Тонкий клиент (Thinclient). Описание сетевого протокола компоненты «Клиент отображаемых форм»

1. окт. 2017

mailto:nilstarsoft@mail.ru

Оглавление

1 Введение

Приложение для отображения форм предназначено для выполнения на терминале Pax S900 IRAS (Prolin 2.4).

Компонента отображаемых форм предназначена для:

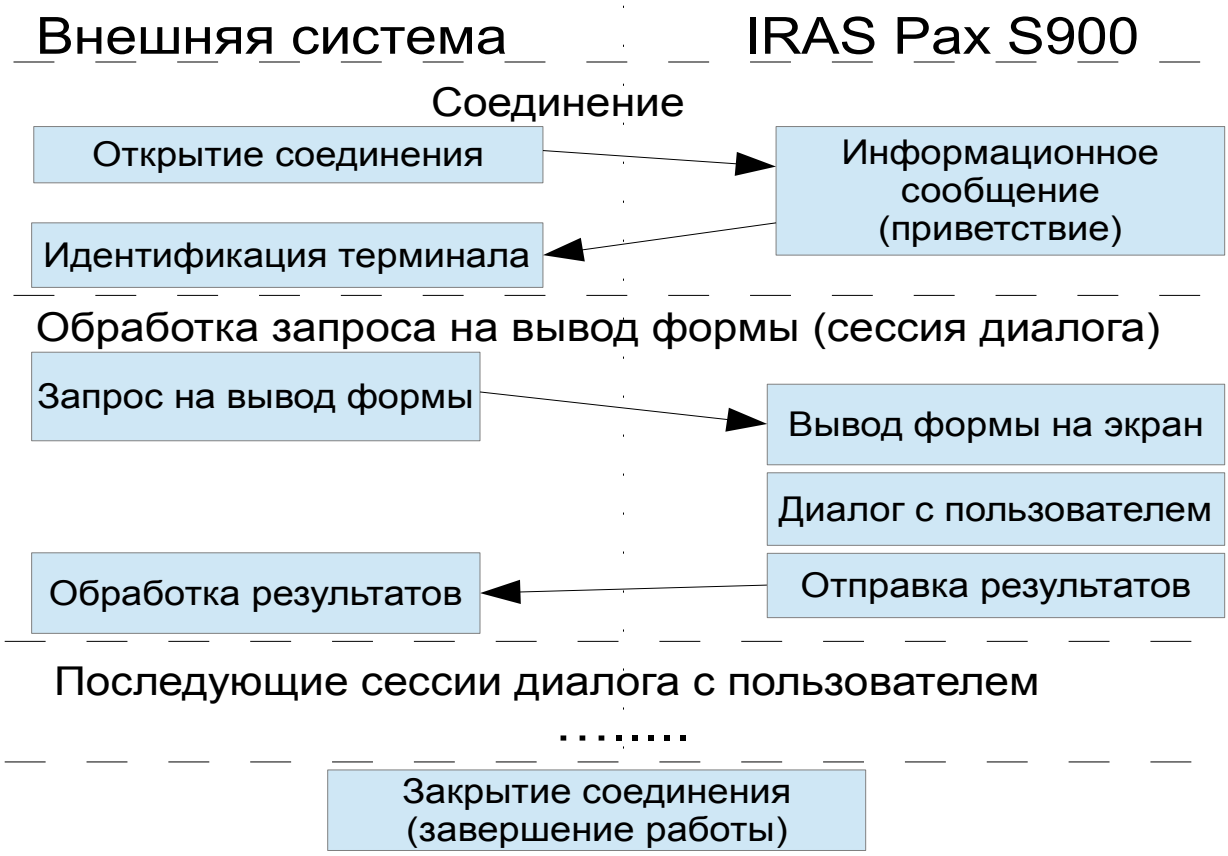
- Получения запроса на отображения формы от внешней системы через сетевое соединение;
- Отображение формы на экране терминала Pax S900 IRAS;
- Обработка вводимой информации с элементов ввода терминала (кнопки, тачскрин, ...);
- Отправка результатов ввода внешней системе.

Обмен сообщениями осуществляется на уровне TCP/IP соединения. В целях защиты данных соединение может быть зашифровано с применением SSL.

2 Сценарии обмена

IRAS Pax S900 является серверной стороной, внешняя система является клиентской стороной. После организации соединения, происходят сессии диалога с пользователем.

2.1 Типовой сценарий работы



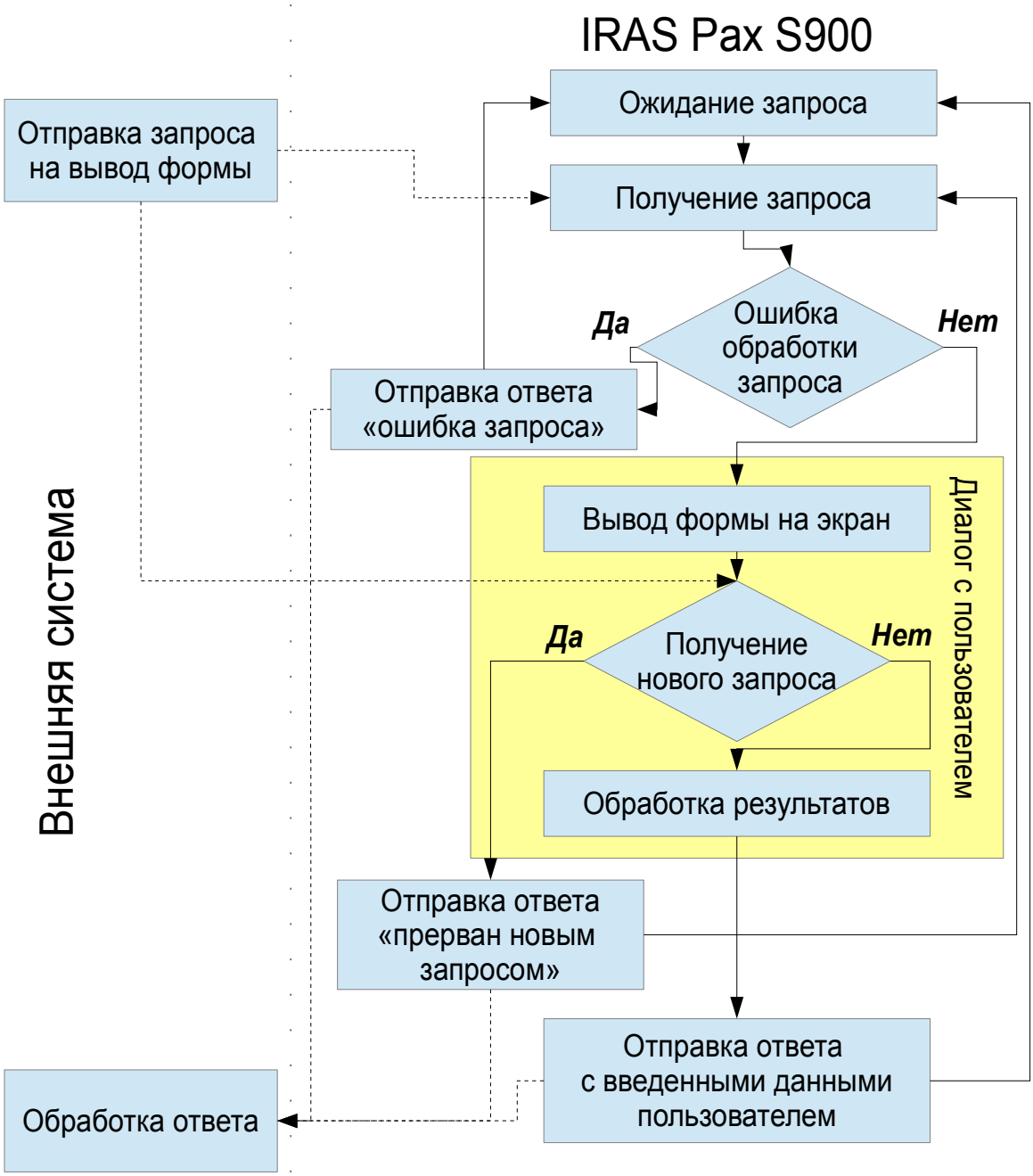
Этот сценарий описывает процесс работы внешней системы с терминалом в рамках одного

соединения.

Упрощенное описание сценария:

1. Открытие соединения внешней системы к терминалу IRAS;
2. Отправка «приветствия» от терминала IRAS к внешней системе;
3. Отправка запроса на вывод формы от внешней системы к терминалу IRAS;
4. Сессия диалога с пользователем терминала IRAS;
5. Отправка ответа с результатами от терминала IRAS к внешней системе;
6. Обработка ответа внешней системой. Переход к пункту 3 или к следующему пункту;
7. Закрытие соединения внешней системой.

2.2 Алгоритм сессии диалога с пользователем



2.3 Квитирование

- Каждый запрос/ответ содержит поле «идентификатор запроса», содержащее уникальный идентификатор. Этот идентификатор должны содержать все запросы/ответы в рамках одной сессии диалога с пользователем.
- Так как TCP/IP является соединением с гарантированной доставкой пакетов, перезапроса доставки пакетов не требуется. То есть ситуация «разрыв соединения» интерпретируется, как сбой со сбрасыванием терминалом IRAS текущей сессии диалога и переходом терминала в начальное состояние.

2.4 Защита соединения

Для защиты соединения от несанкционированного доступа используется схема защиты SSL.

В случае схемы односторонней аутентификации:

- Издается один набор сертификат + закрытый ключ на базе корневого сертификата (CA), для серверной стороны.
- Набор сертификат + закрытый ключ устанавливается на серверную сторону.
- При инициализации соединения, клиентской стороной производится проверка сертификата сервера с помощью корневого сертификата (CA).

В случае схемы двухсторонней аутентификации:

- Издается два набора сертификат + закрытый ключ на базе корневого сертификата (CA), для серверной и клиентской стороны.
- Один набор сертификат + закрытый ключ устанавливается на серверную сторону.
- Второй набор сертификат + закрытый ключ устанавливается на клиентскую сторону
- При инициализации соединения, клиентской стороной производится проверка сертификата сервера с помощью корневого сертификата (CA), и серверной стороной также производится проверка сертификата клиента с помощью корневого сертификата (CA).

2.5 Детектирование нахождения терминала IRAS в сети

При установленном соединении физического уровня (т. е. соединения с WiFi точкой доступа), терминал IRAS отсылает широковещательные сообщения (broadcast) с информацией об себе с регулярным периодом (период и порт для отправки сообщения указывается в настройках терминала).

Внимание: Отправку сообщений можно настроить на конкретный адрес сервера (смотри «Руководство администратора»).

Формат содержимого сообщения:

```
00000900001<x00>  //серийный номер фискального модуля
50340622<x00>  //серийный номер терминала IRAS
1.0.0 Feb 16 2015<x00> //версия ПО и дата сборки
<x00> //завершающий ноль
```

2.6 Ситуации, при которых обработка запросов происходит с задержками

Так как при обработке запросов, может происходить взаимодействие с другими приложениями в терминале IRAS, то в этом случае могут быть задержки обработки запроса, который поступил при работе другого приложения.

Это возможно в следующих ситуациях:

- Идет работа с модулем обслуживания банковских карт. Этот модуль является сторонним приложением и при вызове его функций, основное приложение ожидает возврата из модуля. На время работы модуля обслуживания банковских карт, основное приложение не обрабатывает запросы.

- Идет работа с модулем обслуживания фискального регистратора (KKTIRAS). Этот модуль является сторонним приложением и при вызове его функций, основное приложение ожидает возврата из модуля. На время работы модуля обслуживания фискального регистратора, основное приложение не обрабатывает запросы.

3 Форматы сообщений

Запросы/ответы имеют внутреннее представление согласное формату XML.

Для региональных параметров используется кодировка UTF8.

Все запросы/ответы завершаются стоповым символом (байт со значением 0), т. е. получаемые данные являются строкой.

3.1 Формат «приветствия»

<XML ...> - заголовок XML документа
<HELLO> - начало данных запроса

```
  <TD>[время/дата]</TD>
  <PROG>[название программы]</PROG>
  <VERSION>[версия]</VERSION>
  <FR>[серийный номер фискального регистратора]</FR>
  <SN>[серийный номер терминала]</SN>
  <PN>[продуктовый номер терминала]</PN>
  <SCR w="[ширина экрана]" h="[высота экрана]"
       cw="[ширина клиентской части экрана]"
       ch="[высота клиентской части экрана]"
       touch="[0: нет тачскрина, 1: есть тачскрин]" />
```

</HELLO> - конец данных запроса
[0x00] – стоповый символ

Пример сообщения:

```
<?xml version="1.0" encoding="utf-8"?>
<HELLO>
  <TD>2127 050916</TD>
  <PROG>THINCLIENT</PROG>
  <VERSION>1.1.0 Sep 5 2016</VERSION>
  <FR>00000000001</FR>
  <SN>55001894</SN>
  <PN>S900-0G0-363-05CC</PN>
  <SCR w="240" h="320" cw="240" ch="300" touch="1" />
</HELLO>
```

3.2 Формат запроса на отображение формы

Структура запроса:

<XML ...> - заголовок XML документа
<REQUEST> - начало данных запроса

```
  <ID>[идентификатор запроса (число: unsigned int)]</ID>
  <TD>[время/дата]</TD>
  <FORM>[описание параметров и содержимого формы]</FORM>
  <VALUES>[начальные значения полей ввода формы]</VALUES>
```

</REQUEST> - конец данных запроса
[0x00] – стоповый символ

В зависимости от типа формы тэг FORM может заменен на другой, например MENU – для отображения меню и т. п.

3.2.1 Формат поля VALUES

```
<VALUES>
  <[имя переменной]>[значение переменной]</[имя переменной]>
  ...
  <[имя переменной]>[значение переменной]</[имя переменной]>
</VALUES>
```

Имя переменной должно соответствовать в точности имени поля (атрибут «name») в описании отображаемого объекта формы (смотри 4).

Значения переменной может иметь любого типа (допустимого для соответствующего отображаемого объекта, перечень типов смотрите ниже), в качестве значения может указываться несколько типов одновременно (если это допустимо в отображаемом объекте).

Например установка значений для отображаемого объекта «кнопка» с именем «BUTTON_OK»:

```
<VALUES>
  <BUTTON_OK>
   <TEXT>OK</TEXT>
   <IMAGE>ok.jpg</IMAGE>
  </BUTTON_OK>
</VALUES>
```

Внимание: Если в форме указано несколько объектов с одним именем, значения будут присваиваться первому.

3.3 Формат ответа ошибки обработки запроса (или в случае прерывания новым запросом)

Структура ответа:

`<XML ...>` - заголовок XML документа
`<RESPONSE>` - начало данных подтверждения

```
  <ID>[идентификатор запроса (число: unsigned int)]</ID>
  <TD>[время/дата]</TD>
  <BT>[заряд батареи]</BT>
  <ERROR>[код ошибки обработки запроса]</ERROR>
```

`</RESPONSE>` - конец данных подтверждения
`[0x00]` – стоповый символ

Код ошибки является десятичным числом и может принимать следующие значения:

- 0 — запрос является ответом с введенными данными (смотри раздел 3.4);
- 1 — текущая сессия диалога прервана новым запросом;
- 2 — ошибка обработки данных запроса (данные запроса не соответствуют описанному формату — возможно устаревшая версия ПО);
- 3 — ошибка создания ответа;
- 4 — ошибка модуля поддержки платежных карт;
- 5 — работает фоновое меню настроек (оператор терминала вошел в фоновое меню);
- 6 — сканер штрихкодов не поддерживается;

- 7 — ошибка инициализации безконтактного ридера (для чтения mifare карт);
- 8 — ошибка инициализации считывателя магнитных карт.

3.3.1 Значение поля BT ответа (текущий заряд батареи)

Значение поля BT ответа содержит целое число означающее текущий заряд батареи в процентах (0..100) и флаги состояния.

Внимание: Младший байт числа соответствует значению в процентах текущего заряда батареи.

Могут использоваться следующие флаги состояния:

- x1000 — батарея имеет критически низкий заряд (рекомендуется установить устройство на подзарядку или сменить батарею на заряженную);
- x2000 – терминал подключен к внешнему источнику питания или зарядному устройству.

3.4 Формат ответа с введенными данными

Структура ответа:

`<XML ...>` - заголовок XML документа
`<RESPONSE>` - начало данных ответа

```
  <ID>[идентификатор запроса (число: unsigned int)]</ID>
  <TD>[время/дата]</TD>
  <BT>[заряд батареи]</BT>
  <ERROR>0</ERROR>
  <STATE>[код завершения формы]</STATE>
  <VALUES>[введенные значения полей ввода формы]</VALUES>
```

`</RESPONSE>` - конец данных ответа
`[0x00]` – стоповый символ

3.4.1 Значения поля STATE ответа

Значение поле STATE ответа формируется на основе состояние завершения формы и может принимать следующие значения:

Значение	**Тип формы**	**Причина**
-1	Все типы форм	Форма завершена с состоянием «Отмена».
1	FORM, MENU	Форма завершена с состоянием «OK» (т. е. ввод успешен или подтверждение ввода).
10	Все типы форм	Форма завершена с состоянием «TimeOut» (т. е. форма завершена потому что превышено время ожидания ввода).
20+<индекс в **scodes**>	Все типы форм	Форма завершена с помощью дополнительной клавиши завершения (смотри описание атрибута **scodes** в разделе 4.2).

3.4.2 Содержимое поля VALUES ответа

В поле VALUES ответа отображаются введенные данные и состояние объектов ввода.

Формат поля соответствует описанному в пункте 3.2.1.

Если отображалась форма обычного типа («FORM»), то как правило выводятся значения для всех типов отображаемых объектов имеющих атрибут «name» (т. е. для всех именованных отображаемых объектов). Тип значения переменной зависит от типа отображаемого объекта и описано в разделе посвященному соответствующему отображаемому объекту.

Описание содержимого поля VALUES в ответах для меню и форм специальных типов смотрите в соответствующих разделах.

4 Формат описания форм и отображаемых объектов форм

Шаблон описания формы:

```
<[FORM:ТИП ФОРМЫ] [атрибут1]="[значение]" … [атрибут n]="[значение]">
  <[TEXT:текст]>[заголовок формы]</TEXT>
  <[IMAGE:изображение]>[иконка отображаемая в заголовке]</IMAGE>
  <[ТИП ОТОБРАЖАЕМОГО ОБЪЕКТА 1]></[ТИП ОТОБРАЖАЕМОГО ОБЪЕКТА 1]>
     …
  <[ТИП ОТОБРАЖАЕМОГО ОБЪЕКТА n]></[ТИП ОТОБРАЖАЕМОГО ОБЪЕКТА n]>
</FORM>
```

Шаблон описания отображаемого объекта (на примере объекта «кнопка»):

```
<[BUTTON:ТИП ОБЪЕКТА] [атрибут1]="[значение]" … [атрибут n]="[значение]">
  <TEXT>[текст кнопки]</TEXT>
  <IMAGE>[иконка кнопки]</IMAGE>
</BUTTON>
```

Внимание: Типы форм, отображаемых объектов и примитивов всегда набираются в верхнем регистре.
Внимание: Атрибуты форм и отображаемых объектов всегда набираются в нижнем регистре.

Атрибуты могут иметь следующие типы:

- текстовое — латиница, только буквы цифры и символы разрешенные в xml для атрибута. Например `name="BUTTON_OK"`;
- числовое — число в десятичном или шестнадцатеричном представлении (начинается с символа 'x' – латинское). Например `options="x6002"`;
- перечисляемый — несколько значений простого типа разделенных символом разделителя (в качестве разделителя используется символ запятая: ',');
- цветовое — значения компонентов RGB цвета разделенные запятой (перечисление числового типа). Например `color="xFF,0,xFF" backcolor="127,0,0"`;
- координаты объекта — значения в пикселях горизонтали (верхняя левая точка объекта), вертикали (верхняя левая точка объекта), ширины и высоты объекта, разделенные запятыми (перечисление числового типа). Например `frame="10,10,220,20".`

4.1 Формат описания примитивов

Примитивы являются минимальными компонентами-контейнерами и могут быть частями форм и отображаемых объектов.

На текущий момент поддерживается несколько видов примитивов:

- **TEXT** – текстовый примитив, для текстовых значений отображаемых объектов и/или заголовков форм. В случае использования кириллицы нужно использовать UTF8 кодировку;
- **IMAGE, BACKIMAGE, ACTIVEBACKIMAGE** – графический примитив, контейнер изображений.

4.1.1 Формат описания примитива контейнер графики (IMAGE)

Примитив контейнер графики поддерживает следующие типы графики: BMP (1bit,4bit,8bit,24bit без компрессии), JPG, PNG, GIF, монохромные без компрессии (внутренний формат).

Внимание: Изображение может передаваться либо непосредственно в запросе (кодированные в BASE64), либо быть предварительно загруженными в терминал, как дополнительные данные к приложению (в этом случае в запросе передается имя файла изображения).

Имя примитива:

- **IMAGE** – изображение;
- **BACKIMAGE** – фоновое изображение (в пассивном режиме отображаемого объекта);
- **ACTIVEBACKIMAGE** – фоновое изображение (в активном режиме отображаемого объекта).

Примитив контейнер графики может иметь следующие атрибуты:

- **type** – тип источника графики, может иметь следующие значения (если не указано, то подразумевается тип **file**):
  - **file** – в качестве источника указывается имя файла (файлы с графикой содержатся в подкаталоге res терминала);
  - **base64 -** в качестве источника указывается данные в соответствующем формате закодированные BASE64.
- **ext** – расширение формата исходного файла, используется только в случае типа **base64.** По умолчанию: монохромные без компрессии (внутренний формат). Может принимать следующие значения: **bmp**, **jpg**, **png**.
- **transparent** – прозрачный цвет (цветовой тип), поддерживается для форматов **bmp, png**.

Примеры использования:

```
<IMAGE type="base64" ext="bmp" transparent="0,0,0">
  Qk1uAAAAAAAAAD4AAAAoAAAAEgAAAAwAAAABAAEAAAAAADAAAADEDgAAxA4AAAAAAAAAAAAA
  AAD/AAAAAAAAAEAAf/8AAHfvgAB734AAfb+AAH5/gAB+f4AAfb+AAHvfgAB374AAf/8AAAA
  QAA=
</IMAGE>
Тип источника: BASE64 кодированные данные;
Расширение формата файла: bmp – битмап;
Прозрачный цвет: черный (0,0,0).

<BACKIMAGE type="file">fon.jpg</BACKIMAGE>
Тип источника: файл.
```

Для описания простейших монохромных изображений, используется формат описанный ниже.
Внимание: Этот формат не предусматривает хранение цвета, рисуются точки изображения тем цветом, который установлен для объекта отображения (если бит точки равен 1), и фоновым цветом или прозрачным (если бит точки равен 0).
Структура файла/данных формата монохромное изображение без компрессии:
<1 байт: ширина изображения>
<2 байт: высота изображения>

<(ширина/8+ (((ширина%8)== 0)?0:1))*высота байт: данные изображения>
Пример изображения (данные приведены побайтно в шестнадцатеричном виде):
`0A 0A 0C 3F 1E 3F 3F 3F 7F BF ED FF CC FF 0C 3F 0C 3F 0C 3F 0C 3F`
Ширина изображения: 0A – 10 пикселей.
Высота изображения: 0A – 10 пикселей.
Байт на строку: (10/8 + (((10%8)==0)?0:1)) = 2 байт.
Всего байт на изображение: 2*10 = 20байт.
Расшифровка содержимого изображения:

Байты	**Биты изображения**										**Неиспользуемые биты**					
	1 байт в строке								2 байт в строке							
0C 3F	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	1
1E 3F	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1	1
3F 3F	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1
7F BF	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
ED FF	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
CC FF	1	1	0	0	1	1	0	0	1	1	1	1	1	1	1	1
0C 3F	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	1
0C 3F	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	1
0C 3F	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	1
0C 3F	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	1

4.2 Формат описания формы (FORM)

Тип формы имеет значение: **FORM** .

Форма может иметь следующие атрибуты:

- **options** – устанавливаемые опции формы (числовой тип: маска устанавливаемых опций);
- **clearoptions** – сбрасываемые опции формы (числовой тип: маска сбрасываемых опций);
- **font** – индекс шрифта заголовка (числовой тип);
- **height** – высота заголовка формы в пикселях (числовой тип);
- **timeout** – таймаут вывода формы в милисекундах (числовой тип);
- **color** – цвет элементов формы — например рамки (цветовой тип);
- **backcolor** - цвет фона формы (цветовой тип);
- **hcolor** — цвет заголовка формы (цветовой тип);
- **hbackcolor** – цвет фона заголовка формы (цветовой тип);
- **ccodes** – клавиши управления формой (перечисляемый числовой тип). Используются четыре кода клавиши (коды клавиш описаны в разделе 7.1) в следующем порядке: клавиша подтверждения (OK), клавиша отмены, клавиша выбора следующего элемента ввода, клавиша выбора предыдущего элемента ввода. Например: ccodes="13,x1B,x6C,x67" (такая комбинация установлена по умолчанию);
- **scodes** – дополнительные клавиши завершения работы формы (перечисляемый числовой тип). Используются для прекращения работы формы при нажатии

дополнительной клавиши заданной в этом перечислении (коды клавиш описаны в разделе 7.1), при этом в тэге STATE (смотри раздел 3.4.1) ответа будет возвращено значение 20+<индекс клавиши в перечислении>. Например: scodes=”8,x8B” (клавиша корректировки и клавиша меню, при нажатии клавиши корректировки будет возвращен в STATE значение 20, при нажатии клавиши меню — значение 21);

- **barscan** – значение «on» инициализирует сканер штрихкодов, значение «off» выключает сканер штрихкодов. Этот атрибут используется для работы со сканером в поточном режиме (смотри раздел 5.1.1).

Значения опций формы:

Маска опции	**Действие, если опция установлена**
x0001	Форма имеет заголовок.
x0002	Нажатие кнопки OK на клавиатуре завершит форму со состоянием «OK» (значение STATE: 1). По умолчанию эта функция установлена.
x0004	Нажатие кнопки Cancel на клавиатуре завершит форму со состоянием «Отмена» (значение STATE: -1). По умолчанию эта функция установлена.
x0008	Иконка не рисуется в заголовке формы. По умолчанию рисуется.
x0010	Рамка не рисуется у формы. По умолчанию рисуется.
x0020	Форма рисуется непрозрачной (фон заливается фоновым цветом: атрибут backcolor). По умолчанию полупрозрачная.
x0040	Форма рисуется полностью прозрачной. По умолчанию полупрозрачная.

Форма может содержать следующие примитивы:

- **TEXT** – текст заголовка;
- **IMAGE** – изображение заголовка.

Форма может содержать следующие отображаемые объекты:

- **STATIC** - статический элемент отображения (смотри раздел 4.2.2);
- **BUTTON** - элемент кнопка (смотри раздел 4.2.3);
- **INPUT** - поле ввода (смотри раздел 4.2.4);
- **PATTERNINPUT** – поле ввода по шаблону (смотри раздел 4.2.5);
- **MULTILINE** – многострочное поле ввода/вывода (см. TODO ссылка);

Внимание: Отображаемые объекты выводятся в той очередности, как перечислены в описании формы. Если объекты перекрывают друг друга, то объект, описанный первым, будет выведен под объектом, описанным далее.

Пример описания формы:

```
<FORM>
  <TEXT>Static</TEXT>
  <STATIC frame="10,10,220,20" name="version"/>
  <STATIC frame="10,30,220,230">
    <IMAGE type="file">palma.jpg</IMAGE>
  </STATIC>
  <STATIC frame="10,30,220,20" justify="x01" border="1">
```

```
    <TEXT>Justify LEFT</TEXT>
  </STATIC>
  <STATIC frame="10,110,220,100">
    <IMAGE type="file">example.gif</IMAGE>
  </STATIC>
  <BUTTON frame="60,265,120,32" control="x0D">
    <TEXT>OK</TEXT>
    <IMAGE>ok.b16</IMAGE>
  </BUTTON>
</FORM>
```

4.2.1 Формат описания отображаемых объектов

Отображаемые объекты используются для построения содержимого форм и каждый тип объекта определяет свойственную ему функциональность. В этом разделе рассмотрены общие для отображаемых объектов функции.

Внимание: Объекты, которые могут иметь фокус ввода (например объект INPUT), бывают в пассивном (фокус ввода не принадлежит объекту) и активном состоянии (фокус ввода принадлежит объекту). Остальные объекты всегда находятся в пассивном состоянии (например объект STATIC) и соответственно для них действуют атрибуты и примитивы пассивного состояния.

Отображаемые объекты могут иметь следующие атрибуты:

- **frame** – координаты отображаемого объекта (отсчет идет от верхнего правого угла рабочего пространства формы);
- **name** – имя отображаемого объекта (имя может содержат только латинские буквы и цифры). Если установлено имя для отображаемого объекта, то для объекта можно установить значение и значение этого объекта будет возвращено в ответе, после диалога с пользователем;
- **font** – индекс шрифта отображаемого объекта (числовой тип);
- **justify** – тип выравнивания для отображаемого объекта (числовой тип);
- **transparency** – прозрачность фона отображаемого объекта в пассивном режиме, т. е. когда фокус ввода не принадлежит объекту (числовой тип: 0 — непрозрачный, 1 — прозрачный);
- **activetransparency** – прозрачность фона отображаемого объекта в активном режиме, т. е. когда фокус ввода принадлежит объекту (числовой тип: 0 — непрозрачный, 1 — прозрачный);
- **border** - рамка отображаемого объекта в пассивном режиме, т. е. когда фокус ввода не принадлежит объекту (числовой тип: 0 — рамки нет, 1 — рамка есть);
- **activeborder** - рамка отображаемого объекта в активном режиме, т. е. когда фокус ввода принадлежит объекту (числовой тип: 0 — рамки нет, 1 — рамка есть);
- **color** – цвет элементов отображаемого объекта в пассивном режиме, т. е. когда фокус ввода не принадлежит объекту (цветовой тип);
- **back** - цвет фона отображаемого объекта в пассивном режиме, т. е. когда фокус ввода не принадлежит объекту (цветовой тип);
- **activecolor** – цвет элементов отображаемого объекта в активном режиме, т. е. когда фокус ввода принадлежит объекту (цветовой тип);

- **activeback** - цвет фона отображаемого объекта в активном режиме, т. е. когда фокус ввода принадлежит объекту (цветовой тип).

Возможные значения атрибута **justify**:

Маска атрибута	Действие, если атрибут установлен
x01	Выравнивание влево.
x02	Выравнивание вправо.
0	Выравнивание по центру.
x04	Выравнивание вверх.
x08	Выравнивание вниз.

Любой объект может содержать следующие примитивы:

- **BACKIMAGE** – фоновое изображение в пассивном режиме;
- **ACTIVEBACKIMAGE** – фоновое изображение в активном режиме.

Формат значения поля **VALUES** ответа:

```
<VALUES>
  <[<name>:имя отображаемого объекта]>
    <[<primitive>:тип примитива]>
      [возвращаемое значение]
    </[<primitive>:тип примитива]>
  </[<name>:имя отображаемого объекта]>
</VALUES>
```

Если несколько отображаемых объектов должны вернуть значение в ответе, то будет несколько полей с именами отображаемых объектов.

Например:

```
<VALUES>
  <summa>
    <TEXT>100000</TEXT>
  </summa>
</VALUES>
```

4.2.2 Формат описания объекта «статичный текст и/или изображение» (STATIC)

Объект предназначен для отображения статичного текста или/и изображения. Объект не принимает фокуса ввода и не может быть в активном состоянии.

Значения по умолчанию для объекта STATIC:

Атрибут	Значение	Описание
font	1 (средний)	Индекс шрифта
justify	0 (по центру)	Тип выравнивания
transparency	1 (фон прозрачный)	Прозрачность фона в пассивном режиме
border	0 (без рамки)	Рамка

Объект STATIC может содержать следующие примитивы:

- **TEXT** – текст;

- **IMAGE** – изображение.

Если заданы оба примитива, то одновременно выводится текст и изображение.

Пример описания объекта в форме:

```
<STATIC frame="10,30,220,230">
  <IMAGE type="file">palma.jpg</IMAGE>
</STATIC>
<STATIC frame="10,30,220,20" justify="x01" border="1" name="justify">
  <TEXT>Justify LEFT</TEXT>
</STATIC>
```

Если для объекта STATIC задано имя, то в ответе возвращается содержимое примитива TEXT.

Пример ответа:

```
<VALUES>
  <justify>
    <TEXT>Justify LEFT</TEXT>
  </justify>
</VALUES>
```

4.2.3 Формат описания объекта «кнопка» (BUTTON)

Объект предназначен для совершения действий или ввода/выбора значений в управляемый объект. Объект принимает активное состояние в случае касание области его расположения на тачскрине.

Дополнительные атрибуты для объекта BUTTON:

- **control** – код клавиши, передающийся форме при нажатии кнопки, смотри раздел 7.1 (числовой тип);
- **accelcode** – код клавиши, при нажатии которой отрабатывает функционал кнопки, смотри раздел 7.1 (числовой тип);
- **options** - устанавливаемые опции объекта BUTTON (числовой тип: маска устанавливаемых опций);
- **clearoptions** – сбрасываемые опции объекта BUTTON (числовой тип: маска сбрасываемых опций);
- **slave** – ссылка на управляемый объект (значением атрибута является имя [атрибут **name**] управляемого объекта).

Внимание: Управляемый объект должен быть определен в описании формы раньше, чем объект BUTTON, в котором есть ссылка на него.

Внимание: Если атрибуту **control** задан код клавиши «Отмена» [x1B] – то при нажатии ее форма будет завершена с состоянием «Отмена» (смотри раздел 3.4.1). Если атрибуту **control** задан код клавиши «OK» [x0D] – то при нажатии ее форма будет завершена с состоянием «OK» (смотри раздел 3.4.1)

Значения опций объекта BUTTON:

Маска опции	**Действие, если опция установлена**
x0001	Код кнопки не передается форме, если опция установлена. По умолчанию опция не установлена.
x0002	Кнопка рисуется с тенью, если опция установлена. По умолчанию эта

	опция установлена.

Значения по умолчанию для объекта BUTTON:

Атрибут	**Значение**	**Описание**
font	1 (средний)	Индекс шрифта
justify	0 (по центру)	Тип выравнивания
transparency	0 (фон не прозрачный)	Прозрачность фона в пассивном режиме
border	0 (без рамки)	Рамка

Объект BUTTON может содержать следующие примитивы:

- **TEXT** – текст;
- **IMAGE** – изображение;
- **VALUE –** текстовое значение передаваемое управляемому объекту в случае нажатия кнопки.

Если заданы оба примитива TEXT и IMAGE, то одновременно выводится текст и изображение.

Пример описания объекта в форме:

```
<BUTTON frame="2,225,74,50" control="x0D" slave="summa">
  <TEXT>500</TEXT>
  <VALUE>50000</VALUE>
</BUTTON>
```

Объект BUTTON не возвращает никакого ответа, даже если задано для него имя.

4.2.4 Формат описания объекта «поле ввода» (INPUT)

Объект предназначен для ввода числовых или текстовых значений. Объект может принимать фокус ввода.

Внимание: Ввод осуществляется для поля только если поле ввода находится в активном режиме, т. е. когда фокус ввода принадлежит этому полю.

Дополнительные атрибуты для объекта INPUT:

- **length** – максимальная длина вводимых данных (числовой тип), по умолчанию равно 32;
- **options** - устанавливаемые опции объекта (числовой тип: маска устанавливаемых опций);
- **radix** – количество чисел после запятой (числовой тип);
- **extfont** – индекс шрифта дробной части числа (числовой тип).

Значения опций объекта INPUT:

Маска опции	**Действие, если опция установлена**
x0001	Поле ввода предназначена для ввода числа.
x0002	Поле ввода предназначено для ввода шестнадцатеричных значений.
x0004	Вывод подсказки со значениями символов при вводе значения в телефонном режиме (когда при многократном нажатии клавиши

	вводится соответствующий символ).
x0008	При вводе игнорируется точка, т. е. число вводится последовательно от крайней правой позиции (только для объекта INPUT).
x0010	Скрывать вводимые символы при вводе (режим ввода пароля).
x0020	Курсор моргает.
x0040	Область шаблона, прозрачная (только для объекта PATTERNINPUT).
x0080	Допустимо вводить лидирующие нули.
x0100	Разрешено вводить минус для числа.
x0200	При достижении максимальной длины не вводится последний символ (т. е. его предварительно надо удалить, а потом вводить).

Значения по умолчанию для объекта INPUT:

Атрибут	**Значение**	**Описание**
font	1 (средний)	Индекс шрифта
justify	0 (по центру)	Тип выравнивания
transparency	0 (фон не прозрачный)	Прозрачность фона в пассивном режиме
border	0 (без рамки)	Рамка в пассивном режиме
activetransparency	0 (фон не прозрачный)	Прозрачность фона в активном режиме
activeborder	0 (без рамки)	Рамка в активном режиме

Объект INPUT может содержать следующие примитивы:

- **TEXT** – текст, начальное значение.

Внимание: Значение вводимого числа передается и возвращается из поля ввода без учета дробной части (т. е. не должно содержать дробную часть, дробная часть задается атрибутом radix).

Пример описания объекта в форме:

```
<INPUT frame="20,50,198,40" maxlen="8" options="1" radix="2" font="5" extfont="4"
name="summa"/>
```

Если для объекта **INPUT** задано имя, то в ответе возвращается содержимое примитива **TEXT**.

Пример ответа:

```
<VALUES>
  <summa>
    <TEXT>100000</TEXT>
  </summa>
</VALUES>
```

4.2.5 Формат описания объекта «поле ввода по шаблону» (PATTERNINPUT)

Объект предназначен для ввода числовых или текстовых значений с использованием шаблона. Объект может принимать фокус ввода. Ввод осуществляется для поля только если поле ввода находится в активном состоянии.

Дополнительные атрибуты для объекта **PATTERNINPUT**:

- **options** - устанавливаемые опции объекта (числовой тип: маска устанавливаемых опций);
- **fillsymbol** – символ, который используется в качестве заполнителя в еще не введенных позициях, по умолчанию пробел;
- **fillsymbolcode** – код символа, который используется в качестве заполнителя в еще не введенных позициях (числовой тип), по умолчанию пробел.

Значения опций объекта **PATTERNINPUT**:

Маска опции	**Действие, если опция установлена**
x0001	Поле ввода предназначена для ввода числа.
x0002	Поле ввода предназначено для ввода шестнадцатеричных значений.
x0004	Вывод подсказки со значениями символов при вводе значения в телефонном режиме (когда при многократном нажатии клавиши вводится соответствующий символ).
x0008	При вводе игнорируется точка, т. е. число вводится последовательно от крайней правой позиции (только для объекта **INPUT**).
x0010	Скрывать вводимые символы при вводе (режим ввода пароля).
x0020	Курсор моргает.
x0040	Область шаблона, прозрачная (только для объекта **PATTERNINPUT**).
x0080	Допустимо вводить лидирующие нули.
x0100	Разрешено вводить минус для числа.
x0200	При достижении максимальной длины не вводится последний символ (т. е. его предварительно надо удалить, а потом вводить).

Значения по умолчанию для объекта PATTERNINPUT:

Атрибут	**Значение**	**Описание**
font	1 (средний)	Индекс шрифта
justify	0 (по центру)	Тип выравнивания
transparency	0 (фон не прозрачный)	Прозрачность фона в пассивном режиме
border	0 (без рамки)	Рамка в пассивном режиме
activetransparency	0 (фон не прозрачный)	Прозрачность фона в активном режиме
activeborder	0 (без рамки)	Рамка в активном режиме

Объект **INPUT** может содержать следующие примитивы:

- **PATTERN** – шаблон, текстовая строка в которой символ '#' означает места где будут вводиться символы;
- **TEXT** – текст, начальное значение.

Внимание: Значение вводимой строки передается и возвращается из поля без учета шаблона (т. е. содержит только те символы которые были введены).

Пример описания объекта в форме:

```
<PATTERNINPUT frame="2,50,234,40" options="3" fillsymbol="." name="key">
 <PATTERN>#### #### #### ####
 #### #### #### ####
 </PATTERN>
</PATTERNINPUT>
```

Если для объекта **PATTERNINPUT** задано имя, то в ответе возвращается содержимое примитива **TEXT**.

Пример ответа:

```
<VALUES>
 <key>
  <TEXT>1237545ABCDEF00851986542139BDF08</TEXT>
 </key>
</VALUES>
```

4.3 Формат описания меню (MENU)

Тип формы для меню имеет значение: **MENU** .

Меню всегда возвращает стандартизированное значение в поле ответа **VALUES**, соответствующее выбранному пункту в процессе вывода диалога.

Меню может иметь все атрибуты присущие форме, но к тому же может иметь собственные атрибуты:

- **type** – тип меню (числовой тип: 0 — меню-список, 1 — меню иконного типа);
- **rowdelta** – дополнительная высота строки в пикселях (числовой тип);
- **accelfont** – индекс шрифта для цифр «акселератора» (числовой тип);
- **columns** – количество колонок для меню иконного типа (числовой тип);
- **itransparency** – прозрачность для пассивных пунктов меню (числовой тип: 0 – непрозрачный, 1 - прозрачный);
- **iactivetransparency** - прозрачность для активного пункта меню (числовой тип: 0 – непрозрачный, 1 - прозрачный);
- **iborder** — рамка для пассивных пунктов меню (числовой тип: 0 – рамки нет, 1 — рамка есть);
- **iactiveborder** – рамка для активных пунктов меню (числовой тип: 0 – рамки нет, 1 — рамка есть).

Внимание: Для **MENU** атрибут **ccodes** в отличие от **FORM** означает другие управляющие клавиши. Используются четыре кода клавиши (коды клавиш описаны в разделе 7.1) в следующем порядке: клавиша перемещения на пункт меню вверх, клавиша перемещения на пункт меню вниз, клавиша перемещения на страницу вверх, клавиша перемещения на страницу вниз.

Дополнительные значения опций (атрибут **options**) для меню:

Маска опции	**Действие, если опция установлена**
x0100	Запрещает прокрутку меню с использованием сенсорного экрана (тачпада)
x0200	Включает циклический выбор пунктов меню.

x1000	При выводе меню первый пункт будет иметь статус выбранного. По умолчанию эта функцию установлена.
x2000	Включается режим «акселерации», т. е. если пунктов меню меньше 9 то каждой цифровой кнопке терминала соответствует пункт меню. По умолчанию эта функция не установлена.
x4000	В случае режима «акселерации», при установке этой опции у каждого пункта меню рисуется цифра номера. По умолчанию эта функция не установлена.
x8000	Завершение работы меню только по клавише “OK” клавиатуры (зеленая).

Меню может содержать следующие примитивы:

- **TEXT** – текст заголовка;
- **IMAGE** – изображение заголовка.

Меню может содержать некоторое количество описывающее пунктов меню: **ITEM** (смотрите раздел 4.3.1).

Формат поля **VALUES** в возвращаемом ответе. Поле содержит три поля:

- **SELECTED** – номер выбранного пункта меню (перечисление начинается с 1);
- **TEXT** – текст выбранного пункта меню;
- **VALUE** – значение пункта меню.

Пример описания меню:

```
<MENU height="20" options="x6009" type="0">
  <TEXT>МЕНЮ ICONS</TEXT>
  <ITEM value="201">
    <TEXT>ИТЕМ 1</TEXT>
    <IMAGE>icon1.jpg</IMAGE>
  </ITEM>
  <ITEM value="202">
    <TEXT>ИТЕМ 2</TEXT>
    <IMAGE>icon2.jpg</IMAGE>
  </ITEM>
  <ITEM value="203">
    <TEXT>ИТЕМ 3</TEXT>
    <IMAGE>icon3.jpg</IMAGE>
  </ITEM>
  <ITEM value="204">
    <TEXT>ИТЕМ 4</TEXT>
    <IMAGE>icon4.jpg</IMAGE>
  </ITEM>
  <ITEM value="205">
    <TEXT>ИТЕМ 5</TEXT>
    <IMAGE>icon5.jpg</IMAGE>
  </ITEM>
</MENU>
```

Пример возвращаемого значения при выборе 3 пункта меню:

```
<VALUES>
  <SELECTED>3</SELECTED>
  <TEXT>ИТЕМ 3</TEXT>
  <VALUE>203</VALUE>
</VALUES>
```

4.3.1 Формат описания пунктов меню (ITEM)

Описание пункта меню соответствует шаблону описания отображаемого элемента.

Пункт меню может иметь следующие атрибуты:

- **name** — имя пункта меню, как отображаемого элемента;
- **select** – определяет выбранность пункта меню по умолчанию (числовое значение: 0 — не выбран, 1 — выбран). Рекомендуется ставить этот атрибут только одному пункту, который будет при начале показа меню выбранным;
- **code** – код;
- **value** – значение пункта меню (числовое значение);
- **font** - индекс шрифта пункта меню (числовой тип).

Пункт меню может содержать следующие примитивы:

- **TEXT** – текст пункта меню;
- **IMAGE** – изображение пункта меню.

5 Специальные типы форм

Специальные типы форм применяются для использования дополнительных возможностей оборудования терминала IRAS.

Сами формы и объекты отображения описываются так же как описано ранее для объекта FORM (смотри раздел 4). Но при этом используется дополнительное оборудование терминала IRAS.

На текущий момент используются следующие специальные типы форм:

- **FORM_BARCODE** – отображение формы с использованием сканера штрихкодов (чтения штрихкода) (смотри раздел 5.1);
- **FORM_MIFARE** – отображение формы с использованием бесконтактного ридера для чтения карт MIFARE[1] (смотри раздел 5.2);
- **FORM_MSR** – отображение формы с использованием считывателя магнтных карты (смотри раздел 5.3).

5.1 Использование формы для чтения штрихкода (FORM_BARCODE)

Описание формы для такого типа должно содержать ключевое слово **FORM_BARCODE** вместо слова **FORM**.

Внимание: Встроенный сканер штрихкодов имеет достаточно долгое время включения и выключения. При использовании этой формы (в режиме единичного сканирования), сканер включается и выключается при каждом вызове формы. Если в Вашем приложении требуется более быстрая реакция в режиме сканирования, рекомендуется использовать поточный способ работы со сканером штрихкодов (смотри раздел 5.1.1).

Формат запроса для такого типа форм никак не отличается от описания обычной формы, кроме использования ключевого слова.

Пример запроса:

```
<?xml version="1.0" encoding="utf-8"?>
<REQUEST>
  <ID>1</ID>
  <TD>1824 050916</TD>
  <FORM_BARCODE height="20" options="x0009">
    <TEXT>ВНИМАНИЕ</TEXT>
    <MULTILINE frame="2,2,234,220" options="1">
      <TEXT>ВВЕДИТЕ ШТРИХКОД</TEXT>
    </MULTILINE>
    <BUTTON frame="60,245,120,32" control="x1B">
      <TEXT>ОТМЕНА</TEXT>
    </BUTTON>
  </FORM_BARCODE>
  <VALUES />
</REQUEST>
```

Формат ответа отличается от ответа обычной формы, тем что в разделе **VALUES** добавляется подраздел с описанием ответа сканера штрихкода:

```
<BARCODE state="[состояние сканера]">[значение штрихкода]</BARCODE>
```

1 MIFARE – торговая марка компании NXP B.V. Тип бесконтактных карт.

Возможные значения атрибута state (состояния сканера):

- -1 – ошибка инициализации сканера;
- 0 — штрихкод не был прочитан;
- 1 — штрихкод успешно прочитан.

Внимание: Значение штрихкода возвращается только при значении state равным 1. При этом ответ формы возвращается как соответствующий нажатию кнопки ОК (<STATE>1</STATE>).

Внимание: Если терминал не оснащен сканером штрихкодов или он программно не поддержан, то форма не будет выводится, а будет возвращена ошибка обработки запроса с кодом 6 (смотри раздел 3.3).

Пример успешного ответа со значением штрихкода:

```
<?xml version="1.0" encoding="utf-8"?>
<RESPONSE>
  <ID>1</ID>
  <TD>2139 050916</TD>
  <ERROR>0</ERROR>
  <STATE>1</STATE>
  <VALUES>
    <BARCODE state="1">4607024895313</BARCODE>
  </VALUES>
</RESPONSE>
```

5.1.1 Использование поточного метода работы со сканером штрихкодов

В случае, когда задержки включения/выключения сканера штрихкодов, становятся критически долгими для функционирования приложения, рекомендуется использовать поточный способ сканирования.

Суть этого метода заключается в использовании следующий последовательности действий при работе со сканером штрихкодом:

1. Сканер штрихкодов включается предварительно при выводе произвольной формы (не обязательно использовать форму **FORM_BARCODE**) - с помощью атрибута формы **barscan**, установленного в значение «on» (смотри раздел 4.2);
2. Производится сканирование штрихкодов с помощью форм **FORM_BARCODE** (при этом при сканировании не происходит включение/выключение сканера штрихкодов, а только отрабатывает сама процедура сканирования штрихкода);
3. Сканер штрихкодов отключается после сканирования при выводе произвольной формы (не обязательно использовать форму **FORM_BARCODE**) - с помощью атрибута формы **barscan**, установленного в значение «off» (смотри раздел 4.2).

Внимание: Если не планируется использовать терминал в режиме работы от батареи, то рекомендуется инициализировать сканер сразу при старте приложения (с помощью атрибута формы **barscan**, установленного в значение «on») и не отключать его.

5.2 Использование формы для работы с картами MIFARE (FORM_MIFARE)

Описание формы для такого типа должно содержать ключевое слово **FORM_MIFARE** вместо слова **FORM**.

В разделе **VALUES** запроса должен быть подраздел **MIFARE**, описывающий планируемые операции с картой.

Внимание: На текущий момент поддерживаются операция чтения [R] и операция записи [W] на карту.

Формат раздела **MIFARE** в запросе:

```
<MIFARE>
  <SECTOR num="[номер сектора]" oper="[тип операции]"
          passa="[пароль типа A]" passb="[пароль типа B]">
    <BLOCK num="[номер блока в секторе]">[данные: hex]</BLOCK>
    ...
  </SECTOR>
  ...
</MIFARE>
```

Раздел **SECTOR** описывает сектор карты к которому будет применена операция и может содержать следующие атрибуты:

- **num** – целое число: номер сектора карты к которому будет применена операция (для карт MIFARE 1K определено 16 секторов начиная с 0). Обязательный атрибут;
- **oper** – R или W: тип операции - операция чтения [R] или операция записи [W] на карту. Обязательный атрибут;
- **passa** – 6 символов: пароль типа A доступа к сектору в виде строки. Необязательный атрибут — при операции используется passa или passb (один из указанных);
- **passb** – 6 символов: пароль типа B доступа к сектору в виде строки. Необязательный атрибут — при операции используется passa или passb (один из указанных);
- **passahex** – 12 символов: пароль типа A доступа к сектору в виде шестнадцатеричных данных. Необязательный атрибут — при операции используется passa или passb (один из указанных);
- **passbhex** – 12 символов: пароль типа B доступа к сектору в виде шестнадцатеричных данных. Необязательный атрибут — при операции используется passa или passb (один из указанных).

Раздел **BLOCK** указывается только при операции записи данных на карту. С помощью него описывается номер блока в секторе на запись и данные для записи. Данные для записи указываются в виде шестнадцатеричной строки (длиной 32 символа — что описывает 16 байт данных для записи).

Внимание: При операции чтение с карты, считываются все три блока сектора. Блок с данными доступа к сектору (3ий блок) не считывается.

Пример описания сектора для операции чтение:

```
<SECTOR num="1" oper="R" passa="PASS01" />
```

(считать сектор 1, пароль типа A для чтения в виде строки «PASS01»).

Пример описания сектора для операции запись:

```
<SECTOR num="2" oper="W" passb="PASS02">
  <BLOCK num="0">0102030405060708090A0B0C0D0E0F00</BLOCK>
  <BLOCK num="1">00000000000000000000000000000000</BLOCK>
  <BLOCK num="2">00000000000000000000000000000000</BLOCK>
</SECTOR>
```

(записать сектор 2 блоки 0..2, пароль типа B для записи в виде строки «PASS02»).

Пример запроса для чтения нескольких секторов карты MIFARE:

```
<?xml version="1.0" encoding="utf-8"?>
<REQUEST>
  <ID>1</ID>
  <TD></TD>
  <FORM_MIFARE height="20" options="x0009">
    <TEXT>ВНИМАНИЕ</TEXT>
    <MULTILINE frame="2,2,234,220" options="1">
      <TEXT>ПРИСЛОНИТЕ КАРТУ MIFARE</TEXT>
    </MULTILINE>
    <BUTTON frame="60,245,120,32" control="x1B">
      <TEXT>ОТМЕНА</TEXT>
    </BUTTON>
  </FORM_MIFARE>
  <VALUES>
    <MIFARE>
      <SECTOR num="1" oper="R" passa="PASS01" />
      <SECTOR num="2" oper="R" passahex="4D4153544552" />
      <SECTOR num="3" oper="R" passa="PASS03" />
      <SECTOR num="4" oper="R" passa="PASS04" />
    </MIFARE>
  </VALUES>
</REQUEST>
```

Формат ответа отличается от ответа обычной формы, тем что в разделе **VALUES** добавляется подраздел **MIFARE** с описанием ответа с данными о примененной операции к карте.

Внимание: Раздел **MIFARE** появляется в ответе, только если операция была применена к карте. Если форма была завершена без предъявления карты, то этого раздела не будет в ответе.

Формат раздела **MIFARE** ответа:

```
<MIFARE uid="[идентификатор карты: hex]">
  <SECTOR num="[номер сектора]" oper="[тип операции]">
    <BLOCK num="[номер блока в секторе]"
          err="[код ошибки]" errext="[дополнительный код  ошибки]"
>[данные: hex]</BLOCK>
    ...
  </SECTOR>
  ...
</MIFARE>
```

Раздел **MIFARE** содержит следующие атрибуты:

- **uid** – шестнадцатеричная строка [8 символов]: уникальный идентификатор карты, над которой проводилась операция.

Раздел **SECTOR** описывает сектор карты к которому была применена операция и может содержать следующие атрибуты:

- **num** - целое число: номер сектора карты к которому была применена операция. Обязательный атрибут;
- **oper** – R или W: тип операции - операция чтения [R] или операция записи [W] на карту. Обязательный атрибут.

Раздел **BLOCK** описывает блок данных прочитанных или записанных на карту. Данные выводятся в виде шестнадцатеричной строки (длиной 32 символа — что описывает 16 байт

данных для записи). Данные блока появляются только при успешной операции или чтения на карту. Раздел может иметь следующие атрибуты:

- **num** - целое число: номер сектора карты к которому была применена операция. Обязательный атрибут;
- **err** – отрицательное число: код ошибки при применении операции. Присутствует только если операция была не успешна;
- **errext** – число: дополнительный код ошибки. Присутствует только если операция была не успешна.

Внимание: При всех успешных операциях с картой возвращается значение формы как при нажатии кнопки ОК (<STATE>1</STATE>), при возникновении хоть одного сбоя при операции с картой возвращается значение формы как при нажатии кнопки ОТМЕНА (<STATE>-1</STATE>).

Внимание: Если терминал не смог инициализировать бесконтактный ридер , то форма не будет выводится, а будет возвращена ошибка обработки запроса с кодом 7 (смотри раздел 3.3).

Пример успешного ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<RESPONSE>
  <ID>1</ID>
  <TD>2226 050916</TD>
  <ERROR>0</ERROR>
  <STATE>1</STATE>
  <VALUES>
    <MIFARE uid="39553FB4">
      <SECTOR num="1" oper="R">
        <BLOCK num="0">41314645535441324645535441334645</BLOCK>
        <BLOCK num="1">53544134464553540000000000000000</BLOCK>
        <BLOCK num="2">00000000000000000000000000000000</BLOCK>
      </SECTOR>
      <SECTOR num="2" oper="R">
        <BLOCK num="0">00000000000000000000000000000000</BLOCK>
        <BLOCK num="1">00000000000000000000000000000000</BLOCK>
        <BLOCK num="2">00000000000000000000000000000000</BLOCK>
      </SECTOR>
     </MIFARE>
  </VALUES>
</RESPONSE>
```

Пример ответа со сбоем:

```
<?xml version="1.0" encoding="utf-8"?>
<RESPONSE>
  <ID>1</ID>
  <TD>2305 050916</TD>
  <ERROR>0</ERROR>
  <STATE>-1</STATE>
  <VALUES>
    <MIFARE uid="39553FB4">
      <SECTOR num="1" oper="R">
        <BLOCK num="0">41314645535441324645535441334645</BLOCK>
        <BLOCK num="1">53544134464553540000000000000000</BLOCK>
        <BLOCK num="2">00000000000000000000000000000000</BLOCK>
      </SECTOR>
      <SECTOR num="2" oper="R">
        <BLOCK num="0">00000000000000000000000000000000</BLOCK>
```

```
        <BLOCK num="1">00000000000000000000000000000000</BLOCK>
        <BLOCK num="2" err="-4" errext="-2903" />
      </SECTOR>
    </MIFARE>
  </VALUES>
</RESPONSE>
```

5.2.1 Коды ошибок при работе с картами MIFARE

Дополнительные коды ошибок являются системными кодами считывателя бесконтактных карт и могут иметь следующие значения:

Код	Значение
-2901	Parity error. Ошибка проверки четности.
-2902	CRC error. Ошибка проверка циклического кода.
-2903	Timeout or no card. Таймаут ответа карты или ее отсутствие.
-2904	Multiple cards collision. Коллизия от присутствия нескольких карт.
-2905	Frame format error. Ошибка формата кадра протокола обмена.
-2906	Interference. Интерференция сигнала (сильные помехи).
-2907	Chip error, fail to communicate correctly. Ошибка ответа чипа карты.
-2908	M1 authentication error. Ошибка аутентификации.
-2909	Transmission error. Ошибка отправки ответа карте.
-2910	Protocol error. Ошибка протокола обмена.

5.3 Использование формы для работы со считывателем магнитных карт (FORM_MSR)

Описание формы для такого типа должно содержать ключевое слово **FORM_MSR** вместо слова **FORM**.

Внимание: При чтении магнитной карты, считываются все три трека карты.

Пример запроса для чтения магнитной карты:

```
<?xml version="1.0" encoding="utf-8"?>
<REQUEST>
  <ID>1</ID>
  <TD></TD>
  <FORM_MSR height="20" options="x0009">
    <TEXT>ВНИМАНИЕ</TEXT>
    <MULTILINE frame="2,2,234,220" options="1">
```

```
      <TEXT>СЧИТЫВАЙТЕ МАГНИТНУЮ КАРТУ</TEXT>
    </MULTILINE>
    <BUTTON frame="60,245,120,32" control="x1B">
      <TEXT>ОТМЕНА</TEXT>
    </BUTTON>
  </FORM_MSR>
  <VALUES>
  </VALUES>
</REQUEST>
```

Формат ответа отличается от ответа обычной формы, тем что в разделе **VALUES** добавляется подраздел **MSR** с описанием ответа с данными считанными с магнитной карты.

Внимание: Раздел **MSR** появляется в ответе, только если магнитная карта была считана. Если форма была завершена без предъявления карты, то этого раздела не будет в ответе. Формат раздела **MSR** ответа:

```
<MSR err="[ошибка функции считывания карты]">
  <TRACK1 err="[ошибка считывания трека]">[данные трека]</TRACK1>
  <TRACK2 err="[ошибка считывания трека]">[данные трека]</TRACK2>
  <TRACK3 err="[ошибка считывания трека]">[данные трека]</TRACK3>
</MSR>
```

Раздел **MSR** может содержать следующий атрибут:

- **err** – десятичное число: код ошибки считывания карты (атрибут отсутсвует если ошибок нет).

Разделы **TRACK1 … TRACK3** описывают данные трека считанного с карты и содержат следующий атрибут:

- **err** – десятичное число: код ошибки считывания трека (0 — трек считан успешно).

Раздел **TRACK[n]** описывает данные трека прочитанные с магнитной карты. Данные выводятся в виде текстовой строки. Данные трека появляются только при успешной операции или чтения трека карты.

Внимание: При всех успешных операциях с картой возвращается значение формы как при нажатии кнопки ОК (<STATE>1</STATE>), при возникновении хоть одного сбоя при операции чтения карты возвращается значение формы как при нажатии кнопки ОТМЕНА (<STATE>-1</STATE>).

Внимание: Если терминал не смог инициализировать считыватель магнитных карт, то форма не будет выводится, а будет возвращена ошибка обработки запроса с кодом 8 (смотри раздел 3.3).

Пример успешного ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<RESPONSE>
  <ID>1</ID>
  <TD>1642 050917</TD>
  <BT>8292</BT>
  <ERROR>0</ERROR>
  <STATE>1</STATE>
  <VALUES>
    <MSR>
  <TRACK1 err="0">B5338461157764818^SUPPLIED/NOT^14102260713702860000000</TRACK1>
  <TRACK2 err="0">5338461157764818=14102260713702860000</TRACK2>
  <TRACK3 err="0" />
    </MSR>
```

```
  </VALUES>
</RESPONSE>
```

Пример ответа со сбоем:

```
<?xml version="1.0" encoding="utf-8"?>
<RESPONSE>
  <ID>1</ID>
  <TD>1647 050917</TD>
  <BT>8292</BT>
  <ERROR>0</ERROR>
  <STATE>-1</STATE>
  <VALUES>
    <MSR>
      <TRACK1 err="-2703" />
      <TRACK2 err="-2703" />
      <TRACK3 err="0" />
    </MSR>
  </VALUES>
</RESPONSE>
```

5.3.1 Коды ошибок при работе со считывателем магнитных карт

Дополнительные коды ошибок являются системными кодами считывателя бесконтактных карт и могут иметь следующие значения:

Код	**Значение**
-2701	Fail to operate MSR. Ошибка работы с магнитным считывателем.
-2702	Head mark is not found. Метка начала трека не найдена.
-2703	End mark is not found. Метка конца трека не найдена.
-2704	LRC check error. Не совпадает проверочное значение.
-2705	Check error of a certain bit of MSR. Ошибка проверки установленных флагов считывателя.
-2706	No card is swiped. Нет информации об считанных картах.
-2707	No data in Mag card. Не найдены данные на магнитной карте.
-2708	Magnetic stripe card format error. Ошибка формата данных магнитной карты.

6 Работа с модулем обслуживания платежных карт

Структура запроса/ответа полностью соответствует описанному в пункте 3.

Внимание: Модуль обслуживания платежных карт является внешним модулем и работает автономно от приложения. Поэтому надо учитывать, что во время его работы, основная программа находится в ждущем режиме и не отслеживает состояние терминала и соединения.

6.1 Формат запроса «ПРОВЕДЕНИЕ КАРТОЧНОЙ ОПЕРАЦИИ» к модулю обслуживания платежных карт

Структура запроса:

`<XML ...>` - заголовок XML документа
`<REQUEST>` - начало данных запроса
`<ID>[идентификатор запроса (число: unsigned int)]</ID>`
`<TD>[время/дата]</TD>`
`<PAYCARD>`
`<TYPE>OPERATION</TYPE>` - тип запроса
`<PROTO>[тип протокола]</PROTO>` - тип протокола
`<PACKET>[содержимое запроса]</PACKET>` - пакет запроса
`</PAYCARD>`
`<VALUES>[значения полей запроса]</VALUES>`
`</REQUEST>` - конец данных запроса
`[0x00]` – стоповый символ

На текущий момент поддерживаются следующие типы протоколов (поле **PROTO**):

- SA – Smart Access.

6.1.1 Формат поля PACKET запроса (протокол SA)

```
<PACKET>
  <FIELD code="[код поля:число]" name="имя поля">[значение
поля]</FIELD>
  ...
  <FIELD code="[код поля:число]" name="имя поля">[значение
поля]</FIELD>
</PACKET>
```

Атрибуты поля FIELD:

- code – числовой код поля запроса (смотри спецификацию SA протокола);
- name - имя поля запроса (используется в поле VALUES). Если имя этого поля используется для задачи значения в VALUES, то будет использоваться значение из VALUES.

Внимание: Краткое описание полей протокола SA приведено в приложении 7.2.

Пример PACKET:

```
<PACKET>
  <!-- сумма в минимальных единицах валюты -->
```

```
  <FIELD code="0" name="summa">1000</FIELD>
  <!-- код валюты операции -->
  <FIELD code="4" name="currency">643</FIELD>
  <!-- код карточной операции операции "оплата"-->
  <FIELD code="25">1</FIELD>
</PACKET>
```

6.1.2 Формат поля VALUES запроса (протокол SA)

```
<VALUES>
  <[имя поля PACKET/FIELD]>[значение поля]</[имя поля]>
  ...
  <[имя поля PACKET/FIELD]>[значение поля]</[имя поля]>
</VALUES>
```

Имя поля должно соответствовать в точности имени поля FIELD (атрибут «name») в описании PAYCARD/PACKET (смотри 6.1.1).

Значения переменной может иметь любого типа (допустимого для соответствующего поля).

Например:

```
<VALUES>
  <summa>2000</summa>
</VALUES>
```

Внимание: Если в PACKET указано несколько полей FIELD с одним именем, значения будут присваиваться первому.

6.2 Формат ответа на запрос «ПРОВЕДЕНИЕ КАРТОЧНОЙ ОПЕРАЦИИ»

Внимание: Ответ с кодом ошибки отправляется только в случае если запрос не передан модулю обслуживания платежных карт (например если ошибка формата запроса или модуль обслуживания платежных карт не найден). Если операция отклонена или произошла любая другая ошибка в модуле обслуживания платежных карт, то в этом случае возвращается нормальный ответ и в этом случае надо анализировать поля VALUES ответа.

Ответ имеет следующую структуру:

`<XML ...>` - заголовок XML документа
`<RESPONSE>` - начало данных ответа

```
  <ID>[идентификатор запроса (число: unsigned int)]</ID>
  <TD>[время/дата]</TD>
  <ERROR>0</ERROR>
  <STATE>0</STATE>
  <VALUES>[значения полей ответа]</VALUES>
```

`</RESPONSE>` - конец данных ответа
`[0x00]` – стоповый символ

6.2.1 Формат поля VALUES ответа (протокол SA)

```
<VALUES>
  <FIELD code="[код поля:число]">[значение поля]</FIELD>
  ...
  <FIELD code="[код поля:число]">[значение поля]</FIELD>
```

```
</VALUES>
```

Атрибуты поля FIELD:

- code – числовой код поля запроса (смотри спецификацию SA протокола).

В ответе передаются все поля из пакета ответа модуля обслуживания платежных карт.

Пример ответа (операция одобрена):

```
<VALUES>
 <FIELD code="0">2000</FIELD>
 <FIELD code="4">643</FIELD>
 <FIELD code="6">20150402131349</FIELD>
 <FIELD code="10">xxxxxxxxxxxxxx</FIELD>
 <FIELD code="11">1612</FIELD>
 <FIELD code="13">312581</FIELD>
 <FIELD code="14">000010000010</FIELD>
 <FIELD code="15">00</FIELD>
 <FIELD code="19">ОДОБРЕНО</FIELD>
 <FIELD code="21">20150402131349</FIELD>
 <FIELD code="23">-1</FIELD>
 <FIELD code="25">1</FIELD>
 <FIELD code="26">-1</FIELD>
 <FIELD code="27">00199991</FIELD>
 <FIELD code="28">1111111111</FIELD>
 <FIELD code="39">1</FIELD>
 <FIELD code="90">0xDF^^DEMO</FIELD>
</VALUES>
```

Пример ответа (операция отклонена):

```
<VALUES>
  <FIELD code="0">2000</FIELD>
  <FIELD code="4">643</FIELD>
  <FIELD code="6">20150402133250</FIELD>
  <FIELD code="10">xxxxxxxxxxxxxx</FIELD>
  <FIELD code="11">1612</FIELD>
  <FIELD code="15">ER3</FIELD>
  <FIELD code="19">ОПЕРАЦИЯ ОТКЛОНЕНА</FIELD>
  <FIELD code="21">20150402133250</FIELD>
  <FIELD code="23">-1</FIELD>
  <FIELD code="25">1</FIELD>
  <FIELD code="26">-1</FIELD>
  <FIELD code="27">00199991</FIELD>
  <FIELD code="28">1111111111</FIELD>
  <FIELD code="39">34</FIELD>
</VALUES>
```

6.3 Формат запроса «ПЕЧАТЬ ДОКУМЕНТА»

Структура запроса:

```
<XML ...> - заголовок XML документа
<REQUEST> - начало данных запроса
  <ID>[идентификатор запроса (число: unsigned int)]</ID>
  <TD>[время/дата]</TD>
  <PAYCARD>
    <TYPE>PRINTLAST</TYPE> - тип запроса
   </PAYCARD>
  <VALUES/>
```

</REQUEST> - конец данных запроса
[0x00] – стоповый символ

С помощью этого запроса можно распечатать документ полученный от модуля обслуживания платежных карт.

Внимание: Определение наличия документа для печати, производится путем анализа ответа, полученного при проведении операции модулем обслуживания платежных карт. Наличие поля 90, говорит о том что документ для печати получен.

Внимание: При выполнении этого запроса, не происходит обращение к модулю обслуживания платежных карт.

6.4 Формат ответа на запрос «ПЕЧАТЬ ДОКУМЕНТА»

Внимание: Ответ с кодом ошибки отправляется только в случае если не активирован модуль обслуживания платежных карт. Если ошибка произошла при печати документа, то чтобы ее определить — нужно анализировать поле STATE ответа.

Ответ имеет следующую структуру:

<XML ...> - заголовок XML документа
<RESPONSE> - начало данных ответа
  <ID>[идентификатор запроса (число: unsigned int)]</ID>
  <TD>[время/дата]</TD>
  <ERROR>0</ERROR>
  <STATE>[состояние печати документа (число: int)]</STATE>
</RESPONSE> - конец данных ответа
[0x00] – стоповый символ

Поле STATE может принимать следующие значения:

Значение	Комментарий
0	Документ напечатан успешно.
-1	Нет данных для печати.
-2	Неправильный формат данных для печати.
-3	Ошибка при печати документа.
-4	Закончилась бумага в принтере.

Внимание: При успешной печати, данные документа удаляются. Также данные документа удаляются при проведении новой операции с модулем обслуживания платежных карт.

6.5 Формат запроса «ПОВТОР ОТВЕТА ПРОВЕДЕННОЙ КАРТОЧНОЙ ОПЕРАЦИИ»

Внимание: Этот запрос предназначен для повторения ответа последней проведенной операции с модулем обслуживания платежных карт. Его рекомендуется применять в случае восстановления связи, если во время операции с картой была потеряна связь с терминалом IRAS.

Структура запроса:

<XML ...> - заголовок XML документа

<REQUEST> - начало данных запроса
  <ID>[идентификатор запроса (число: unsigned int)]</ID>
  <TD>[время/дата]</TD>
  <PAYCARD>
    <TYPE>RESPLAST</TYPE> - тип запроса
  </PAYCARD>
  <VALUES/>
</REQUEST> - конец данных запроса
[0x00] – стоповый символ

6.6 Формат ответа на запрос «ПОВТОР ОТВЕТА ПРОВЕДЕННОЙ КАРТОЧНОЙ ОПЕРАЦИИ»

Внимание: Ответ с кодом ошибки отправляется только в случае если не активирован модуль обслуживания платежных карт. Если ошибка произошла при чтении файла ответа, то чтобы ее определить — нужно анализировать поле STATE ответа.

Ответ имеет следующую структуру:

<XML ...> - заголовок XML документа
<RESPONSE> - начало данных ответа
  <ID>[идентификатор запроса (число: unsigned int)]</ID>
  <TD>[время/дата]</TD>
  <ERROR>0</ERROR>
  <STATE>[состояние файла ответа (число: int)]</STATE>
  <VALUES>[данные ответа: смотри раздел 6.2]</VALUES>
</RESPONSE> - конец данных ответа
[0x00] – стоповый символ

Поле STATE может принимать следующие значения:

Значение	Комментарий
0	Файл ответа успешно прочитан.
-1	Файл ответа не найден.

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<RESPONSE>
  <ID>105</ID>
  <TD>1920 220415</TD>
  <ERROR>0</ERROR>
  <STATE>0</STATE>
  <VALUES>
    <RESPONSE>
      <ID>101</ID>
      <TD>1919 220415</TD>
      <ERROR>0</ERROR>
      <STATE>0</STATE>
      <VALUES>
        <FIELD code="0">2000</FIELD>
        <FIELD code="4">643</FIELD>
        <FIELD code="6">20150422191908</FIELD>
        <FIELD code="15">ER3</FIELD>
        <FIELD code="19">ОПЕРАЦИЯ ПРЕРВАНА</FIELD>
        <FIELD code="21">20150422191908</FIELD>
```

```
        <FIELD code="23">-1</FIELD>
        <FIELD code="25">1</FIELD>
        <FIELD code="26">-1</FIELD>
        <FIELD code="27">0</FIELD>
        <FIELD code="28">0</FIELD>
        <FIELD code="39">53</FIELD>
      </VALUES>
    </RESPONSE>
  </VALUES>
</RESPONSE>
```

6.7 Особенности работы с модулем обслуживания платежных карт «Сбербанк UPOS»

Так как модуль обслуживания платежных карт «Сбербанк UPOS» не является модулем, работающим по SA протоколу, то работа с этим модулем идет в режиме эмуляции. То есть программа сама анализирует SA поля и делает запрос к модулю обслуживания карт согласно его требований. В случае неправильного значения полей (например неправильный код валюты или не поддерживаемый код операции) из за которых не происходит передача запроса модулю обслуживания платежных карт, будет возвращена ошибка работы с модулем. Если запрос передан модулю обслуживания платежных карт, то состояние обработки запроса нужно анализировать по соответствующим полям ответа.

Поддерживаемые операции и возможные поля запросов и ответов (неуказанные поля игнорируются):

Операция	**Поля запроса (синим отмечены необязательные поля)**	**Поля ответа (синим отмечены поля, появляющиеся в случае успешной операции)**
Оплата	0: сумма операции 4: «643» (код валюты) 25: «1» (код операции) 26: внутренний номер операции	0: сумма операции 6: дата и время операции 10: PAN 11: срок действия карты 12: хэш от номера карты 13: код авторизации 14: ссылочный номер операции 15: код ответа 19: дополнительные данные ответа 25: «1» (код операции) 26: внутренний номер операции 27: номер терминала 39: статус («1»: успех) 62: признак карты сбербанка («1») 63: название типа карты
Отмена	0: сумма операции 4: «643» (код валюты) 12: хэш от номера карты 25: «4» (код операции) 26: внутренний номер операции	0: сумма операции 6: дата и время операции 10: PAN 11: срок действия карты 13: код авторизации 14: ссылочный номер операции 15: код ответа 19: дополнительные данные ответа 25: «4» (код операции) 26: внутренний номер операции 27: номер терминала 39: статус («1»: успех) 62: признак карты сбербанка («1») 63: название типа карты

Возврат	0: сумма операции 4: «643» (код валюты) 12: хэш от номера карты 25: «29» (код операции) 26: внутренний номер операции	0: сумма операции 6: дата и время операции 10: PAN 11: срок действия карты 13: код авторизации 14: ссылочный номер операции 15: код ответа 19: дополнительные данные ответа 25: «29» (код операции) 26: внутренний номер операции 27: номер терминала 39: статус («1»: успех) 62: признак карты сбербанка («1») 63: название типа карты
Преавторизация	0: сумма операции 4: «643» (код валюты) 25: «15» (код операции) 26: внутренний номер операции	0: сумма операции 6: дата и время операции 10: PAN 11: срок действия карты 13: код авторизации 14: ссылочный номер операции 15: код ответа 19: дополнительные данные ответа 25: «15» (код операции) 26: внутренний номер операции 27: номер терминала 39: статус («1»: успех) 62: признак карты сбербанка («1») 63: название типа карты
Завершение расчета	0: сумма операции 4: «643» (код валюты) 12: хэш от номера карты 14: ссылочный номер операции 25: «16» (код операции) 26: внутренний номер операции	0: сумма операции 6: дата и время операции 10: PAN 11: срок действия карты 13: код авторизации 14: ссылочный номер операции 15: код ответа 19: дополнительные данные ответа 25: «16» (код операции) 26: внутренний номер операции 27: номер терминала 39: статус («1»: успех) 62: признак карты сбербанка («1») 63: название типа карты

Откат последней транзакции	0: сумма операции 4: «643» (код валюты) 13: код авторизации 25: «53» (код операции) 26: внутренний номер операции	0: сумма операции 6: дата и время операции 10: PAN 11: срок действия карты 13: код авторизации 14: ссылочный номер операции 15: код ответа 19: дополнительные данные ответа 25: «53» (код операции) 26: внутренний номер операции 27: номер терминала 39: статус («1»: успех) 62: признак карты сбербанка («1») 63: название типа карты
Сверка итогов	25: «59» (код операции)	15: код ответа 19: дополнительные данные ответа 25: «59» (код операции) 39: статус («1»: успех)
Краткий отчет	25: «63» (код операции) 65: «20» (дополнительный код)	15: код ответа 19: дополнительные данные ответа 25: «63» (код операции) 65: «20» (дополнительный код) 67: статус («0»: успех)
Полный отчет	25: «63» (код операции) 65: «21» (дополнительный код)	15: код ответа 19: дополнительные данные ответа 25: «63» (код операции) 65: «21» (дополнительный код) 67: статус («0»: успех)
Сервисное обслужива-ние	25: «63» (код операции) 65: «40» (дополнительный код)	15: код ответа 19: дополнительные данные ответа 25: «63» (код операции) 65: «40» (дополнительный код) 67: статус («0»: успех)

Замечание: Работа с полем 26 имеет следующие особенности. В случае если это поле задано в запросе, то оно возвращается в ответе (копируется из запроса), при этом поле не передается в модуль обслуживания платежных карт. В случае если это поле не задано в запросе, то оно копируется из ответа модуля обслуживания платежных карт «Сбербанк» (т. е. возвращается внутренний номер, присвоенный модулем обслуживания платежных карт).

Замечание: В случае использования модуля обслуживания платежный карт «Сбербанк UPOS» данные для печати не возвращаются в поле 90. Вызывать нужно функцию печати в случае получения успешного кода ответа.

6.7.1 Пример работы с модулем «Сбербанк UPOS» (оплата/частичная отмена)

Ниже приведен пример запросов/ответов на проведение операций:

- Оплата платежной картой (с предъявлением платежной карты);

- Частичная отмена оплаты платежной картой (без предъявления платежной карты).

Запрос «Оплата платежной картой» на сумму 20руб:

```
<?xml version="1.0" encoding="utf-8"?>
<REQUEST>
  <ID>101</ID>
  <TD>1157 310515</TD>
  <PAYCARD>
    <TYPE>OPERATION</TYPE>
    <!-- используется протокол SA -->
    <PROTO>SA</PROTO>
    <PACKET>
      <!-- сумма в минимальных единицах валюты -->
      <FIELD code="0" name="summa">1000</FIELD>
      <!-- код валюты операции -->
      <FIELD code="4" name="currency">643</FIELD>
      <!-- код карточной операции операции "оплата"-->
      <FIELD code="25">1</FIELD>
    </PACKET>
  </PAYCARD>
  <VALUES>
    <!-- если для операции используется другое значение какого либо поля:
указываем его здесь по имени поля (name) -->
    <summa>2000</summa>
  </VALUES>
</REQUEST>
```

Ответ на запрос «Оплата платежной картой»:

```
<?xml version="1.0" encoding="utf-8"?>
<RESPONSE>
  <ID>101</ID>
  <TD>1159 310515</TD>
  <ERROR>0</ERROR>
  <STATE>0</STATE>
  <VALUES>
    <FIELD code="25">1</FIELD>
    <FIELD code="0">2000</FIELD>
    <FIELD code="15">0</FIELD>
    <FIELD code="19">ОДОБРЕНО </FIELD>
    <FIELD code="39">1</FIELD>
    <FIELD code="10">462776******3536</FIELD>
    <FIELD code="11">1003</FIELD>
    <FIELD code="13">06E080</FIELD>
    <FIELD code="26">0001</FIELD>
    <FIELD code="63">Visa</FIELD>
    <FIELD code="62">0</FIELD>
    <FIELD code="27">00409999</FIELD>
    <FIELD code="6">20150531115929</FIELD>
    <FIELD code="14">143306277446</FIELD>
    <FIELD code="12">C560EB7664D7F8D36EF9A7032959215176CC13D7</FIELD>
  </VALUES>
</RESPONSE>
```

Из ответа берем поле «12» с хэшом трека 2 платежной карты и передаем его в запрос частичной отмены. Передаем запрос на частичную отмену 10руб:

```
<?xml version="1.0" encoding="utf-8"?>
<REQUEST>
  <ID>106</ID>
  <TD>1202 310515</TD>
  <PAYCARD>
    <TYPE>OPERATION</TYPE>
```

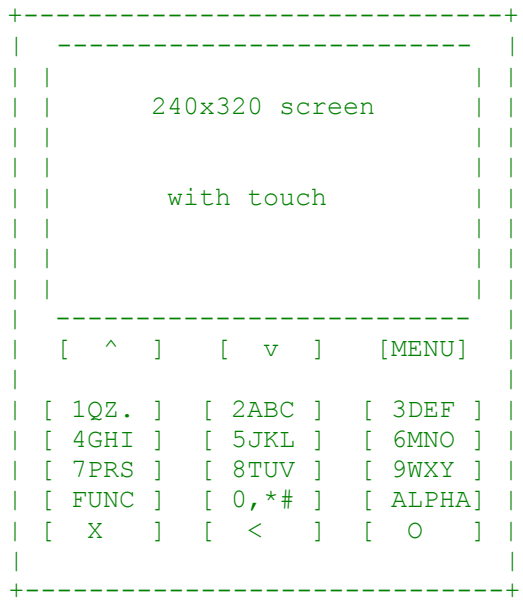
```
    <!-- используется протокол SA>
    <PROTO>SA</PROTO>
    <PACKET>
      <!-- сумма в минимальных единицах валюты -->
      <FIELD code="0" name="summa">1000</FIELD>
      <!-- код валюты операции -->
      <FIELD code="4" name="currency">643</FIELD>
      <!-- код карточной операции операции "отмена"-->
      <FIELD code="25">4</FIELD>
      <!-- хэш трека 2 -->
      <FIELD code="12">C560EB7664D7F8D36EF9A7032959215176CC13D7</FIELD>
    </PACKET>
  </PAYCARD>
  <VALUES>
    <!-- если для операции используется другое значение какого либо поля:
указываем его здесь по имени поля (name) -->
    <summa>1000</summa>
  </VALUES>
</REQUEST>
```

Обработка запроса «Частичная отмена» происходит без предъявления платежной карты.
Ответ на запрос «Частичная отмена»:

```
<?xml version="1.0" encoding="utf-8"?>
<RESPONSE>
  <ID>106</ID>
  <TD>1204 310515</TD>
  <ERROR>0</ERROR>
  <STATE>0</STATE>
  <VALUES>
    <FIELD code="25">4</FIELD>
    <FIELD code="0">1000</FIELD>
    <FIELD code="15">0</FIELD>
    <FIELD code="19">ОДОБРЕНО </FIELD>
    <FIELD code="39">1</FIELD>
    <FIELD code="10">462776******3536</FIELD>
    <FIELD code="11">1003</FIELD>
    <FIELD code="13">06E080</FIELD>
    <FIELD code="26">0001</FIELD>
    <FIELD code="63">Visa</FIELD>
    <FIELD code="62">0</FIELD>
    <FIELD code="27">00409999</FIELD>
    <FIELD code="6">20150531115929</FIELD>
    <FIELD code="14">143306277446</FIELD>
    <FIELD code="12">C560EB7664D7F8D36EF9A7032959215176CC13D7</FIELD>
  </VALUES>
</RESPONSE>
```

7 Приложение

7.1 Коды и значения клавиш терминала IRAS



Клавиша	Код	Значение
[^]	x67	Перемещение на предыдущий пункт меню или перемещение к предыдущему активному объекту.
[v]	x6C	Перемещение на следующий пункт меню или перемещение к следующему активному объекту.
[MENU]	x8B	
[1QZ.]	x31	Ввод 1. В режиме ввода нецифровых значений — ввод дополнительных символов при повторном нажатии.
[2ABC]	x32	Ввод 2. В режиме ввода нецифровых значений — ввод дополнительных символов при повторном нажатии.
[3DEF]	x33	Ввод 3. В режиме ввода нецифровых значений — ввод дополнительных символов при повторном нажатии.
[4GHI]	x34	Ввод 4. В режиме ввода нецифровых значений — ввод дополнительных символов при повторном нажатии.
[5JKL]	x35	Ввод 5. В режиме ввода нецифровых значений — ввод дополнительных символов при повторном нажатии.
[6MNO]	x36	Ввод 6. В режиме ввода нецифровых значений — ввод дополнительных символов при повторном нажатии.
[7PRS]	x37	Ввод 7. В режиме ввода нецифровых значений — ввод дополнительных символов при повторном нажатии.
[8TUV]	x38	Ввод 8. В режиме ввода нецифровых значений — ввод дополнительных символов при повторном нажатии.
[9WXY]	x39	Ввод 9. В режиме ввода нецифровых значений — ввод

		дополнительных символов при повторном нажатии.
[0,*#]	x30	Ввод 0. В режиме ввода нецифровых значений — ввод дополнительных символов при повторном нажатии.
[FUNC]	x66	Переключение между дробной и целой частью вводимых чисел. **Внимание:** В случае режима ввода шестнадцатеричных чисел, включает режим ввода цифры A-F (при условии нажатия затем соответственно клавиши 1-6).
[ALPHA]	x45	
[X]	x1B	Отмена (Cancel).
[<]	x08	Удаление последнего введенного символа.
[O]	x0D	Подтверждение (OK).

7.2 Значения и описание полей FIELD (протокол SA)

Внимание: Здесь приводится описание полей исключительно для справочной информации, более подробную информацию необходимо смотреть в описании протокола SmartAccess.

Тип поля указывается в виде «**T[[M]..]N**», где:

T – Один из следующих префиксов, определяющих тип используемых символов в значении данного поля:

- a – Только алфавитные символы, и без привязки к какому-либо языку;
- n – Только цифровые символы ‘0’, ‘1’, …, ‘9’;
- an – Алфавитные или цифровые символы;
- z – Любые символы, в том числе и бинарные (‘\x00’ – ‘\x31’), т.е. со значением байта 0 – 255.

[..] – Многоточие, указывается опционально и является признаком переменного размера поля;

M – Минимальное количество символов, указывается опционально и при наличии многоточия [..];

N – Если многоточие [..] отсутствует, то N - обязательный размер в байтах данного поля, иначе максимальный размер.

Код	**Описание**	**Тип**
0	Сумма операции, выраженная в минимальных единицах валюты	n12
4	Код валюты операции	n3
6	Оригинальная дата и время совершения операции YYYYMMDDHHMMSS на хосте	n14
10	Номер карты	an13..19
11	Срок действия карты YYMM	n4
12	Данные Track2	z21..37
13	Код авторизации	an..8
14	Номер ссылки	n..12
15	Код ответа	an..3
19	Дополнительные данные ответа	an..99
21	Оригинальная дата и время совершения операции YYYYMMDDHHMMSS на внешнем устройстве	n14
23	Идентификатор транзакции в коммуникационном сервере	n..6
25	Код операции	n..2
26	Уникальный номер транзакции на стороне внешнего устройства	n..6
27	Идентификатор внешнего устройства	an..15
28	Идентификатор продавца	an..15

39	Статус проведения транзакции	n..3
47	CVV2	n4
53	Флаг обработки операции	n3
54	Уникальный номер транзакции на хосте авторизации	n..10
90	Данные для печати на чеке	z..9999

Значение поля 25 для различных операций:

Значение	**Описание операции**
1	Оплата товаров и услуг
2	Выдача наличных
3	Пополнение счета клиента
13	Состояние счета клиента
14	Авторизация
15	Преавторизация
16	Завершение преавторизации
22	Заказ по почте
23	Отмена заказа по почте
24	Открытие смены
25	Закрытие смены
29	Возврат
47	Оплата товаров с выдачей наличных
49	Оплата услуг
52	Запрос данных у кассира
53	Аварийная отмена
57	Получить данные TRACK1 и TRACK2
58	Преавторизация платежа в сторону третьих лиц
59	Сверка итогов
60	Мини-выписка
61	Платеж
62	Платеж в сторону третьих лиц
71	Кредит ваучер

8 История изменений

01.10.2017:

- Добавлено описание атрибута barscan для формы (смотри раздел 4.2);
- Добавлено описание поточного режима работы со сканером штрихкодов (смотри раздел 5.1.1).

05.09.2017:

- Добавлен раздел с описанием специльной формы **FORM_MSR** (смотри раздел 5.3).

09.09.2016:

- Расширено описание значений возврата **STATE** (смотри раздел 3.4.1);
- Добавлено описание атрибутов **ccodes**, **scodes** для **FORM** (смотри раздел 4.2).

07.09.2016:

- Добавлено описание атрибутов **passahex**, **passbhex** для специальной формы **FORM_MIFARE** (смотри раздел 5.2);
- Добавлен раздел « Значение поля **BT** ответа (текущий заряд батареи)» (смотри раздел 3.3.1).

05.09.2016:

- Добавлен раздел «Специальные типы форм»: **FORM_BARCODE**, **FORM_MIFARE** (смотри раздел 5);
- Добавлено описание новых полей приветствия: **SN**, **PN**, **SCR** (смотри раздел 2.4);
- Уточнены флаги атрибута **options** для **MENU** (смотри раздел 4.3).

11.01.2016:

- Добавлен пример сообщения «Приветствие» (смотри раздел 3.1);
- Уточнен пример широковещательного пакета (смотри раздел 2.5).

23.12.2015:

- Добавлена операция «Сервисное обслуживание» в описание операций модуля обслуживания карт «Сбербанк» (смотри раздел 6.7);
- Уточнено описание примитива контейнер графики **IMAGE** (смотри раздел 4.1.1);
- Добавлено описание атрибута **accelfont** объекта отображения **BUTTON** (смотри раздел 4.2.3).