ***********************************************************/

for (j=0;j<count-N_data;j=j+N_over)
{
     c1_real=0; /* real part of the up. sideband fourier coefficient */
     c1_imag=0; /* imag part of the up. sideband fourier coefficient */
     c2_real=0; /* real part of the lo. sideband fourier coefficient */
     c2_imag=0; /* imag part of the lo. sideband fourier coefficient */

     for( i=0;i<N_data;i++)
     {
          c1_real=c1_real+vd[i+j]*Yuc[i];
          c1_imag=c1_imag+vd[i+j]*Yus[i];
          c2_real=c2_real+vd[i+j]*Ylc[i];
          c2_imag=c2_imag+vd[i+j]*Yls[i];

     }

     /****************** DFT scale ***********************************************/

     c1_real=2.0F*cf*c1_real/(double) N_data;
     c1_imag=2.0F*cf*c1_imag/(double) N_data;
     c2_real=2.0F*cf*c2_real/(double) N_data;
     c2_imag=2.0F*cf*c2_imag/(double) N_data;
```

```
            /**************** absolute fourier coefficient **********/
            USB[k]=(double)sqrt(c1_real*c1_real + c1_imag*c1_imag);
            LSB[k]=(double)sqrt(c2_real*c2_real+c2_imag*c2_imag);

            /************** Half window time ************************/
            LMA_time[k]=vtime[j+N_middle];    /* Half window time */
            k++;
            if (k > lout) break; /* stop if array size is reached*/
      }

      free(Yuc);
      free(Yus);
      free(Ylc);
      free(Yls);
      return k;

}
/********* End DFT ***************************************************************************/

/*************************************************************************************************/
/*** Noise_STD : Noise standard deviation                                                      ***/
/*************************************************************************************************/
/*** Description:                                                                             ***/
/*** This function calculates the standard deviations                                         ***/
/*** at fc+fs and fc-fs as required by the noise                                              ***/
/*** precondition test of ISO/IEC 10373-6.                                                    ***/
/*** Results are meaningful only when the sense coil's                                        ***/
/*** signal is recorded with a reference PICC.                                                ***/
/***                                                                                         ***/
/*** Arguments:                                                                               ***/
/*** n_LMA = number of input values                                                           ***/
/*** USB[] = load modulation amplitude at fc+fs                                               ***/
/*** LSB[] = load modulation amplitude at fc-fs                                               ***/
/*** pSTD_USB= standard deviation at fc+fs                                                    ***/
/*** pSTD_LSB= standard deviation at fc-fs                                                    ***/
/***                                                                                         ***/
/*************************************************************************************************/
void Noise_STD(int n_LMA,double USB[],double LSB[], double *pSTD_USB,double *pSTD_LSB)
{
      double P_USB=0,P_LSB=0;
      int i;
      /**************** Square summation ********************/
      for( i=0;i<n_LMA;i++)
      {
            P_USB += USB[i]*USB[i];
            P_LSB += LSB[i]*LSB[i];
      }
      *pSTD_USB=sqrt(P_USB/n_LMA);
      *pSTD_LSB=sqrt(P_LSB/n_LMA);

}

/*************************************************************************************************/
/*** MAIN Program                                                                             ***/
/*************************************************************************************************/
int main(int argc, char *argv[])
{
      char fname[256];
      char fout[256];
      int sample_count;
      int lout;         /*Maximun length of result arrays */
```

```
int n_LMA;          /* Number of computed LMA */
int status;
double fc;              /* Carrier frequency */
double std_USB,std_LSB,dt;
double *pTime, *pVolts, *pLMA_time, *pUSB, *pLSB;

fc=13.56e6;

printf("\n");
printf("*************************************************************************\n");
printf("**** ISO/IEC 10373-6 EMD Test-Program ***********************************\n");
printf("**** Version: 1.0 January 2010 ******************************************\n");
printf("****(according to ISO/IEC 10373-6 , clause 5.5.2)************************\n");
printf("*************************************************************************\n");
if (argc > 1)
{
     /*** First input parameter is taken as input file name ***/
     strcpy(fname, argv[1]);
}
else
{
     /*** No input parameter ***/
     printf("\nCSV File name :");
     scanf("%s",fname);
}

if (!strchr(fname, '.')) strcat(fname, ".csv");
if (!(sample_count=File_info(fname , &dt))) return 0;
lout= (int) (sample_count/(int)(0.5+1.0/dt/fc));
if (lout > MAX_LMA) lout = MAX_LMA;

/*** Start of memory allocation                                              ***/
pTime= (double *) malloc(sample_count * sizeof(double));
if (pTime == NULL)
{
     printf("Cannot allocate memory");
     return 0;
}
pVolts= (double *) malloc(sample_count * sizeof(double));
if (pVolts == NULL)
{
     printf("Cannot allocate memory");
     free(pTime);
     return 0;
}
pUSB= (double *) malloc(lout * sizeof(double));
if (pUSB == NULL)
{
     printf("Cannot allocate memory");
     free(pTime); free(pVolts);
     return 0;
}
pLSB= (double *) malloc(lout * sizeof(double));
if (pLSB == NULL)
{
     printf("Cannot allocate memory");
     free(pTime); free(pVolts); free(pUSB);
     return 0;
}
pLMA_time= (double *) malloc(lout * sizeof(double));
if (pLMA_time == NULL)
```

```
    {
        printf("Cannot allocate memory");
        free(pTime); free(pVolts); free(pUSB); free(pLSB);
        return 0;
    }
    /*** End of memory allocation                                                    ***/

    if (!(sample_count=readcsv(fname,pTime,pVolts))) return 0; /* reading data */
    if (!(n_LMA=Sliding_LMA(fc,sample_count,pTime,pVolts,fout,pLMA_    time,pUSB,pLSB)))
return 0;

        /***** processing data ***************************************************/
    strcpy(fout,"LMA_");
    strcat(fout,fname);
    status=writecsv(fout,n_LMA,pLMA_time,pUSB,pLSB); /* writing results in a file */
    Noise_STD(n_LMA,pUSB,pLSB,&std_USB,&std_LSB); /* evaluating noise floor */

    /************** Result Display ***************************************************/
    printf("\n");
    printf("*************************************************************************\n");
    printf(" Noise floor : \n");
    printf(" standard deviation at fc+fs: %7.3f mV\n",std_USB*1000);
    printf(" standard deviation at fc-fs: %7.3f mV\n",std_LSB*1000);
    printf(" Note: Displayed results are meaningful only when\n");
    printf("       the sense coil's signal is recorded with a\n");
    printf("       reference PICC.");
    printf("\n*************************************************************************\n");

    free(pTime);
    free(pVolts);
    free(pLMA_time);
    free(pUSB);
    free(pLSB);

    return 1;
}
/*********** End Main ********************************************************************/
```

(Введено дополнительно. Изм. А2:2012.)
(Измененная редакция. Поправка Cor.1:2013.)

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации и межгосударственным стандартам

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК 7810:2003	IDT	ГОСТ Р ИСО/МЭК 7810—2015 «Карты идентификационные. Физические характеристики»
ИСО/МЭК 14443-1:2008	IDT	ГОСТ Р ИСО/МЭК 14443-1—2013 «Карты идентификационные. Карты на интегральных схемах бесконтактные. Карты близкого действия. Часть 1. Физические характеристики»
ИСО/МЭК 14443-2:2010	IDT	ГОСТ Р ИСО/МЭК 14443-2—2014 «Карты идентификационные. Карты на интегральных схемах бесконтактные. Карты близкого действия. Часть 2. Радиочастотный энергетический и сигнальный интерфейс»
ИСО/МЭК 14443-3:2011	IDT	ГОСТ Р ИСО/МЭК 14443-3—2014 «Карты идентификационные. Карты на интегральных схемах бесконтактные. Карты близкого действия. Часть 3. Инициализация и антиколлизия»
ИСО/МЭК 14443-4:2008	IDT	ГОСТ Р ИСО/МЭК 14443-4—2014 «Карты идентификационные. Карты на интегральных схемах бесконтактные. Карты близкого действия. Часть 4. Протокол передачи»
МЭК 61000-4-2:2008	MOD	ГОСТ 30804.4.2—2013 (IEC 61000-4-2:2008) «Совместимость технических средств электромагнитная. Устойчивость к электростатическим разрядам. Требования и методы испытаний»

Примечание — В настоящей таблице использованы следующие условные обозначения степени соответствия стандартов:
- IDT — идентичные стандарты;
- MOD — модифицированные стандарты.

Библиография

[1] ISO/IEC Guide 98-3:2008, Uncertainty of measurement — Part 3: Guide to the expression of uncertainty in measurement (GUM:1995). [Руководство ИСО/МЭК 98-3:2008[1] Неопределенность измерений. Часть 3. Руководство по выражению неопределенности измерений (GUM:1995)]

[2] ISO/IEC 9646-1:1994, Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 1: General concepts. (ИСО/МЭК 9646-1:1994 Информационные технологии. Взаимосвязь открытых систем. Методология и основы аттестационного тестирования. Часть 1. Общие положения)

[3] ISO/IEC 9646-2:1994, Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 2: Abstract Test Suite specification. (ИСО/МЭК 9646-2:1994 Информационные технологии. Взаимосвязь открытых систем. Методология и основы аттестационного тестирования. Часть 2. Спецификация комплекта абстрактных тестов)

[4] ISO/IEC 9646-3:1998, Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 3: The Tree and Tabular Combined Notation (TTCN). [ИСО/МЭК 9646-3:1998 Информационные технологии. Взаимосвязь открытых систем. Методология и основы аттестационного тестирования. Часть 3. Комбинированная древовидная и табличная нотация (КДТН)]

[5] ISO/IEC 9646-4:1994, Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 4: Test realization. (ИСО/МЭК 9646-4:1994 Информационная технология. Взаимосвязь открытых систем. Методология и основы аттестационного тестирования. Часть 4. Реализация тестов)

[6] ISO/IEC 9646-5:1994, Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 5: Requirements on test laboratories and clients for the conformance assessment process. (ИСО/МЭК 9646-5:1994 Информационная технология. Взаимосвязь открытых систем. Методология и основы аттестационного тестирования. Часть 5. Требования к испытательным лабораториям и клиентам в процессе оценки соответствия)

[7] ISO/IEC 9646-6:1994, Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 6: Protocol profile test specification. (ИСО/МЭК 9646-6:1994 Информационные технологии. Взаимосвязь открытых систем. Методология и основы аттестационного тестирования. Часть 6. Спецификация теста для профилей протоколов)

[8] ISO/IEC 9646-7:1995, Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 7: Implementation Conformance Statements. (ИСО/МЭК 9646-7:1995 Информационные технологии. Взаимодействие открытых систем. Методология и основы аттестационного тестирования. Часть 7. Отчет о реализации совместимости)

[1] Действует ГОСТ Р 54500.3–2011/Руководство ИСО/МЭК 98-3:2008 «Неопределенность измерения. Часть 3. Руководство по выражению неопределенности измерения».

УДК 336.77:002:006.354 ОКС 35.240.15 Э49 ОКСТУ 4084

Ключевые слова: обработка данных, обмен информацией, устройства хранения данных, идентификационные карты, карты близкого действия, методы испытаний

Редактор *Л.И. Потапова*
Технический редактор *В.Н. Прусакова*
Корректор *Е.Р. Ароян*
Компьютерная верстка *И.В. Белюсенко*

Сдано в набор 09.11.2015. Подписано в печать 15.12.2015. Формат 60 × 84 $^{1}/_{8}$. Гарнитура Ариал.
Усл. печ. л. 24,18. Уч.-изд. л. 21,75. Тираж 33 экз. Зак 627.

Набрано в ИД «Юриспруденция», 115419, Москва, ул. Орджоникидзе, 11.
www. jurisizdat. ru gost.e-ecolog.ru

Издано и отпечатано во
ФГУП «СТАНДАРТИНФОРМ», 123995 Москва. Гранатный пер., 4
www.gostinfo.ru gost.e-ecolog.ru

gost.e-ecolog.ru