

Протокол обмена для кассовых аппаратов "Хелп Микро".

Общие положения.

Кассовые аппараты "Хелп Микро" 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, 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":"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,
- 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 (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&param2&...&paramN
```

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]][THH[:MM[:SS]]]
Текст	Text	В формате кодирования URL http://en.wikipedia.org/wiki/Percent-encoding Пример: ABC%20%E0%B1

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":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. 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": {
      ...
      "хЗА": "Попытка модификации поля только для чтения",
      "хЗВ": "Неверное значение поля",
      "хЗС": "Товар уже существует",
      "хЗD": "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 <code>/cgi/tbl/имя ресурса</code>
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

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 (до 14 цифр)														
Name	Название товара. Может быть разной длины для разных моделей аппаратов. 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(Tax1-Tax4) 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" data-bbox="337 1474 1317 1814"> <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														

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

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
CP	Строка описывает выплату наличных собственнику электронного платежного средства (укр. ЕПЗ – електронний платіжний засіб).
IO	строки чека служебного движения средств
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

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
uktzed	Код УКТ ЗЕД, который должен быть одинаковым (или пустым) для товара на протяжении смены
excise	Массив с номерами акцизных марок в случае продажи соответствующих товаров. Если поле задано, количество элементов массива должно совпадать со значением поля qty.

Пример:

```
{ "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 } }
]
```

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.
dn	номер подтверждения льготной скидки. (Согласно требованиям НацБанка к системе передачи данных в ДПА)
tax	Номер налога, на который начислена скидка. Если в чеке проданы товары с разными налогами, то одна строка скидки на промежуточный итог создаст несколько строк скидки в чеке по числу налогов, использующихся в чеке.

Строка оплаты.

Поле	Описание
sum	Сумма оплаты.
no	Номер строки в таблице Pay , который соответствует типу выбранной оплаты.
rrn	Номер транзакции (RRN) при осуществлении оплаты через банковский терминал. (Согласно требованиям НацБанка к системе передачи данных в ДПА)
card	Номер банковской карты при оплате через терминал

Строка оплаты (расчет картой).

Поле	Описание
sum	Сумма оплаты.
no	Номер строки в таблице Pay , который соответствует типу БЕЗНАЛИЧНОЙ оплаты.
rrn	RRN (Reference Retrieval Number) платежа
card	Номер карты клиента
fee	Комиссия банка (в случае наличия)
authcode	Код авторизации платежа
bank	Название банка
term	Номер платежного терминала

Пример:

```
{ "F": [  
  {"S": {"code":1, "price":100.00, "qty":2.000, "name":"Товар 1", "tax":1}},  
  {"S": {"code":200, "price":100.00, "name":"Товар 200", "tax":1}},  
  {"P": {"no":4, "sum":300.00, "fee":2.00, "rrn":"12345678", "bank":"Рога  
та копита", "card":"5166755628000982",  
"term":"S1KIONLY", "authcode":"390546"}}  
]}
```

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 текстового комментария.

Поле	Описание
cm	Строка комментария

Строка выплаты наличных собственнику электронного платежного средства.

Поле	Описание
sum	Сумма выплаченной наличности.
no	Номер строки в таблице Pay , который соответствует типу БЕЗНАЛИЧНОЙ оплаты.
rrn	RRN (Reference Retrieval Number) платежа
card	Номер карты клиента
fee	Комиссия банка (в случае наличия)
authcode	Код авторизации платежа
bank	Название банка
term	Номер платежного терминала

Пример:

```
{ "CP": [  
  { "P": { "no": 4, "sum": 100.00, "fee": 2.00, "rrn": "12345678", "bank": "Рога  
та копита", "card": "5166*****0982",  
"term": "S1KIONLY", "authcode": "390546" } }  
]}
```

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.

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`.

Идентификация устройства.

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

Поле	Описание
<code>dev_zn</code>	Заводской номер
<code>dev_ver</code>	Версия ПО
<code>dev_id</code>	НСМЕП номер устройства
<code>dev_dat</code>	Дата прошивки устройства
<code>dev_fn</code>	Фискальный номер аппарата
<code>dev_nn</code>	Налоговый номер аппарата
<code>prot</code>	Версия протокола аппарата. Поле отсутствует для аппаратов, в которых номер оплаты при печати чеков на единицу меньше, чем <code>id</code> 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.

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:

Адрес	Действие
/cgi/do_conn	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.

Все действия с модемом производятся HTTP методом GET.

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	<p>Битовая маска статуса аппарата:</p> <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> </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
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
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	
pers_sam_id	Номер модуля ID_SAM, с которым персонализирован модем	
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	

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

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

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 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" data-bbox="349 1543 1404 1816"> <thead> <tr> <th>Бит</th> <th>Значение</th> </tr> </thead> <tbody> <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> </tbody> </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
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;
        }
    }
}
}

```

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
	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 **Flg3**.

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

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**:

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.

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
x43	Включено перенаправление с карты памяти
x44	501 отчет не пуст
x87	Ошибка фискальной памяти
x88	Ошибка карты памяти
x89	Переполнение карты памяти
x8A	Нет бумаги для контрольной ленты
x8B	Нет бумаги
x8C	Переполнение фискальной памяти
x8D	Выдача сдачи запрещена
xA3	Операция прекращена устройством
xA5	Дневной отчет не найден
xA7	ММС запрещено
xA8	Неверен номер фискальной памяти
xA9	Фискальная память не пуста
xBB	Лента не пуста
xBC	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	Необходима персонализация

xС6	Отсутствует обмен с сервером НСМЭП на протяжении 72 часов
xС7	Запрещено обнуление данных ЭКЛ
xС8	Запрещена работа с GPRS модемом
xСС	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
xEС	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