

Протокол обмена Титан-Плюс.

Общие положения.

Регистраторы “Титан-А” обладают возможностью работы в TCP/IP сетях и имеют встроенный веб сервер (порт 80 регистратора). Протокол обмена построен на [протоколе HTTP](#) и представляет собой набор запросов и ответов в [формате JSON](#). Например, послав в TCP соединение с аппаратом сообщение:

```
GET /cgi/state HTTP/1.1
Host: 192.168.8.2
Authorization: Digest username="service", realm="HTROM",
nonce="527b004c29df30afd42c9dbf43dcb6d9", uri="/cgi/state",
algorithm=MD5, response="3f4f49f5adefafdf19ce9148103486af",
opaque="60DB81DD", qop=auth, nc=00000001, cnonce="669bcf2a9b1c9deb"
User-Agent: test
```

получим ответ:

```
HTTP/1.1 200 OK
Server: host name
Content-Type: application/json
Cache-Control: no-cache

{"model": "ТИТАН-
A", "name": "TA0000001", "serial": "TA0000001", "time": 1481198752, "chkId":
:0, "JrnTime": 0, "currZ": 0, "IsWrk": 0, "Fiscalization": 1, "FskMode": 1, "SK
NOState": 2626, "err": []}
```

Протокол позволяет:

- Создавать и получать и синхронизировать информацию по чекам,
- Регистрировать оператора и вести список операторов аппарата,
- Настраивать параметры и проверять состояние фискальной памяти,
- Настраивать параметры работы с сетью TCP/IP,
- Настраивать параметры видов оплат,
- Настраивать внешний вид чека,
- Печатать отчеты

Протокол представляет собой набор процедур, которые вызываются для выполнения тех или иных действий или запроса информации от аппарата на основании входных параметров. Информация, которую аппарат предоставляет по протоколу организована в виде таблиц, которые представлены как объекты JSON (однострочные таблицы) или массивы JSON (многострочные таблицы). Таблицы и процедуры являются элементами протокола, операции над которыми определены ниже.

Операции над элементами протокола.

Вызов процедуры.

Процедура вызывается **HTTP** методом **GET**. Имя ресурса для вызова процедуры имеет вид:

/cgi/proc/**name?param1¶m2&...¶mN**

где:

proc - имя процедуры

paramX - значение параметра процедуры.

Ниже приведены форматы параметров различных типов:

Тип	Название	Формат
Двоичная информация	Binary	х с набором шестнадцатеричных цифр Пример: x010203aaBC
Число	Number	Набор цифр с необязательным знаком минус. Дробная часть, если она нужна, пишется через точку. Примеры: 12345, -2345, 12.5
Дата	Date	YYYY-MM-DD
Время	Time	[T]HH:MM[:SS]
Дата и время	ISODate	YYYY[-MM[-DD]][T]HH[:MM[:SS]]]
Текст	Text	В формате кодирования URL http://en.wikipedia.org/wiki/Percent-encoding Пример: ABC%20%E0%B1

Перечень процедур

fiscalization – регистрация

Входящие параметры: Нет.

Выходящие параметры: Нет.

getfmroom – состояние фискальной памяти

Входящие параметры: Нет

Выходящие параметры:

Описание	Тип	Значения
Общее количество записей	Number	
Свободно записей	Number	

getjrnroom– состояние электронного журнала

Входящие параметры: Нет

Выходящие параметры:

Описание	Тип	Значения
Общее количество записей	Number	
Свободно записей	Number	

lastreceipt– информация по последнем зарегистрированном документе

Входящие параметры: Нет

Выходящие параметры:

Описание	Тип	Значения
Общая сумма документа	Number	
Номер документа	Number	
Дата и время документа	Number	
Номер оператора	Number	

printfmreport – печать отчета с БЭП

Входящие параметры:

Описание	Тип	Значения
Тип отчета	Number	1 – полный по датам 2 – полный по номерам 3 – сокращенный по датам 4 – сокращенный по номерам
Начальная дата диапазона	Date	
Конечная дата диапазона	Date	
Начальный номер отчета	Number	
Конечный номер отчета	Number	

Выходящие параметры: Нет

printreport – печать отчета

Входящие параметры:

Описание	Тип	Значения
Тип отчета	Number	0 – дневной обнуляющий отчет (Z1) 1 – обнуление электронного журнала 2 – печать контрольной ленты с обнулением 3 – печать отчета состояния СКНО 4 – печать отчета по открытым ресторанным заказам с автоматическим закрытием наличными 5 – печать отчета по открытым ресторанным заказам 10, 100 – дневной отчет без обнуления (X1) 20 – Отчет по проданным товарам 21 – Отчет по проданным товарам с обнулением этого отчета 102 – отчет по денежному ящику 107 – статистика продаж по времени 201 – 216 – статистика продаж по кассирам

Выходящие параметры: Нет

puthdrfm – перерегистрация

Входящие параметры: Нет.

Выходящие параметры: Нет.

puttaxfm – запись налоговых ставок в БЭП

Входящие параметры: Нет.

Выходящие параметры: Нет.

setclock – установка времени

Входящие параметры:

Описание	Тип	Значения
----------	-----	----------

Дата и время	ISODate	
--------------	---------	--

Процедура может вызываться только в случае замкнутой перемычки SA2 на системной плате или когда аппарат находится в состоянии ошибки часов.

Выходящие параметры: Нет.

sknstate—состояние модуля СКНО

Входящие параметры: Нет

Выходящие параметры:

Описание	Тип	Значения
Битовая маска состояния	Number	

sound – звуковой сигнал

Входящие параметры:

Описание	Тип	Значения
Длительность (мс)	Number	
Частота (Гц)	Number	60-4096

Выходящие параметры: Нет.

state—состояние текущего документа

Входящие параметры: Нет

Выходящие параметры:

Описание	Тип	Значения
Битовая маска состояния документа	Number	0x0001 – открыт фискальный чек 0x0002 – открыт не фискальный чек 0x0004 – для закрытия чека требуется оплата 0x0008 – напечатаны итоги по чеку 0x0010 – напечатан заголовок чека 0x0020 – в чеке были продажи товара 0x0040 – в чеке были вносы/выносы 0x0080 – начата расплата по чеку 0x0100 – печать контрольной ленты 0x0200 – печать служебного отчета 0x0400 – отмена документа 0x0800 – в чеке есть операции 0x1000 – начат чек возврата 0x2000 – запись занесена в ленту
Номер оператора, начавшего документ	Number	
Начальная запись документа в журнале	Number	
Начальная запись отчета в журнале	Number	

Пример вызова процедуры

```
GET /cgi/proc/sound?1000&1000 HTTP/1.1
Host: 192.168.8.2
User-Agent: test
Authorization: Digest username="admin", realm="HTROM",
nonce="5d6553aca400536fcd1dca1d467bc428",
uri="/cgi/proc/sound?1000&1000", algorithm=MD5,
```

```
response="7053664025903c9c1485a188023909bd", opaque="3115FB22",
qop=auth, nc=00000001, cnonce="63d86b75894ca977"
```

HTTP/1.1 200 OK

Server: host name

Content-Type: application/json

Cache-Control: no-cache

Content-Length: 2

```
{}
```

Таблицы. Чтение таблицы.

Таблица представляет собой одну или несколько строк с несколькими полями. Как правило все таблицы аппарата имеют фиксированное количество строк. Вставка и удаление строк в таких таблицах невозможна. Однострочные таблицы представляются как объекты JSON.

Многострочные - как массивы объектов.

В многострочных таблицах одно из полей является ключевым, По этому полю происходит изменение строки. В большинстве случаев это поле имеет имя **id**. В случаях, когда поле имеет другое имя, это оговаривается дополнительно в описании конкретной таблицы.

Чтение таблицы производится HTTP методом GET ресурса `/cgi/tbl/name`, где `name` - имя таблицы.

Пример чтения однострочной таблицы:

```
GET /cgi/tbl/Flg HTTP/1.1
Host: 192.168.8.2
User-Agent: test
Authorization: Digest username="admin", realm="HTROM",
nonce="f7c7d74960ba779d53fcbb891d571ba1", uri="/cgi/tbl/Flg",
algorithm=MD5, response="4c8504fb97b6ef3bac0532959d3b6277",
opaque="33539145", qop=auth, nc=00000001, cnonce="58eb5e9073bd08bc"
```

HTTP/1.1 200 OK

Server: host name

Content-Type: application/json

Cache-Control: no-cache

```
{"Lang":0,"AutoOff":0,"PrintOff":0,"Flg1":8448,"Flg2":0,"Feed":7,"Flg3":4}
```

Пример чтения многострочной таблицы:

```
GET /cgi/tbl/Pay HTTP/1.1
Host: 192.168.8.2
```

```
User-Agent: test
Authorization: Digest username="admin", realm="HTROM",
nonce="f7c7d74960ba779d53fcbb891d571ba1", uri="/cgi/tbl/Pay",
algorithm=MD5, response="c89610a4a3f08f91ad45fe8fb3ac747b",
opaque="33539145", qop=auth, nc=00000004, cnonce="32292630ba10c70b"
```

```
HTTP/1.1 200 OK
Server: host name
Content-Type: application/json
Cache-Control: no-cache
```

```
[
{"id":1,"Param":35,"Name":"I"},
{"id":2,"Param":1,"Name":""},
{"id":3,"Param":17,"Name":""},
{"id":4,"Param":17,"Name":""}]
```

Изменение таблицы.

Изменение содержимого таблицы **/cgi/tbl/name** производится HTTP методом **PATCH** (или методом **POST** с HTTP заголовком **X-HTTP-Method-Override=PATCH**).

Для изменения многострочной таблицы в изменяемых строках обязательно нужно указывать ключевое поле. Если изменение затрагивает одну строку таблицы, измененная строка возвращается в ответе при успешной модификации. При успешной модификации нескольких строк таблицы возвращается пустой объект. При ошибке модификации строки, возвращается ошибка с идентификатором этой строки, модификация дальнейших строк не производится. Формат возврата ошибок приводится в соответствующем разделе ниже.

Пример модификации однострочной таблицы.

```
POST /cgi/tbl/Flg HTTP/1.1
Host: 192.168.8.2
Content-Length: 10
Authorization: Digest username="admin", realm="HTROM",
nonce="8697c1e6d12141d6b4d8f74a9e53dc57", uri="/cgi/tbl/Flg",
algorithm=MD5, response="66af9754042863ae698045430bd1f934",
opaque="3F839E97", qop=auth, nc=00000001, cnonce="c1e81fbae20ea51c"
User-Agent: test
X-HTTP-Method-Override: PATCH
Content-Type: text/plain; charset=UTF-8
```

```
{"Feed":1}
```

```
HTTP/1.1 200 OK
Server: host name
Content-Type: application/json
Cache-Control: no-cache
```

```
{"Lang":0,"AutoOff":0,"PrintOff":0,"Flg1":8448,"Flg2":0,"Feed":7,"Flg3":4}
```

Пример модификации нескольких строк многострочной таблицы

```
POST /cgi/tbl/Pay HTTP/1.1
Host: 192.168.8.2
Content-Length: 41
Authorization: Digest username="admin", realm="HTROM",
nonce="a5cba8b926bdcabc203f1b58df8a9bbdc", uri="/cgi/tbl/Pay",
algorithm=MD5, response="44959fdb6d2cb645631bc9f282dc43ca",
opaque="6ABBEF89", qop=auth, nc=00000002, cnonce="a4c55c6905bd1ea7"
User-Agent: test
X-HTTP-Method-Override: PATCH
Content-Type: text/plain;charset=UTF-8
```

```
[{"id":3,"Param":16},{"id":4,"Param":16}]
```

```
HTTP/1.1 200 OK
Server: host name
Content-Type: application/json
Cache-Control: no-cache
```

```
{}
```

Пример модификации одной строки многострочной таблицы

```
POST /cgi/tbl/Pay HTTP/1.1
Host: 192.168.8.2
Content-Length: 21
Authorization: Digest username="admin", realm="HTROM",
nonce="948dd8135bdfdfdc86f72e639daf2b4", uri="/cgi/tbl/Pay",
algorithm=MD5, response="3b593598d154d0d778849ed63827ffff",
opaque="6CE1D139", qop=auth, nc=00000002, cnonce="a58471689c9a5979"
User-Agent: test
X-HTTP-Method-Override: PATCH
Content-Type: text/plain;charset=UTF-8
```

```
[{"id":3,"Param":17}]
```

```
HTTP/1.1 200 OK
Server: host name
Content-Type: application/json
Cache-Control: no-cache
```

```
{"id":3,"Param":17,"Name":""}
```

Пример модификации таблицы с ошибкой

```
POST /cgi/tbl/Pay HTTP/1.1
Host: 192.168.8.2
Content-Length: 20
Authorization: Digest username="admin", realm="HTROM",
nonce="26ef38e8241ae74b8bb1f88692d5aa72", uri="/cgi/tbl/Pay",
algorithm=MD5, response="c9c42ee5e587b8a922d7520559cda5bf",
opaque="6E81E29C", qop=auth, nc=00000001, cnonce="2c118b13a5a80880"
User-Agent: test
X-HTTP-Method-Override: PATCH
Content-Type: text/plain; charset=UTF-8
```

```
[{"id":4,"Parm":17}]
```

```
HTTP/1.1 200 OK
Server: host name
Content-Type: application/json
Cache-Control: no-cache
```

```
{"id":4,"err":{"e":"xC2"}}
```

Изменение строки многострочной таблицы.

Изменение строки таблицы по адресу **/cgi/tbl/name/id** производится HTTP методом **PATCH** (или методом **POST** с HTTP заголовком **X-HTTP-Method-Override=PATCH**).

Пример изменения строки

```
POST /cgi/tbl/Pay/4 HTTP/1.1
Host: 192.168.8.2
Content-Length: 12
Authorization: Digest username="admin", realm="HTROM",
nonce="baa2d2b3937b0ec7564f1432b3a52703", uri="/cgi/tbl/Pay/4",
algorithm=MD5, response="c49320d67e2083f1a11070165e2f90ab",
opaque="3A9493BB", qop=auth, nc=00000001, cnonce="155183b79fb9ffe0"
User-Agent: test
X-HTTP-Method-Override: PATCH
Content-Type: text/plain; charset=UTF-8
```

```
{"Param":17}
```

```
HTTP/1.1 200 OK
Server: host name
Content-Type: application/json
Cache-Control: no-cache
```

```
{"id":4,"Param":17,"Name":""}
```

Перезапись содержимого таблицы. (только для таблицы PLU)

Полное стирание таблицы и вставка в неё переданной информации. Производится HTTP методом PUT (или методом POST с HTTP заголовком X-HTTP-Method-Override=PUT) по адресу /cgi/tbl/PLU.

Пример перезаписи таблицы

```
POST /cgi/tbl/PLU HTTP/1.1 Host: 192.168.0.104 Content-Length: 151
Authorization: Digest username="1", realm="HTROM",
nonce="3a3de38611220f47b71069c54cf7809c", uri="/cgi/tbl/PLU",
algorithm=MD5, response="95f0adeec51c3579f9a918aa93d84023",
opaque="64114B62", qop=auth, nc=00000001, cnonce="0099af6f5cda7aec"
User-Agent: test X-HTTP-Method-Override: PUT Content-Type:
text/plain;charset=UTF-8
```

```
[{"Code":1,"Name":"tov1","Price":1,"Dep":1,"Grp":1,"Tax":1,"Qty":0,"
Flg":0},{ "Code":2,"Name":"tov2","Price":2,"Dep":1,"Grp":1,"Tax":1,"Q
ty":0,"Flg":0}]
```

```
HTTP/1.1 200 OK Server: lw-http 1.03 (32-bit) Content-Type:
application/json Cache-Control: no-cache
{}
```

Вставка строки в таблицу. (только для таблицы PLU)

Вставка в таблицу переданной информации производится HTTP методом POST по адресу /cgi/tbl/PLU.

Пример вставки строк в таблицу

```
POST /cgi/tbl/PLU HTTP/1.1 Host: 192.168.0.104 Content-Length: 151
Authorization: Digest username="1", realm="HTROM",
nonce="3a3de38611220f47b71069c54cf7809c", uri="/cgi/tbl/PLU",
algorithm=MD5, response="95f0adeec51c3579f9a918aa93d84023",
opaque="64114B62", qop=auth, nc=00000001, cnonce="0099af6f5cda7aec"
User-Agent: test Content-Type: text/plain;charset=UTF-8
```

```
[{"Code":3,"Name":"tov3","Price":1,"Dep":1,"Grp":1,"Tax":1,"Qty":0,"
Flg":0},{ "Code":4,"Name":"tov4","Price":2,"Dep":1,"Grp":1,"Tax":1,"Q
ty":0,"Flg":0}]
```

```
HTTP/1.1 200 OK Server: lw-http 1.03 (32-bit) Content-Type:
application/json Cache-Control: no-cache
{}
```

Удаление строки из таблицы. (только для таблицы PLU)

Производится HTTP методом DELETE (или методом POST с HTTP заголовком X-HTTP-Method-Override=DELETE) по адресу /cgi/tbl/PLU/code.

Пример удаления строк таблицы

```
POST /cgi/tbl/PLU/2 HTTP/1.1 Host: 192.168.0.104 Connection: keep-
alive Content-Length: 0 Authorization: Digest username="1",
realm="HTROM", nonce="50965acc9a82b832266d803132c29c02",
uri="/cgi/tbl/PLU/2", algorithm=MD5,
response="99098aefb4a811fe54030522affc00e3", opaque="360ADFE3",
qop=auth, nc=00000001, cnonce="c73c05fa688b8fb9" User-Agent: test X-
HTTP-Method-Override: DELETE Content-Type: text/plain;charset=UTF-8

HTTP/1.1 200 OK Server: lw-http 1.03 (32-bit) Content-Type:
application/json Cache-Control: no-cache
{}
```

Возврат ошибок.

При возникновении ошибки операция протокола возвращает объект, в котором присутствует поле **err**.

```
{"err":...}
```

В простейшем случае это поле содержит текстовую строку с сообщением об ошибке или с кодом сообщения об ошибке. Код сообщения об ошибке имеет вид "xNN", где N - hex цифра. Список ошибок кассы вынесен в приложение 1.

Ресурс **/desc** содержит информацию по локализации, в том числе и локализованную версию сообщений об ошибках на разных языках. В настоящий момент поддерживаются украинская (поле **uk**), английская (поле **en**) и русская (поле **ru**) локализации. В поле **err**, соответствующего объекта содержатся текстовые сообщения, передаваемые кассой и их перевод на выбранный язык. Место сообщений об ошибках в ресурсе **/desc** показано ниже:

```
{
  ...
  "en":{
    ...
  },
  "uk":{
    ...
  },
  "ru":{
    ...
    "err": {
      ...
      "x3A": "Попытка модификации поля только для чтения",
      "x3B": "Неверное значение поля",
      "x3C": "Товар уже существует",
      "x3D": "По товару были продажи",
      ...
    },
    ...
  }
}
```

```

    },
    ...
}

```

В ошибке кроме кода или текста могут передаваться параметры. В этом случае поле **err** представляет собой объект с полями **e** (текст ошибки) и **p** (параметры ошибки). Например, `{"err":{"e":"Error %i", "p":[100]}}` представляет собой текст ошибки "Error 100".

Если ошибка связана с параметром процедуры, номер параметра может передаваться в поле **f** объекта ошибки. Например: `{"err":{"e":"x25", "f":2}}`.

Если ошибка связана с определенным полем строки таблицы, имя этого поля может передаваться в поле **f** объекта ошибки. Например: `{"err":{"e" : "x25", "f":"tax"}}`.

Если ошибка связана с определенной строкой чека, номер этой строки может передаваться в поле **line** объекта ошибки. Например: `{"err":{"e" : "x25", "line":5}}`.

Если вызов процедуры или другая операция генерирует несколько ошибок, полю **err** может быть присвоен массив ошибок. Например:

```

{ "err": [ { "e": "x25"}, { "e" : "x26" } ] }

```

Список таблиц кассы.

Список таблиц кассы можно прочитать используя HTTP метод GET из ресурса **/cgi/tbl**. Список таблиц является JSON массивом объектов, каждый из которых описывает одну таблицу и содержит поля:

Поле	Описание
id	имя ресурса. Таблица доступна по адресу /cgi/tbl/имя ресурса
tbl	Необязательное поле. Если присутствует и не нулевое, таблица представляется JSON массивом. Иначе - JSON объектом.
prefix	Необязательное поле. Имя группы, для логически связанных таблиц.
insertable	Необязательное поле. Если присутствует и ненулевое, в таблицу можно вставлять и удалять строки.
key	Необязательное поле. Если присутствует, то его значение является именем ключевого поля для строк таблицы. Если отсутствует, имя ключевого поля "id"
elems	Массив с описанием колонок таблицы.

Описание колонки является JSON объектом с полями:

Поле	Описание
name	Имя поля
type	Тип поля. Возможные типы: "text", "number", "date", "datetime", "datetime-local", "email", "password", "time", "url", "percent", "select-one", "select-multiple", "checkbox", "radio", "range"
pattern	regex шаблон или имя шаблона в поле regex ресурса /desc
editable	Если присутствует и равно 0 - поле не редактируется
renderable	Если присутствует и равно 0 - поле скрытое

labels	метки для типов "select-one", "select-multiple", "checkbox", "radio". Могут быть в формате: <ul style="list-style-type: none"> • строки "label1 label2 "complex label"" • массива строк ["label1", "label2", "complex label"] • массива объектов [{"value":1,"label":"label1"}, {"value":2,"label":"label2"}, {"value":3,"label":"complex label"}]
min	минимальное значение числа или даты
max	максимальное значение числа или даты
step	шаг изменения числа или даты

Печать чеков.

Чек, который должен распечатать регистратор, формируется в виде текста или в формате JSON. Этот текст методом **POST** отправляется на ресурс **/cgi/chk**. Документ должен иметь кодировку UTF-8 без BOM.

После приема чека регистратор проводит следующие действия:

Вариант работы с модулем СКНО:

1. Проверяет наличие бумаги. При её отсутствии возвращается ошибка и дальнейшая обработка чека не производится. Наличие возможности печати чека можно проверить чтением ресурса **/cgi/state** позволяет получить объект текущего состояния регистратора. Для успешной печати чека поле **FskMode** объекта состояния должно быть равно **1** и поле **err** должно быть пустым.
2. Проводятся проверки на возможность работы с модулем СКНО.
3. Анализирует чек на наличие ошибок. При их наличии, ошибка возвращается пользователю, и чек не печатается.
4. Посылает документ в модуль СКНО. После получения ответа, запрещающего работу, чек не печатается.
5. Печатает чек. При возникновении ошибки печати чек оканчивается успешно. Возвращается JSON содержимое чека. Объект чека имеет поле **pending** со значением **1**. При последующем устранении ошибки печати чек печатается заново. При необходимости после окончания печати кассир может выдать копию чека.
6. После успешного окончания чека возвращается JSON содержимое чека.

Вариант работы без модуля СКНО:

1. Проверяет наличие бумаги. При её отсутствии возвращается ошибка и дальнейшая обработка чека не производится. Наличие возможности печати чека можно проверить чтением ресурса **/cgi/state** позволяет получить объект текущего состояния регистратора. Для успешной печати чека поле **FskMode** объекта состояния должно быть равно **1** и поле **err** должно быть пустым.
2. Анализирует чек на наличие ошибок. При их наличии, ошибка возвращается пользователю, и чек не печатается.
3. Начинает печать чека. При возникновении ошибок печати – возвращается ошибка отсутствия бумаги, чек прекращает печататься и отменяется.

4. Печатает фискальное окончание чека (номер чека, серийный номер, фискальный логотип). При возникновении ошибки печати чек оканчивается успешно. Возвращается JSON содержимое чека. Объект чека имеет поле **pending** со значением 1. При последующем устранении ошибки печати чек допечатывается и печатается квитанция о прерывании печати. При необходимости после окончания печати кассир может выдать копию чека.
7. После успешного окончания чека возвращается JSON содержимое чека.

Текстовый формат чека.

Первая строка документа может определять вид распечатанного чека.

Содержимое строки	Вид чека
(F)	фискальный чек
(R)	возвратный чек
(I)	чек внесения-изъятия денег из сейфа
(V)	чек аннулирования
(P)	не фискальный чек
(L)	копия чека
(O)	пустой чек

если в первой строке документа не найден тип чека, этот чек считается фискальным.

Строка продажи.

Для чеков типа **F, R**.

[Количество*]Цена "Название" Код [[Налог] [[Секция] [[Группа]
[Тип кода]]]

Поле	Описание
Количество	Необязательное поле. Должно быть цифрой с не более чем тремя знаками после десятичной точки. Если поле пропущено, количество принимается равным 1.
Цена	Должно быть цифрой с не более чем двумя знаками после десятичной точки.
Название	Название товара. Если название содержит символ ", он удваивается.
Код	Код товара из не более чем 13 цифр
Налог	Необязательное поле. Номер налоговой ставки из таблицы Tax . 0 - без налога. При отсутствии поля продажа происходит с первой налоговой ставкой из таблицы налогов.
Секция	Необязательное поле. Номер секции из таблицы Dep . При отсутствии поля продажа происходит в первую секцию.
Группа	Необязательное поле. Номер группы из таблицы Grp . При отсутствии поля продажа происходит в первую группу.
Тип кода	Необязательное поле. Тип кода товара (0 – без EAN, 1 – EAN, 2 - услуга). При отсутствии поля продажа происходит с типом 1.

Примеры:

2*10.50 "йогурт" 300 1

20 "колбаса" 2306547

Строка скидки.

Для чеков типа F, R.

+/- [=] число% [;номер скидки]

Символ	Значение
+	Наценка
-	Скидка
=	на промежуточный итог
%	Процентная
номер скидки	номер подтверждения льготной скидки. (Согласно требованиям НацБанка к системе передачи данных в ДПА)

Пример	Описание
-10	Скидка 10 грн на предыдущий товар
-2.5%	Скидка 2.5% на предыдущий товар
+=20	Наценка 20 грн на промежуточный итог
+=5.5%	Наценка 5.5% на промежуточный итог

Строка оплаты.

\$ [сумма] [; [номер оплаты] [;номер транзакции[;номер карты]]]

Поле	Описание
Сумма	Сумма оплаты. Если сумма отсутствует, считается, что она соответствует сумме всего чека.
Номер оплаты	Номер строки в таблице Pay , который соответствует типу выбранной оплаты. При отсутствии происходит оплата наличными.
Номер транзакции	Номер транзакции (RRN) при осуществлении оплаты через банковский терминал.(Согласно требованиям НацБанка к системе передачи данных в ДПА)
Номер карты	Номер банковской карты при оплате через терминал

Строка внесения-изъятия денег.

Для чеков типа I.

\$ [-] сумма [;номер оплаты]

Только для типа чека (IO).

Поле	Описание
-	При наличии операция является операцией изъятия. При отсутствии операция является операцией внесения.
Сумма	Сумма внесения или изъятия.
Номер оплаты	Номер строки в таблице Pay , который соответствует типу выбранной оплаты. При отсутствии производится внесение или изъятие наличных.

Строка текстового комментария.

Для чеков типа F, R, P и I.

Для чеков типа **F, R и I** комментарии, по желанию пользователя, могут вноситься или не вноситься в электронную ленту продаж. Строки комментариев, внесенные на ленту продаж, будут печататься в копии чека и отсылаться в копии чека в ГНА. Строки комментариев, не внесенные в ленту, будут распечатаны только на оригинале чека.

Для нефискальных чеков никакие комментарии в ленту продаж не вносятся.

Строка комментариев, которая будет внесена в ленту продаж, должна начинаться символом **#**.

#строка комментария, сохраняемая в ленту

любая другая строка чека, не начинающаяся с символов **0-9+-x\$%<**, является строкой комментариев, не вносимых в ленту. Если комментарий является самой первой строкой фискального чека, его нельзя начинать также символами **{{**.

Строка, начинающаяся с **"**, всегда является комментарием, не вносимым в ленту.

"строка комментария, не вносимая в ленту

Первый символ, который является признаком комментария, не выводится на чек.

Печать штрих-кода.

Строка, начинающаяся с **<** считается строкой содержащей информацию о штрихкоде. Формат строки:

<"код" [тип [ширина [высота [отступ]]]]

Поле	Описание
Код	Текст штрих-кода 1-24 символа
Тип	Тип штрих-кода 1 – EAN-13 (по умолчанию) 2 – CODE-128 3 – CODE-39
Ширина	Ширина символа в пикселях 2(по умолчанию),3,4
Высота	Высота штрих-кода в пикселях 1-150. По умолчанию 60
Отступ	Отступ до и после штрих-кода в пикселях 0-30. По умолчанию 20.

Если код выходит за границы печатаемой области, чек отменяется с ошибкой **xС2** (Неверный параметр у процедуры).

Аннулирование документа.

Строка, начинающаяся с **>** считается строкой содержащей информацию о номере аннулируемого документа. Формат строки:

>номер

Поле	Описание
номер	Номер аннулируемого документа

Коррекция документа.

Строка, начинающаяся с @ считается строкой содержащей информацию о коде отменяемого товара. Отменяется первая с начала чека продажа товара с указанным кодом (если код не указан, то отменяется последняя операция). Формат строки:

@ [код]

Поле	Описание
код	Код отменяемого товара

Формат чека JSON.

Чек представляет собой объект с одним из полей:

Имя поля	Вид чека
F	фискальный чек
R	возвратный чек
IO	чек внесения-изъятия денег из сейфа
VD	чек аннулирования
P	нефискальный чек
L	копия чека или ресторанного счета
RO	ресторанный заказ
VB	отмена ресторанного заказа

Содержимое поля представляет собой массив строк чека. Пустой чек печатается последовательностью {}. Регистратор ожидает символ '{' в качестве первого символа данных POST для чеков в формате JSON. Иначе данные интерпретируются как данные в текстовом формате.

Строка чека представляет собой объект с одним из полей:

Имя поля	Вид строки	Может находиться в чеках с типом
S	Строка продажи	F,R,RO
D	Строка скидки	F,R,RO
P	Строка оплаты	F,R
IO	Строка служебного внесения/изъятия средств	IO
C	Строка текстового комментария	F,R,IO,P,RO
N	Строка нефискального комментария	F,R,IO,P,RO
BC	Штрих-код	F,R,IO,P,RO
VD	Аннулирование документа	VD
VP	Коррекция документа	F,R,RO
OB	Открытие нового счета	RO
NO	Открытие нового заказа	RO
CB	Закрытые счета	F
VB	Отмена счета	VB

Строка продажи.

Значение поля S есть объект с полями:

Поле	Описание
qty	Необязательное поле. Должно быть цифрой с не более чем тремя знаками после десятичной точки. Если поле пропущено, количество принимается равным 1.
price	Должно быть цифрой с не более чем двумя знаками после десятичной точки.
name	Название товара.
code	Код товара из не более чем 13 цифр
tax	Необязательное поле. Номер налоговой ставки из таблицы Tax . 0 - без налога. При отсутствии поля продажа происходит с первой налоговой ставкой из таблицы налогов.
ctype	Необязательное поле. Тип кода товара (0 – без EAN, 1 – EAN, 2 - услуга). При отсутствии поля продажа происходит с типом 1.
dep	Необязательное поле. Номер секции из таблицы Dep . При отсутствии поля продажа происходит в первую секцию.
grp	Необязательное поле. Номер группы из таблицы Grp . При отсутствии поля продажа происходит в первую группу.

Строка скидки.

Значение поля **D** есть объект с полями:

Поле	Описание
sum	сумма скидки или наценки. Положительное число описывает наценку. Отрицательное - скидку. Если поле присутствует в объекте, значит, строка чека является скидкой на сумму. Для процентных скидок поле должно отсутствовать.
prc	процент скидки или наценки. Положительное число описывает наценку. Отрицательное - скидку. Поле игнорируется, если объект имеет поле sum .
all	если присутствует и ненулевое - скидка производится на промежуточный итог.
dn	номер подтверждения льготной скидки. (Согласно требованиям НацБанка к системе передачи данных в ДПА)

Строка оплаты.

Значение поля **P** есть объект с полями:

Поле	Описание
sum	Сумма оплаты. Если сумма отсутствует, считается, что она соответствует сумме всего чека.
no	Номер строки в таблице Pay , который соответствует типу выбранной оплаты. При отсутствии, происходит оплата наличными.

Строка внесения-изъятия денег.

Значение поля **IO** есть объект с полями:

Поле	Описание
type	"IO" - строка описывает служебное движение денег
sum	Сумма внесения или изъятия. Сумма внесения должна быть положительна. Изъятия - отрицательна.
no	Номер строки в таблице Pay , который соответствует типу выбранной оплаты. При отсутствии производится внесение или изъятие наличных.

Строка текстового комментария.

Значение поля **C** есть объект с полями:

Поле	Описание
Cm	Строка текстового комментария.

Строка нефискального комментария.

Нефискальные комментарии не вносятся в ленту продаж. Строки комментариев, внесенные на ленту продаж (**C**), будут печататься в копии чека и отсылаться в копии чека в ГНА. Строки комментариев, не внесенные в ленту (**N**), будут распечатаны только на оригинале чека.

Значение поля **N** есть объект с полями:

Поле	Описание
Cm	Строка нефискального комментария.
Attr	"w" – символы двойной ширины "h" – символы двойной высоты "l" – символы половинной высоты Допускаются комбинации "wh" и "wl".

Печать штрих-кода.

Значение поля **BC** есть объект с полями:

Поле	Описание
code	Текст штрих-кода 1-24 символа
type	Тип штрих-кода 1 – EAN-13 (по умолчанию) 2 – CODE-128 3 – CODE-39
width	Ширина символа в пикселях 2(по умолчанию),3,4
height	Высота штрих-кода в пикселях 1-150. По умолчанию 60
feed	Отступ до и после штрих-кода в пикселях 0-30. По умолчанию 20.

Если код выходит за границы печатаемой области, чек отменяется с ошибкой xC2 (Неверный параметр у процедуры).

Строка аннулирования документа.

Значение поля **VD** есть объект с полями:

Поле	Описание
no	Номер аннулируемого документа.

Коррекция документа.

Значение поля **VP** есть объект с полями:

Поле	Описание
code	Код товара (не более чем 13 цифр). Отменяется первая с начала чека продажа товара с указанным кодом (если код не указан, то отменяется последняя операция)

Копия документа или ресторанного счета.

Значение поля **L** есть объект с одним из полей

Поле	Описание
bill	Номер открытого счета для копии(1...9999)
no	Номер документа

Открытие ресторанного счета.

Значение поля **NB** есть объект с полями:

Поле	Описание
table	Номер столика (1...65000).
place	Номер места за столиком (1...250, по умолчанию 1)

Новый ресторанный заказ.

Значение поля **NO** есть объект с полями:

Поле	Описание
bill	Номер счета (1...9999)

Отмена ресторанного счета

Значение поля **VB** есть объект с полями:

Поле	Описание
bill	Номер счета (1...9999)

Заккрытие ресторанного счета

Значение поля **CB** есть объект с полями:

Поле	Описание
bill	Номер счета (1...9999)

Примеры чеков в формате JSON.

Чек продажи

```
{"F":[
  {"C":{"cm":"Кассир: Светлана"}},
  {"S":{"code":"1","price":"5","name":"Конфета"}},
  {"S":{"code":"2","price":"15","name":"Печенье","qty":"0.5"}},
  {"D":{"prc":"5","all":"1"}},
  {"P":{}}
]}
```

Чек служебного внесения денег

```
{"IO":[{"C":{"cm":"Кассир: Светлана"}}, {"IO":{"sum":"120"}}, {"IO":{"sum":"-140", "no":"2"}}, ]} {"P":[{"C":{"cm":"Line 1"}}, {"C":{"cm":"Line 2", "attr":"w"}}, {"C":{"cm":"Line 3"}}]} {"P":[{"N":{"cm":"Line 1"}}, {"N":{"cm":"Line 2"}}, {"N":{"cm":"Line 3"}}]}
```

Открытие ресторанного счета

```
{"RO":{"NB":{"table":"1", "place":"1"}}, {"S":{"code":"1", "price":"5", "name":"Конфета"}}, {"S":{"code":"2", "price":"15", "name":"Печенье", "qty":"0.5"}}}
```

Новый ресторанный заказ

```
{"RO":{"NO":{"bill":"1"}}, {"S":{"code":"3", "price":"12", "name":"Кофе"}}}
```

Отмена ресторанного счета

```
{"VB":{"VB":{"bill":"1"}}}
```

Коррекция ресторанных заказов

Новый ресторанный заказ

```
{"RO":{"NO":{"bill":"1"}}, {"VP":{"code":"2"}}, {"S":{"code":"4", "price":"10", "name":"Чай"}}}
```

Заккрытие ресторанного счета

```
{"F":{"CB":{"bill":"1"}}, {"P":{}}}
```

Чтение и синхронизация чековой ленты.

Прочитать имеющуюся в аппарате чековую ленту можно HTTP методом **GET** из ресурса **/cgi/chk**.

Формат ленты.

Чековая лента представляет собой JSON массив с объектами ленты. Эти объекты имеют следующие поля:

Поле	Описание
L	регистрация кассира
Z1	Номер дневного отчета
F	Строки фискального чека
R	строки чека возврата
IO	строки чека служебного движения средств
VD	строки чека аннулирования
RO	строки ресторанных заказов
datetime	Дата и время объекта в ленте. Время приводится в формате Unix-time (количество секунд с начала 1970 года по Гринвичу).
id	уникальный идентификатор объекта в чековой ленте. Это число которое у каждого следующего объекта ленты больше, чем у предыдущего.
oper_id	номер оператора, связанный с объектом ленты
bill	номер ресторанный счета

У объектов, которые представляют чеки, имеются дополнительные поля:

Поле	Описание
no	Номер чека.
Pending	Признак того, что чек еще не напечатан. Необязательное поле. При его отсутствии считается, что чек напечатан.

Представление строк чеков.

Строка чека это объект с одним из полей со следующими именами

Поле	Описание
S	строка описывает продажу
D	Строка описывает скидку
P	Строка описывает оплату
IO	Строка описывает операцию внесения-изъятия
VD	Строка описывает аннулирование документа
C	Строка описывает текстовый комментарий
CB	Строка описывает закрытие ресторанный счета
NO	Строка описывает новый ресторанный заказ
VB	Строка описывает отмененный счет

Содержимое каждого из полей это объект, детализирующий операцию

Строка продажи.

Поле	Описание
qty	Необязательное поле. Должно быть цифрой с не более чем тремя знаками после десятичной точки. Если поле пропущено, количество принимается равным 1.
price	Должно быть цифрой с не более чем двумя знаками после десятичной точки.

name	Название товара.
code	Код товара из не более чем 13 цифр
tax	Необязательное поле. Номер налоговой ставки из таблицы Tax . 0 - без налога. При отсутствии поля продажа происходит с первой налоговой ставкой из таблицы налогов.
sum	Общая сумма строки
ctype	Тип кода товара (0 – без EAN, 1 – EAN, 2 - услуга
dep	Номер секции из таблицы Dep .
grp	Номер группы из таблицы Grp .

Строка скидки.

Поле	Описание
sum	Общая сумма скидки или наценки. Положительное число описывает наценку. Отрицательное - скидку. Это общая сумма строки чека.
prc	процент скидки или наценки. Положительное число описывает наценку. Отрицательное - скидку. Если поле присутствует - скидка процентная. Если отсутствует - скидка сумовая.
subt	сумма на которую начисляется скидка. Если поле присутствует - скидка сделана на промежуточный итог.

Строка оплаты.

Поле	Описание
sum	Сумма оплаты.
no	Номер строки в таблице Pay , который соответствует типу выбранной оплаты.

Строка внесения-изъятия денег.

Только для типа чека (IO).

Поле	Описание
sum	Сумма внесения или изъятия. Сумма внесения должна быть положительна. Изъятия - отрицательна.
no	Номер строки в таблице Pay , который соответствует типу выбранной оплаты. При отсутствии производится внесение или изъятие наличных.

Строка аннулирования документа.

Только для типа чека (VD).

Поле	Описание
sum	Сумма аннулированного документа.
no	Номер аннулированного документа.

Строка текстового комментария.

Поле	Описание
------	----------

Cm	Строка комментария
-----------	--------------------

Строка нового ресторанного заказа

Поле	Описание
bill	Номер счета
table	Номер столика
place	Номер места
order	Номер заказа

Строка закрытия ресторанного счета

Поле	Описание
bill	Номер счета
table	Номер столика
place	Номер места
orders	Количество заказов в счете

Строка отмены ресторанного счета

Поле	Описание
bill	Номер счета
table	Номер столика
place	Номер места
orders	Количество заказов в счете

Определение факта изменения чековой ленты.

Изменение чековой ленты можно обнаружить периодическим чтением состояния регистратора **/cgi/state**. Поля состояния, которые необходимо контролировать для синхронизации чековой ленты.

Название	Описание
serial	Заводской номер аппарата. Несколько последовательных запросов статуса должны давать одинаковое значение заводского номера. В противном случае IP адрес аппарата сменил хозяина между запросами.
JrnTime	Время начала чековой ленты. Если между двумя последовательными запросами это время изменилось, то в аппарате появилась новая чековая лента и ее надо считывать сначала. Если время чековой ленты равно 0 - лента пуста.
chkid	Идентификатор последнего объекта в ленте. Если идентификатор объекта больше чем идентификатор последнего объекта ленты и серийный номер аппарата со временем ленты совпадает с предыдущим, можно дочитать ленту.

Дочитка ленты происходит HTTP методом GET по адресу **/cgi/chk?id=_id**, где **_id** - номер последнего считанного объекта этой ленты. Метод вернет все объекты ленты следующие за указанным.

Прочитать определенный документ из ленты можно HTTP методом GET по адресу **/cgi/chk?no=_num**, где **_num** - номер необходимого документа.

Определение степени заполненности электронной ленты.

Электронная чековая лента имеет определенный фиксированный размер, который может зависеть от модели аппарата. Лента разделена на записи фиксированного размера. Чек разной длины занимает в ней разное число записей. Размер чека в записях можно вычислить как разницу между значениями поля **id** этого и предыдущего чека. Свободное место в ленте можно проверить вызвав процедуру `/cgi/proc/getjrnroom`. Процедура возвращает объект с полями **Total** и **Used**, в которых передается общее количество записей в ленте и использованное количество записей. Вычислив оставшееся количество записей и разделив его на размер среднего чека в ленте можно получить представление об оставшемся в ней месте. Лента очищается печатью дневного отчета (процедура `/cgi/proc/printreport?0`). В этот момент регистратор проводит следующие действия:

1. Передаются данные отчета в СКНО
2. Печатает текст отчета.
3. Переносит итоги дня в фискальную память.

Идентификация устройства.

Информацию об устройстве можно получить, прочитав ресурс `/cgi/dev_info`. Аппарат возвращает JSON объект со следующими полями:

Поле	Описание
dev_zn	Заводской номер
dev_ver	Версия ПО
dev_dat	Дата прошивки устройства
skno_num	Номер СКНО
reg_num	Регистрационный номер
unp_num	Уникальный номер плательщика
prot	Версия протокола аппарата. Поле отсутствует для аппаратов, в которых номер оплаты при печати чеков на единицу меньше, чем id соответствующего вида оплаты в таблице <code>/cgi/tbl/Pay</code> Значение поле равно 1 для всех остальных случаев

Регистрация пользователя и список пользователей. Регистрация по IP.

Получить информацию от аппарата, выбить чек или совершить любое другое действие с аппаратом через веб интерфейс можно только пройдя аутентификацию. Веб сервер аппарата поддерживает [дайджест аутентификацию HTTP \(Wikipedia\)](#).

Пример.

```
GET /cgi/state HTTP/1.1
Host: 192.168.8.2
User-Agent: test
```



```

HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest realm="HTROM",
nonce="527b004c29df30afd42c9dbf43dcb6d9", opaque="60DB81DD",
stale="FALSE", algorithm="md5", qop="auth"
Content-Type: text/html
Content-Length: 248

<html><head><title>LW-HTTP server error</title></head><body
bgcolor="white" text="black"><table width="100%"><tr
valign="top"><td width="20"></td><td width="500"><h1>LW-HTTP server
error</h1><h2>401 Unauthorized</h2></td></tr></table></body></html>

GET /cgi/state HTTP/1.1
Host: 192.168.8.2
Authorization: Digest username="service", realm="HTROM",
nonce="527b004c29df30afd42c9dbf43dcb6d9", uri="/cgi/state",
algorithm=MD5, response="3f4f49f5adefafdf19ce9148103486af",
opaque="60DB81DD", qop=auth, nc=00000001, cnonce="669bcf2a9b1c9deb"
User-Agent: test

```

```

HTTP/1.1 200 OK
Server: host name
Content-Type: application/json
Cache-Control: no-cache

{"model": "ТИТАН-
А", "name": "TA0000001", "serial": "TA0000001", "time": 1481202440, "chkId":
:0, "JrnTime": 0, "currZ": 0, "IsWrk": 0, "Fiscalization": 1, "FskMode": 1, "SK
NOState": 2626,
  "err": []}

```

Все основные библиотеки работы с HTTP поддерживают этот протокол аутентификации и позволяют указывать имя пользователя и пароль для каждого HTTP запроса.

Для печати чеков и редактирования программных установок аппарата имена операторов и пароли берутся из таблицы **Oper** аппарата. В качестве имени пользователя берется номер оператора из таблицы. Поля таблицы **Oper**:

Поле	Описание
id	Номер оператора
Name	Имя оператора
Pswd	Пароль оператора

Для проведения служебных действий: фискализации, изменения времени, проведения технологического сброса, доступа к таблице **Oper** необходима регистрация под именем пользователя **service**, пароль которого фиксирован и зависит от конкретной модели аппарата.

Аппарат ведет список IP адресов, с которых авторизация к ресурсам **/cgi/*** не требуется. Список адресов находится в таблице по адресу **/cgi/tbl/whitelP**. В таблице фиксированное число строк. Поля таблицы:

Поле	Описание
id	идентификатор строки
ip	IP адрес. При адресе 0.0.0.0 строка считается свободной.
login	имя оператора, который считается зарегистрированным по этому адресу.

Если таблица **whitelP** не заполнена до конца, существует упрощенный способ добавить туда строку - вызов процедуры **/cgi/proc/register**. Вызов процедуры приводит к сохранению в свободной строке процедуры IP адреса, с которого она была вызвана, и оператора, от имени которого она была вызвана. Если IP адрес уже присутствует в таблице, номер оператора в строке с ним меняется на номер оператора, который вызвал процедуру. Процедура с параметром **/cgi/proc/register?clear** освобождает строку, если IP адрес был записан ранее.

Фискальная память.

Процедура фискализации.

Процедура фискализации проводится на нефискальном устройстве. Все указанные ниже операции возможны только от имени пользователя **service**. Для фискализации нужно провести следующие шаги:

Шаг	Описание
1	Установка текущего времени. Процедура /cgi/proc/setclock с параметром текущего времени в формате ISODate. Например: /cgi/proc/setclock?2014-10-05T15:56:07
2	Установка номеров СКНО, регистрационного, и уникального номера плательщика через таблицу Fsk .
3	Установка позиции десятичной точки в таблице Fsk , поле NumPnt .
4	Установка налоговых ставок в таблице Tax .
5	Вызов процедуры фискализации /cgi/proc/fiscalization .

Таблица Hdr

Поле	Описание
id	номер строки
Param	1 - обычный шрифт. 2 - выделенный шрифт.
Line	Текст строки

В фискализированной кассе для внесения изменений в 4-5-ю строку заголовка в фискальную память необходимо вызвать процедуру **/cgi/proc/puthdrfm**.

Таблица Tax

Поле	Описание
Id	номер налога
Prc	процент налога
TaxName	Название налога

В фискализированной кассе для записи налоговых ставок и положения десятичной точки (NumPnt таблица Fsk) в фискалку используется вызов процедуры **/cgi/proc/puttaxfm**.

Таблица Fsk

Поле	Описание
NumSKNO	Номер СКНО
RegNum	Регистрационный номер
UNP	Уникальный номер плательщика
NumPnt	Позиция десятичной точки в денежных суммах
NumChk	Номер первого чека в открытой смене

Чтение информации из фискальной памяти. Таблица FDay.

Прочитать информацию из фискальной памяти можно по адресу `/cgi/tbl/FDay`. Поскольку в заполненном аппарате фискальных записей может быть несколько тысяч, предусмотрен режим частичного чтения таблицы по адресу `/cgi/tbl/FDay?s=startIdx&e=endIdx`, где **startIdx** - индекс первого возвращаемого отчета, а **endIdx** - индекс отчета, следующего за последним возвращаемым. Например, команда `/cgi/tbl/FDay?s=1&e=4` возвращает отчеты 1,2,3.

Если в запросе не задан начальный индекс, он считается равным 1. Если не указан конечный индекс, он равен индексу отчета, следующему за последним. Например команда `/cgi/tbl/FDay?e=4` вернет отчеты 1,2,3. Команда `/cgi/tbl/FDay?s=4` вернет отчеты от четвертого до последнего включительно.

Команда воспринимает и отрицательные индексы. Так, -1 - это индекс последнего отчета, -2 - предпоследнего и т.д. Например, команда `/cgi/tbl/FDay?s=-1` вернет только последний отчет.

Поля таблицы FDay.

Поле	Описание
id	номер записи в фискальной памяти
valid	Флаг действительной записи 1-действительна, 0-нет.
Date	Дата записи в Unix-time
STax0-STax4	Сумма, подлежащая налогообложению по соответствующим ставкам.
SRet0-SRet4	Сумма, подлежащая налогообложению в возвратных чеках.
NChk	Количество чеков продаж
NRet	Количество возвратных чеков

Строки в таблицу добавляются вызовом процедуры `/cgi/proc/printreport?0`

Свободное место в фискальной памяти можно проверить, вызвав процедуру `/cgi/proc/getfmroom`. Процедура возвращает объект с полями **Total** и **Used**, в которых передается общее количество записей в фискальной памяти и использованное количество записей.

Поля таблицы FTax.

Эта таблица содержит данные о налоговых ставках и времени их введения в аппарате. Строки в таблицу добавляются вызовом процедуры `/cgi/proc/puttaxfm`

Поле	Описание
id	номер строки
valid	Флаг действительной записи 1-действительна, 0-нет.
Date	Дата записи в Unix-time

NPnt	Позиция десятичной точки в суммах
Tax1-Tax4	Ставки налогов
TaxName1-TaxName4	Названия налогов.
NDay	Номер дневного отчета, после которого записана запись.

Настройка сети TCP/IP.

Настройка сетевых интерфейсов аппарата производится в таблице TCP.

Поле	Описание														
id	Номер интерфейса.														
Addr	IP адрес адаптера. Значение "0.0.0.0" говорит о том, что адрес будет определяться автоматически с использованием протоколов DHCP и AutoIP.														
Mask	Маска сети														
Gate	Шлюз по умолчанию														
DNS	Адрес DNS сервера														
MAC	MAC адрес адаптера														
AdptFlg	Флаги адаптера <table border="1"> <tr> <th>Бит</th><th>Значение</th></tr> <tr> <td>1</td><td>разрешена скорость 10 Mbps</td></tr> <tr> <td>2</td><td>разрешена скорость 100 Mbps</td></tr> <tr> <td>3</td><td>разрешена скорость 1000 Mbps</td></tr> <tr> <td>4</td><td>разрешить дуплекс</td></tr> <tr> <td>5</td><td>автоматический режим</td></tr> <tr> <td>6</td><td>разрешить определение MDI/MDI-X</td></tr> </table>	Бит	Значение	1	разрешена скорость 10 Mbps	2	разрешена скорость 100 Mbps	3	разрешена скорость 1000 Mbps	4	разрешить дуплекс	5	автоматический режим	6	разрешить определение MDI/MDI-X
Бит	Значение														
1	разрешена скорость 10 Mbps														
2	разрешена скорость 100 Mbps														
3	разрешена скорость 1000 Mbps														
4	разрешить дуплекс														
5	автоматический режим														
6	разрешить определение MDI/MDI-X														

Если аппарат не виден в сети, существует возможность сброса таблицы TCP в одно из двух предустановленных состояний. Одно из них с динамическим определением адреса. В этом случае поле **Addr** устанавливается в "0.0.0.0" и аппарат будет пытаться получить адрес из сервера DHCP сети в которую он будет подключен. В другом случае аппарату присваивается адрес "192.168.8.2" с маской "255.255.255.0", что требует установки адреса из этой же подсети на устройстве, которое будет связываться с аппаратом. Как конкретно сброс делается на данной модели аппарата, описано в его инструкции по эксплуатации.

Установка имени аппарата в видимого по протоколам DHCP, UPnP и Syslog происходит через поле **Name** таблицы **Host**. По умолчанию имя равно серийному номеру аппарата.

Настройка выдачи диагностических сообщений по протоколу Syslog происходит через таблицу **SysLog**. Можно настроить:

Поле	Описание
Addr	Адрес или имя сервера syslog, куда будут отсылаться сообщения
Port	Порт сервера syslog
Priority	Фильтр выводимых сообщений.

Обнаружение аппарата по протоколу SSDP.

Устройство может быть обнаружено в сети по протоколу [SSDP](#). Для этого нужно послать UDP пакет на групповой адрес **239.255.255.250:1900**. Содержание пакета:

M-SEARCH * HTTP/1.1
Host:239.255.255.250:1900
ST:тип устройства
Man:"ssdp:discover"
MX:таймаут

Таймаут указывается в секундах. Поддерживаются следующие типы устройств:

Тип	Описание
urn:rootdevice	Все устройства работающие по UPnP. Кроме кассовой техники на запрос могут ответить UPnP роутеры и устройства, проигрывающие мультимедиа, если они присутствуют в локальной сети.
urn:help-micro.kiev.ua:device:ecr:1	Кассовые аппараты разработки Хелп Микро
urn:help-micro.kiev.ua:device:uafpn:1	Фискальные регистраторы для Украины разработки Хелп Микро
urn:help-micro.kiev.ua:device:webdev:1	Устройства с веб интерфейсом разработки Хелп Микро, включая и кассовые аппараты и фискальные регистраторы.

На этот пакет отправят ответ все устройства указанного типа. Если ответ начинается на HTTP/1.1 200 OK, то по адресу, с которого он отослан, находится искомое устройство.

Ниже приводится текст класса Qt который издает соответствующий запрос и обрабатывает ответы.

```
class SsdpPacket : public QMap<QString,QString> {
public:
    SsdpPacket(QString msg);
};

class SsdpQuery : public QUdpSocket {
    Q_OBJECT
public:
    SsdpQuery(QString what, QObject* parent = 0);
    int send(int waitSec);
protected:
    QString name;
signals:
    void Notify(SsdpPacket* p);
private slots:
    void processDatagrams();
};    void processDatagrams();
};

SsdpPacket::SsdpPacket(QString msg) {
    QStringList l = msg.split("\r\n");
    foreach(const QString& s, l) {
```

```

        QString key = s.section(":", 0, 0).trimmed();
        QString val = s.section(":", 1, 1).trimmed();
        if (val.length() && key.length()) { insert(key, val);
        }
    }
}

SsdpQuery::SsdpQuery(QString what, QObject*
parent):QUdpSocket(parent) {
    bool ret = bind(1901);
    ret = connect(this, SIGNAL(readyRead()), this,
SLOT(processDatagrams()));
    name = what;
}

int SsdpQuery::send(int waitSec) {
    QHostAddress groupAddress = QHostAddress("239.255.255.250");
    QByteArray datagram = QString("M-SEARCH *
HTTP/1.1\r\nHost:239.255.255.250:1900\r\nST:%1\r\nMan:\\"ssdp:discove
r\\"\r\nMX:%2\r\n\r\n").arg(name).arg(waitSec).toAscii();
    ret =
writeDatagram(datagram.data(), datagram.size(), groupAddress, 1900);
}

void SsdpQuery::processDatagrams() {
    while(hasPendingDatagrams()) {
        QByteArray datagram;
        datagram.resize(pendingDatagramSize());
        readDatagram(datagram.data(), datagram.size());
        if (datagram.indexOf("HTTP/1.1 200 OK\r\n")!=0) return;

        SsdpPacket *p = new SsdpPacket(datagram);
        if ((*p)["ST"].contains(name)) { emit Notify(p);
        } else { delete p;
        }
    }
}
}

```

Настройка видов оплат.

Настройка видов оплат производится в таблице `/cgi/tbl/Pay`. Таблицу можно изменять только при пустой ленте продаж. В таблице присутствуют следующие поля:

Поле	Описание	
id	Порядковый номер вида оплаты	
Param	Флаги свойств оплаты	
	Величина	Описание
	1	вид оплаты разрешен

	2	по виду оплаты возможна сдача
	16	вид оплаты не лежит в сейфе
	32	вид оплаты может использоваться для оплаты возвратных чеков
	64	безналичный вид оплаты
Значение поля является суммой необходимых флагов		
Name	Название вида оплаты	

Настройка внешнего вида чека.

Таблицы Hdr/Ftr.

Таблицы заголовка чека (/cgi/tbl/Hdr) и таблица окончания чека (/cgi/tbl/Ftr) используются для формирования соответствующих частей чека и имеют следующую структуру:

Поле	Описание
id	номер строки
Param	1 - обычный шрифт. 2 - выделенный шрифт.
Line	Текст строки

Таблицы можно изменять только при пустой ленте продаж.

Логотип.

Ресурс /cgi/logo.bmp позволяет загрузить в него HTTP методом POST монохромный рисунок в формате BMP который будет использоваться в качестве графического логотипа чека. HTTP методом GET можно выкачать текущий используемый логотип. Размеры загружаемого рисунка не должны превышать размеры рисунка, который возвращается кассой.

Настройка принтера.

Настройка контрастности (**PrContr**) и равномерности (**PrEqual**) печати находятся в таблице **Adm**.

Некоторые флаги, влияющие на печать, находятся в таблице **Flg**.

Поле **PrintOff**.

Флаг	Описание
1	выключить принтер
2	печать заголовка следующего чека сразу после окончания печати предыдущего
4	включение обрезчика
8	использовать полную обрезку

Значение поля является суммой используемых флагов.

Поле **Flg1**.

Флаг	Описание
1	Следить за запасом товара
2	Не используется
4	Следить за дробным количеством
8	Не используется
16	Сортировать товары в отчетах
32	Не используется
64	Не используется
128	Не используется

256	Разрешить продажи с 0-й ценой
512	Удалять после Z1 товары с 0-м запасом
1024	Удалять после Z1 все товары
2048	Сквизирование Z отчетов
4096	Автоматический вывод Z2 после Z1
8192	Не используется
16384	Не используется
32768	Не используется

Значение поля является суммой используемых флагов.

Поле **Flg3**.

Флаг	Описание
1	Печатать код товара в чеке
2	Печатать имя отдела в чеке
4	Печатать номер отдела в чеке
8	Не используется
16	Не выводить Z1 отчет без инкассации всех наличных денег
32	Не используется
64	Не печатать имя кассира
128	Не используется
256	Не печатать логотип пользователя
512	включить ресторанный режим
1024	Работа с заказами любым кассиром
2048	Не печатать копии заказов для разноса на кухню
4096	Не используется
8192	Не используется
16384	Не используется
32768	Печатать сумму налога после каждой продажи

Значение поля является суммой используемых флагов.

В поле **Feed** таблицы **Flg** указывается количество строк, которые принтер прогоняет между чеками.

Печать и получение отчетов.

Отчеты аппарата печатаются процедурой `/cgi/proc/printreport?no`, где **no** тип отчета

Отчеты из фискальной памяти печатаются процедурой

`/cgi/proc/printfmreport?RepID&BegDate&EndDate&BegZRep&EndZRep`, где

RepID	Вид фискального отчета 1 - полный отчет по датам 2- полный отчет по номерам Z отчетов 3 - краткий отчет по датам 4 - краткий отчет по номерам Z отчетов
BegDate EndDate	Начальная и конечная дата отчета (RepID 1 или 3)
BegZRep EndZRep	Начальный и конечный номер Z отчета (RepID 2 или 4)

Получить данные по остатку денег в кассе можно из ресурса `/cgi/rep/pay`. Возвращается массив объектов с полями **no** - номер оплаты и **sum** - остаток по этому виду оплаты. Остаток считается как разница служебных внесений и изъятий плюс сумма выторга. Для видов оплаты, которые не находятся в сейфе и для которых не имеют смысла операции служебного движения средств в качестве суммы выступает просто сумма выторга этим видом оплаты.

Приложение 1. Список ошибок кассы.

Код	Описание
x01	Цена не указана
x02	Количество не указано
x03	Отдел не указан
x04	Группа не указана
x25	Нет бумаги
x31	Пользователь уже зарегистрирован
x32	Неверный пароль
x33	Неверный номер таблицы
x34	Доступ к таблице запрещен
x35	Умолчание не найдено
x36	Неверный индекс
x37	Неверное поле
x38	Таблица переполнена
x39	Неверная длина двоичных данных
x3A	Попытка модификации поля только для чтения
x3B	Неверное значение поля
x3C	Товар уже существует
x3D	По товару были продажи
x3E	Запрос запрещен
x3F	Неверная закладка
x40	Ключ не найден
x41	Процедура уже выполняется
x42	Количество товара отрицательно
x8A	Нет бумаги для контрольной ленты
x8B	Нет бумаги
x8D	Выдача сдачи запрещена
xA6	Есть 3 или более непереданных отчета
xBB	Лента не пуста
xB6	Ресторанный режим не активен
xB7	Ресторанный счет не открыт
xB8	Переполнение количества заказов
xB9	Ресторанный чек открыт
xBA	Неверный номер счета
xBС	Режим тренировки
xBD	Текущая дата неверна
xBE	Запрещено изменение времени
xBF	Истек сервисный таймер
xC0	Ошибка работы с терминалом НСМЕП
xC1	Неверный номер налога
xC2	Неверный параметр у процедуры
xC3	Режим фискального принтера не активен

xC4	Изменялось название товара или его налог
xC5	СКНО занято
xC6	ошибка обмена с СКНО (нет связи)
xC7	смена не открыта
xC8	СКНО переполнено
xC9	Неверный статус СКНО
xCA	Ошибка идентификации СКНО
xCB	запрещена операция продажи
xCC	Начата операция возврата
xCE	неверный тип кода товара
xCF	Не выведен отчет Z1
xD0	Не сделана инкассация денег
xD1	Сейф не закрыт
xD2	Печать ленты прервана
xD3	Достигнут конец текущей смены, или изменилась дата
xD4	Не указано значение процентной скидки по умолчанию
xD5	Не указано значение скидки по умолчанию
xD6	Дневной отчет не выведен
xD7	Дневной отчет уже выведен (и пуст)
xD8	Нельзя отменить товар на который сделана скидка без ее предварительной отмены
xD9	Товар не продавался в этом чеке
xDA	Нечего отменять
xDB	Отрицательная сумма продажи товара
xDC	Неверный процент
xDD	Нет ни одной продажи
xDE	Скидки запрещены
xDF	Неверная сумма платежа
xE0	Тип оплаты не предполагает введения кода клиента
xE1	Неверная сумма платежа
xE2	Идет оплата чека
xE3	Товар закончился
xE4	Номер группы не может меняться
xE5	Неверная группа
xE6	Номер отдела не может меняться
xE7	Неверный отдел
xE8	Нулевое произведение количества на цену
xE9	Переполнение внутренних сумм
xEA	Дробное количество запрещено
xEB	Неверное количество
xEC	Цена не может быть изменена
xED	Неверная цена
xEE	Товар не существует
xEF	Начат чек внесения-изъятия денег
xF0	Чек содержит продажи
xF1	Не существующий или запрещенный тип оплаты
xF2	Поле в строке переполнено
xF3	Отрицательная сумма по дневному отчету
xF4	Отрицательная сумма по чеку
xF5	Чек переполнен
xF6	Дневной отчет переполнен

xF7	Чек для копии не найден
xF8	Оплата чека не завершена
xF9	Кассир не зарегистрирован
xFA	У кассира нет прав на эту операцию
xFB	Нефискальный чек не открыт
xFC	Чек не открыт
xFD	Нефискальный чек уже открыт
xFE	Чек уже открыт
xFF	Переполнение ленты