

Протокол обмена фискальных аппаратов серии MG

Редакция 09.2021. Все изменения и дополнения относительно 2020 выделены **КРАСНЫМ**

Общие положения.

Регистраторы расчетных операций серии MG обладают возможностью работы в TCP/IP сетях и имеют встроенный веб сервер (порт 80 аппарата). Протокол обмена построен на [протоколе HTTP](#) и представляет собой набор запросов и ответов в [формате JSON](#). Например, посыл 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":"CHIP 702", "name":"CR00001234", "serial":"CR00001234",
"time":1412063757, chkId":0, "JrnTime":0, "currZ":11, "IsWrk":0, "FskMode":1,
"CurrDI":64, "err":[]}
```

Протокол позволяет:

- Получать и синхронизировать информацию по чекам (создавать для ФР),
- Регистрировать оператора и вести список операторов аппарата,
- Работать со встроенным модемом ДПА,
- 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 (однорочные 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&param2&...&paramN
```

где:

name - 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

Пример вызова процедуры (звуковой сигнал):

```
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":6,"Flg1":8,"Flg2":0,"Feed":3,"Flg3":0}
```

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":6,"Flg1":8,"Flg2":0,"Feed":3,"Flg3":0}
```

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="a5cba8b926bdc203f1b58df8a9bbdc", 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="948dd8135bdfdfdca86f72e639daf2b4", 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,"Qty":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,"Qty":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 (зависит от модели). 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	регехр 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

Таблица товаров.

В таблице товаров сохраняются данные обо всех товарах, которые может продать кассовый аппарат. В описание товара входят следующие поля:

Поле	Описание														
Code	Код товара (Для регистраторов - до 15 цифр, максимальное значение кода - 140'737'488'355'327; Для кассовых аппаратов – максимальное значение задается командой 919, просмотр – команда 916)														
Name	Название товара. Может быть разной длины для разных моделей аппаратов (до 87 либо до 60 знаков). При программировании товара, если передается название длиннее, чем в кассе – в кассу записывается усеченное название. Ошибка в таком случае не возвращается.														
Price	Цена товара. Диапазон 0.00.-.9999999.99														
Dep	Отдел. Число от 0 до 15														
Grp	Группа. Число от 0 до 8														
Tax	Налог. 1 – 4. Соответствует номеру налоговой ставки(Тax1-Тax4) из таблицы FTax. Нулевое значение ставок соответствует продаже этого товара без налогов.														
Qty	Количество товара. 0.000..999999.999.														
Flg	Битовая маска флагов товара <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th>Бит</th> <th>Значение при установленном бите</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Цена товара может быть изменена в момент продажи.</td> </tr> <tr> <td>1</td> <td>При продаже обязательно требуется ввод количества</td> </tr> <tr> <td>2</td> <td>Запрет продажи товара, если его общее проданное количество за текущую смену превышает значение, указанное в поле Qty</td> </tr> <tr> <td>4</td> <td>Автоматически закрывать чек после продажи этого товара если этот товар один в чеке</td> </tr> <tr> <td>5</td> <td>Запрет продажи этого товара</td> </tr> <tr> <td>6</td> <td>Запрет возврата этого товара</td> </tr> </tbody> </table>	Бит	Значение при установленном бите	0	Цена товара может быть изменена в момент продажи.	1	При продаже обязательно требуется ввод количества	2	Запрет продажи товара, если его общее проданное количество за текущую смену превышает значение, указанное в поле Qty	4	Автоматически закрывать чек после продажи этого товара если этот товар один в чеке	5	Запрет продажи этого товара	6	Запрет возврата этого товара
Бит	Значение при установленном бите														
0	Цена товара может быть изменена в момент продажи.														
1	При продаже обязательно требуется ввод количества														
2	Запрет продажи товара, если его общее проданное количество за текущую смену превышает значение, указанное в поле Qty														
4	Автоматически закрывать чек после продажи этого товара если этот товар один в чеке														
5	Запрет продажи этого товара														
6	Запрет возврата этого товара														

	7	Необходимость передачи номера акцизной марки
	8	Необходимость печати в чеке цифрового значения штрихкода товара (для регистраторов – формируется по значению код товара)
UktZed	Необязательное поле. Код УКТВЭД до 10 знаков. Нулевое значение соответствует отсутствию кода.	
BarCode	Только для кассовых аппаратов. Значение штрихкода товара (Минимальное значение задается командой 919, просмотр – команда 916; Максимальное значение – до 14 цифр)	

Таблица товаров доступна по адресу **/cgi/tbl/PLU**. Это единственная в кассовом аппарате таблица, в которой пользователь может удалять и добавлять строки.

Печать чеков (весь раздел чеков только для фискальных регистраторов).

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:

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.
5. После успешного окончания чека возвращается JSON содержимое чека.

Текстовый формат чека. – более не поддерживается

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
P	нефискальный чек
L	копия последнего чека
CP	чек выплаты средств держателю электронных платежных средств

Содержимое поля представляет собой массив строк чека. Для типа L значения строк не анализируются. Т.е. копия чека выдается последовательностью **{“L”:[]}**. 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
D	Строка скидки. Необязательное поле.	F,R
P	Строка оплаты. Необязательное поле (без этого поля оплата проводится по 1 оплате – наличными на всю сумму в чеке).	F,R,CP
IO	Строка служебного внесения/изъятия средств	IO
C	Строка текстового комментария. Необязательное поле.	F,R,IO,P,CP
N	Строка нефискального комментария. Необязательное поле.	F,R,IO,P,CP
BC	Штрих-код. Необязательное поле.	F,R,IO,P,CP

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 15 цифр, максимальное значение кода - 140'737'488'355'327
tax	Необязательное поле. Номер налоговой ставки из таблицы Tax . 0 – без налога. При отсутствии поля продажа происходит с первой налоговой ставкой из таблицы налогов.
uktzed	Необязательное поле. Код классификации товаров внешнеэкономической деятельности, согласно законов Украины
excise	Необязательное поле. Массив с номерами акцизных марок (количество в массиве должно соответствовать значению qty, иначе будет сообщение об ошибке, допускается значение меньше 1 в количестве при одном номере марки).
flg	Необязательное поле. 1 - необходима печать кода товара. 0, либо отсутствие – не печатать

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	Если присутствует. 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 (№1).
rrn	Номер транзакции (RRN) при осуществлении оплаты через банковский терминал. (Согласно требованиям НацБанка к системе передачи данных в ДПА). 1-12 символов Если отсутствует, все поля связанные с банковским терминалом игнорируются.
card	Номер банковской карты при оплате через терминал 1-18 символов

bank	Название банка при оплате через терминал. 1-18 символов
term	Номер платежного терминала. 1-18 символов
authcode	Код авторизации платежа. 1-6 символов
fee	Комиссия банка, при наличии.
paysyst	Необязательный параметр. Название платежной системы. 1-18 символов. Если отсутствует, то название вычисляется из первой цифры номера карты
needsign	Необязательный параметр. Флаг необходимости подписи держателя карты на чеке. Значения: 0, 1. Если отсутствует, принимается значение 0 - подпись не требуется

Строка внесения-изъятия денег.

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

Поле	Описание
type	"Ю" - строка описывает служебное движение денег
sum	Сумма внесения или изъятия. Сумма внесения должна быть положительна. Изъятия - отрицательна.
no	Необязательный параметр. Номер строки в таблице Pay , который соответствует типу выбранной оплаты. При отсутствии производится внесение или изъятие наличных (№1).

Строка текстового комментария.

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

Поле	Описание
Сm	Строка текстового комментария.

Строка нефискального комментария.

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

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

Поле	Описание
Сm	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 1-100 символа
type	Необязательное поле. Тип штрих-кода, если не указан, то по умолчанию №1: 1 – EAN-13 (максимально 12 символов, символ 13 генерируется автоматически) 2 – CODE-128 3 – CODE-39 4 – QR Code (до 100 символов)
width	Необязательное поле. Ширина символа в пикселях 2(по умолчанию),3,4
height	Необязательное поле. Высота штрих-кода в пикселях 1-150. По умолчанию 60
feed	Необязательное поле. Отступ до и после штрих-кода в пикселях 0-30. По умолч. 20.

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

Примеры чеков в формате JSON.

```
{ "IO": { "IO": { "sum": 500.00 } } }
```

```
{ "IO": [ { "C": { "cm": "Касир: Світлана" }, { "IO": { "sum": -20 } }, { "IO": { "sum": 140, "no": 2 } } ] }
```

```
{ "F": [ { "C": { "cm": "Касир: Світлана" }, { "S": { "code": 1, "price": 5, "name": "Цукерка" }, { "S": { "code": 2, "price": 15, "name": "Печиво", "qty": 0.5 } }, { "D": { "prc": 5, "all": 1 } }, { "P": {} } ] }
```

```
{ "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" } } ] }
```

```
{ "R": [ { "C": { "cm": "Касир: Світлана" }, { "S": { "code": 1, "price": 5, "name": "Цукерка" }, { "P": {} } ] }
```

```
{ "F": [ { "C": { "cm": "Касир: Світлана" }, { "S": { "code": "1", "price": "5", "name": "Цукерка" }, { "S": { "code": "2", "price": "6", "name": "Печиво" }, { "BC": { "code": "4823003211527", "type": 1, "width": 2, "height": 80, "feed": 20 }, { "BC": { "code": "653B9E8AB7C09EB9", "type": 2, "width": 1, "height": 80, "feed": 20 }, { "BC": { "code": "*123456789*", "type": 3, "width": 1, "height": 80, "feed": 20 }, { "BC": { "code": "Hello World!!!", "type": 4, "width": 4, "height": 0, "feed": 20 } ] }
```

```
{ "F": [ { "C": { "cm": "Касир: Адміністратор" }, { "S": { "code": 5, "name": "Препарат медичний №5", "price": 270.79, "qty": 1, "tax": 0 }, { "S": { "code": 7, "name": "Препарат медичний №7", "price": 41.82, "qty": 1, "tax": 0 }, { "D": { "sum": -41.28 }, { "P": { "sum": 41.28, "no": "5" }, { "P": { "sum": 300 } } ] }
```

```
{ "F": [
  { "S": { "code": 1, "price": 100.00, "qty": 2.000, "name": "Цукерка", "tax": 1, "excise": { "stamp": "AB ВП 26 ADDB 778549"}, {"stamp": "AB ВП 26 ADDB 778550"} } },
  { "S": { "code": 200, "price": 100.00, "name": "Товар 200", "tax": 1, "flg": 1, "excise": { "stamp": "AB ВП 26 ADDB 778551"} } },
  { "P": { "no": 1, "sum": 300.00 } }
] }

{ "F": [
  { "S": { "code": 6, "uktzed": 1234567890, "price": 100.00, "qty": 2.000, "name": "Товар 6", "tax": 1, "excise": { "stamp": "AB ВП 26 ADDB 778549"}, {"stamp": "AB ВП 26 ADDB 778550"} } },
  { "S": { "code": 200, "price": 100.00, "name": "Товар 200", "tax": 1, "excise": { "stamp": "AB ВП 26 ADDB 778551"} } },
  { "P": { "no": 1, "sum": 300.00 } }
] }

{ "R": [
  { "S": { "code": 1, "price": 100.00, "qty": 2.000, "name": "Цукерка", "tax": 1, "excise": { "stamp": "AB ВП 26 ADDB 778549"}, {"stamp": "AB ВП 26 ADDB 778550"} } },
  { "S": { "code": 200, "price": 100.00, "name": "Товар 200", "tax": 1, "excise": { "stamp": "AB ВП 26 ADDB 778551"} } },
  { "P": { "no": 1, "sum": 300.00 } }
] }

{ "R": [
  { "S": { "code": 6, "uktzed": 1234567890, "price": 100.00, "qty": 2.000, "name": "Товар 6", "tax": 1, "excise": { "stamp": "AB ВП 26 ADDB 778549"}, {"stamp": "AB ВП 26 ADDB 778550"} } },
  { "S": { "code": 200, "price": 100.00, "name": "Товар 200", "tax": 1, "excise": { "stamp": "AB ВП 26 ADDB 778551"} } },
  { "P": { "no": 1, "sum": 300.00 } }
] }

{ "F": [
  { "S": { "code": 1, "price": 100.00, "qty": 2.000, "name": "Цукерка", "tax": 1, "excise": { "stamp": "AB ВП 26 ADDB 778549"}, {"stamp": "AB ВП 26 ADDB 778550"} } },
  { "S": { "code": 200000, "price": 100.00, "name": "Товар 200", "tax": 1, "flg": 1, "excise": { "stamp": "AB ВП 26 ADDB 778551"} } },
  { "P": { "no": 4, "sum": 300.00, "fee": 2.00, "rrn": "abcd12345678", "bank": "Рога та копита", "card": "5166755628000982", "term": "S1KIONLY", "authcode": "39A546", "needsign": 1, "paysyst": "my own pay system"} }
] }

{ "CP": [
  { "P": { "no": 4, "sum": 100.00, "rrn": "12345678", "bank": "Рога та копита", "card": "5166*****0982", "term": "S1KIONLY", "authcode": "390546"} }
] }

{ "CP": [
  { "P": { "no": 4, "sum": 100.00, "fee": 2.00, "rrn": "12345678", "bank": "Рога та копита", "card": "*****0982", "term": "S1KIONLY", "authcode": "390546", "needsign": 1, "paysyst": "my own pay system"} }
] }
```

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 Касира
Z1	Номер дневного отчета
F	Строки фискального чека
R	Строки чека возврата
IO	Строки чека служебного движения средств
CP	Строки чека выдачи средств держателю карты
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

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.
DI	Номер данного документа в системе онлайн отчетности ДПА.

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
C	Строка описывает текстовый комментарий

Содержимое каждого из полей это объект, детализирующий операцию.

Содержание полей соответствует описанию из раздела “Формат чека Json” выше

Определение факта изменения чековой ленты.

Изменение чековой ленты можно обнаружить периодическим чтением состояния аппарата **/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.
CurrDI	DI последнего документа в системе онлайн отчетности ДПА

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.

Описание данных состояния.

Ресурс `/cgi/state` позволяет следить за состоянием аппарата. Может содержать следующие поля:

Поле	Описание
model	Название модели аппарата
name	Имя устройства в сети
serial	Заводской номер аппарата
FN	Фискальный номер аппарата
time	Текущая дата и время
chkId	Идентификатор последнего объекта в электронной ленте
JrnTime	Время начала чековой ленты
currZ	Номер последнего Z-отчета
IsWrk	Состояние ЭКЛ. Обнулена(да/нет, 0/1)
BegSmen	Состояние открытой рабочей смены (да/нет, 1/0)
Fiscalization	Состояние фискальной памяти. Фискализирована (да/нет, 1/0)
FskMode	Состояние готовности для печати документов (да/нет, 1/0)
CurrDI	DI последнего документа в системе онлайн отчетности ДПС
ID_SAM	Идентификатор SAM модуля
ID_DEV	Идентификатор аппарата
OstWrkTime	Счетчик времени в сек до блокировки, если это состоится. (Считает от 72ч до 0сек после последнего обмена информацией с ДПС)
NPrLin	Общее значение напечатанных строк
err	Описание ошибки (см. Приложение 1)

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 аппарат проводит следующие действия:

1. Печатает текст отчета.
2. Переносит итоги дня в фискальную память.
3. Иницирует соединение по отправке данных в ДПА.
4. После успешной передачи документов переносит её в хранилище документов. Текущая лента очищается.
5. При возникновении ошибки в обмене с сервером ДПА лента не очищается. Продажи для следующего отчета записываются в ленту далее. До начала продаж для следующего отчета обмен с сервером ДПА и очистку ленты можно иницировать процедурой `/cgi/proc/printreport?1`.

Получение данных из контрольной ленты.

Ресурс `/cgi/ejournal?Z=Rep` позволяет вычитать данные с контрольной ленты, где **Rep** – номер Z-отчета

Работа с индикатором.

Работа с индикатором клиента производится через ресурс **/cgi/scr**.

Установка содержимого индикатора производится методом POST.

При Content-Type=text/plain содержимое считается многострочным текстом, каждая строка которого соответствует одной строке индикатора.

При Content-Type=application/json содержимое считается массивом JSON. Текстовый элемент массива устанавливает содержимое строки, элемент массива 0 – оставляет имеющуюся строку на индикаторе без изменения. Например: ["Строка 1", "", 0, "Строка 4"]

Поле	Описание
id	номер строки
str	содержимое строки
align	Выравнивание: "l" - влево, "r" - вправо, "c" - (центрирование)

Пример:

```
[  
{"id":1, "str":"Строка 1"},  
{"id":3, "align":"r", "str":"Строка 3"}  
]
```

При Content-Type=application/jpeg содержимое считается изображением формата JPEG. Это изображение будет выведено на индикатор.

Информацию об индикаторе можно получить из ресурса **/cgi/scr** методом GET. Возвращается ширина (w) и высота (h) индикатора в символах (chars) {"chars":{"w":16,"h":2}}

Получить количество строк на используемом индикаторе можно методом GET из **/cgi/scr/lines**.

Получить количество пикселей на экране можно методом GET из **/cgi/scr/pix**. В ответ придёт массив JSON в первом элементе, которого будет количество точек по горизонтали, а во втором – по вертикали. Например: [800,600].

Работа с денежным ящиком.

Ресурс **/cgi/proc/openbox** позволяет принудительно открыть денежный ящик

Работа с модемом по передаче данных в ДПА.

Настройки модема.

Настройка модема производится через таблицу `/cgi/tbl/NSMEP` с полями:

Поле	Описание
Addr	IP адрес или имя хоста экваера
Port	Порт экваера
PackSize	Размер пакета данных при обмене с экваером

С модемом можно проводить следующие действия методом GET:

Адрес	Действие
<code>/cgi/do_conn</code>	Старт обмена модема с экваером
<code>/cgi/sam_info?p=0</code>	Выключение модуля SAM. Снятие питания с модуля, например, для его замены.
<code>/cgi/sam_info?p=1</code>	Включение модуля SAM.
<code>/cgi/sam_info?p=2</code>	Персонализация модуля SAM.
<code>/cgi/pers</code>	Персонализация модема.

Получение информации о номере документа в ДПА.

Адрес	Действие
<code>/cgi/di_chk?c=chkNo&z=zNo</code>	Позволяет получить информацию о номере DI чека с номером chkNo выбитом в дневном отчете с номером zNo
<code>/cgi/di_chk?z=zNo</code>	Позволяет получить информацию о номере DI дневного отчета с номером zNo

Если указанный документ существует, его номер возвращается как объект JSON `{"doc_di" : no}`. Иначе возвращается номер ошибки `{"msg" : no}`. Список ошибок приводится ниже.

Получение текста и проверка правильности документа.

Адрес	Действие
<code>/cgi/ditxt?p=DI</code>	Позволяет получить текст документа с номером DI.
<code>/cgi/verify?di=DI</code>	Позволяет верифицировать документ с номером DI. Возврат приходит в виде <code>{"msg" : no}</code> . Список возможных значений no приводится ниже.

Список кодов возврата msg.

Код	Описание
0	Документ верен
1	Документ неверен
2	Документ не найден
3	Чек не найден
4	Отчет не найден

Статус модема

Статус модема можно получить HTTP методом GET по адресу `/cgi/status`. Статус содержит следующие поля:

Поле	Описание												
dev_state	Битовая маска статуса аппарата: <table border="1"><thead><tr><th>Бит</th><th>Значение</th></tr></thead><tbody><tr><td>1</td><td>0 - модуль SAM не обнаружен 1 - модуль SAM обнаружен</td></tr><tr><td>2</td><td>0 - модуль SAM сопряжен с устройством 1 - модуль SAM не сопряжен с устройством</td></tr><tr><td>3</td><td>0 - персонализация отсутствует 1 - персонализация присутствует</td></tr><tr><td>4</td><td>0 - система персонализирована 1 - ошибка персонализации</td></tr><tr><td>5</td><td>0 - хранилище документов разрушено</td></tr></tbody></table>	Бит	Значение	1	0 - модуль SAM не обнаружен 1 - модуль SAM обнаружен	2	0 - модуль SAM сопряжен с устройством 1 - модуль SAM не сопряжен с устройством	3	0 - персонализация отсутствует 1 - персонализация присутствует	4	0 - система персонализирована 1 - ошибка персонализации	5	0 - хранилище документов разрушено
Бит	Значение												
1	0 - модуль SAM не обнаружен 1 - модуль SAM обнаружен												
2	0 - модуль SAM сопряжен с устройством 1 - модуль SAM не сопряжен с устройством												
3	0 - персонализация отсутствует 1 - персонализация присутствует												
4	0 - система персонализирована 1 - ошибка персонализации												
5	0 - хранилище документов разрушено												

	1 - хранилище документов исправно
6	0 - нет сетевого соединения 1 - сетевое соединение установлено
7	0 - модем функционирует нормально 1 - ошибка модема
tm	Текущая дата-время
tmo	Интервал между сеансами связи с экваером в секундах
ct	Время до следующей связи с экваером в секундах
bt	Время до блокировки аппарата в секундах
ndoc	Количество переданных документов
pers_sam_id	Номер модуля ID_SAM, с которым персонализирован модем
Pers_time	Дата персонализации
card_no	Номер карточки модуля SAM
sam_id	Номер модуля ID_SAM
sam_dev_id	номер устройства ID_DEV с которым сопряжен модуль SAM
eq_id	Номер экваера для которого инициализирован модуль SAM

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_id	НСМЕП номер устройства
dev_dat	Дата прошивки устройства
dev_fn	Фискальный номер аппарата
dev_nn	Налоговый номер аппарата
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": "CHIP 702", "name": "CR00001234", "serial": "CR00001234",
"time": 1412063757, "chkId": 0, "JrnTime": 0, "currZ": 11, "IsWrk": 0, "FskMode": 1,
"CurrDI": 64, "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/whiteIP**. 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 **whiteIP** 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 налогового и фискального номеров как 4-й и 5-й строки заголовка чека через таблицу Hdr .
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
Extra	процент сбора

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
NumPnt	Позиция десятичной точки в денежных суммах
NumChk	Номер первого чека в открытой смене

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
Sbor1-Sbor4	Ставки сборов.
NDay	Номер дневного отчета, после которого записана запись.

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.

Печать и получение отчетов.

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

Значение	Описание
0	Дневной обнуляющий отчет
1	Отчет по обнулению электронной ленты
10	Дневной отчет без обнуления
20	Отчет по проданным товарам
21	Отчет по проданным товарам с обнулением этого 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)

Чеки, сохраненные в электронной ленте печатаются процедурой

`/cgi/proc/printmmcjrn?ZRepID&BegRcpt&EndRcpt`, где

ZRepID	Номер Z отчета
BegRcpt	Начальный номер чека
EndRcpt	Конечный номер чека

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.

Технологический сброс.

Технологический сброс производится после устранения фатальных ошибок в аппарате или смены прошивки аппарата на прошивку с измененной структурой памяти. Технологический сброс переформатирует память аппарата и устанавливает время аппарата, начальный номер чека и начальный номер документа в системе отчетности ДПА. Он производится вызовом процедуры `/cgi/proc/resetram?StartRcpt&Time&StartDI`, где `StartRcpt` - начальный номер чека, `Time` - дата и время для установки в аппарате, `StartDI` - начальный номер документа в системе отчетности ДПА. Каждый сброс записывается в таблицу фискальной памяти FSbr с полями

Поле	Описание
id	номер записи о сбросе
valid	Флаг действительной записи 1-действительна, 0-нет.
datetime	Дата и время записи в Unix-time
NDay	Номер дневного отчета, после которого записана запись.

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 <table border="1" data-bbox="349 1417 1409 1606"> <thead> <tr> <th>Величина</th> <th>Описание</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>вид оплаты разрешен</td> </tr> <tr> <td>2</td> <td>по виду оплаты возможна сдача</td> </tr> <tr> <td>16</td> <td>вид оплаты не лежит в сейфе</td> </tr> <tr> <td>32</td> <td>вид оплаты может использоваться для оплаты возвратных чеков</td> </tr> </tbody> </table> Значение поля является суммой необходимых флагов	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
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										
Name	Название вида оплаты										

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	
	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.

Приоритет некоторых сообщений от подсистем аппарата.

Priority	сообщения от
	подсистемы синхронизации времени
	сетевой подсистемы
	GPRS модема
	загружаемого вебинтерфейса
	подсистемы печати
	подсистемы аудита (регистрация пользователей (успешная, ошибочная), вход/выход в режим программирования)

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.

Печать графического файла.

Ресурс `/cgi/ticket.bmp` позволяет печатать HTTP методом POST монохромный рисунок в формате BMP на служебном документе. Размеры печатаемого рисунка ограничивается следующими параметрами :

Ширина 408 точек

Высота 1000 точек

Настройка принтера.

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

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

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 **Fig3**.

Флаг	Описание
64	не печатать имя Касира
256	не печатать логотип
524288	печатать ленту перед дневным отчетом

Значение поля является суммой используемых флагов.

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

Ресурс `/cgi/proc/feedpaper?1` позволяет выполнять протяжку бумаги на требуемое количество строк.

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
-----	----------

upnp: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:discover\\"\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;
        }
    }
}

```

Приложение 1. Список ошибок кассы.

Код	Описание
х01	Цена не указана
х02	Количество не указано
х03	Отдел не указан
х04	Группа не указана
х25	Нет бумаги
х31	Пользователь уже зарегистрирован
х32	Неверный пароль
х33	Неверный номер таблицы
х34	Доступ к таблице запрещен
х35	Умолчание не найдено
х36	Неверный индекс
х37	Неверное поле
х38	Таблица переполнена
х39	Неверная длина двоичных данных
х3А	Попытка модификации поля только для чтения
х3В	Неверное значение поля
х3С	Товар уже существует
х3D	По товару были продажи
х3E	Запрос запрещен
х3F	Неверная закладка
х40	Ключ не найден
х41	Процедура уже выполняется
х42	Количество товара отрицательно
х43	Включено перенаправление с карты памяти
х44	501 отчет не пуст
х87	Ошибка фискальной памяти
х88	Ошибка карты памяти
х89	Переполнение карты памяти
х8А	Нет бумаги для контрольной ленты
х8В	Нет бумаги
х8С	Переполнение фискальной памяти
х8D	Выдача сдачи запрещена
хА3	Операция прекращена устройством
хА5	Дневной отчет не найден
хА7	ММС запрещено
хА8	Неверен номер фискальной памяти
хА9	Фискальная память не пуста
хВВ	Лента не пуста
хВС	Режим тренировки
хVD	Текущая дата неверна
хВЕ	Запрещено изменение времени
хBF	Истек сервисный таймер
хС0	Ошибка работы с терминалом НСМЕП
хС1	Неверный номер налога
хС2	Неверный параметр у процедуры
хС3	Режим фискального принтера не активен
хС4	Изменялось название товара или его налог
хС5	Необходима персонализация
хС6	Отсутствует обмен с сервером НСМЭП на протяжении 72 часов

xС7	Запрещено обнуление данных ЭКЛ
xС8	Запрещена работа с GPRS модемом
xCC	Начата операция возврата
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	Касир не зарегистрирован
xFa	У Касира 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

