

NV-100

Приложение к руководству по эксплуатации

**Руководство по автоматическому обновлению
программного обеспечения**

Full HD медиацентр

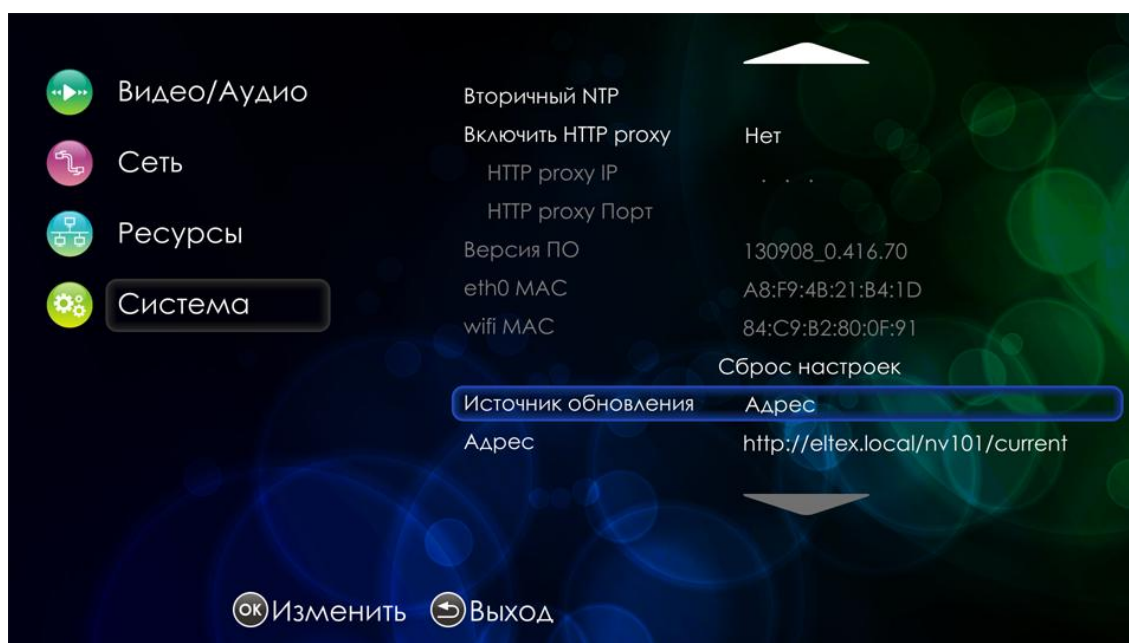
Автоматическая смена ПО возможна следующими способами:

1. загрузка прошивки с сервера провайдера с использованием HTTP;
2. получение прошивки, транслируемой по сети в определённой мультикастовой группе.

1. Настройка HTTP-сервера обновлений

При смене программного обеспечения по HTTP приставка использует для доступа к серверу обновлений прописанный по умолчанию адрес, например:

<http://eltex.local/nv102/current>.



Обращение по этому адресу на сети провайдера должно быть перенаправлено на сервер, на котором размещается адаптированная под данного провайдера прошивка. Помимо файла прошивки на сервере должны размещаться файлы ключей, подписи и файл конфигурации со специфичными для данного провайдера настройками и дополнительными (если это необходимо) программными компонентами. В общем случае наличие файла настроек не является обязательным и он может отсутствовать. Инструкция по созданию ключей и подписи прошивки приведена ниже.

При обновлении по HTTP провайдер может использовать либо свою прошивку, либо воспользоваться стандартной прошивкой. Стандартная прошивка может быть адаптирована под провайдера. Адаптация заключается в смене/добавлении/удалении файлов по отношению к стандартной прошивке. Операции по адаптации выполняются с использованием файла `config.tar.gz`, который должен располагаться в одной папке с прошивкой и файлами ключей и подписи.

В состав этого архива входят:

1. file_list.txt – текстовый файл, содержащий список файлов, которые необходимо положить в файловую систему уже прошитой приставки;
2. files_to_delete.txt – список файлов, подлежащих удалению из файловой системы после прошивки приставки;
3. Файлы, подлежащие записи в приставку, согласно списку в file_list.txt.

При включении по питанию приставка анализирует появление обновлений, прочитав по адресу <http://eltex.local/nv10x/current> файл *current* с версией прошивки. В случае если номер версии на сервере обновления окажется старше номера версии прошивки, приставка начнёт процесс обновления прошивки. Обновление возможно при совпадении электронных подписей прошивки и приставки.

В процессе обновления происходит сначала запись на внутреннюю NAND flash ядра Linux и файловой системы, а затем применение изменений, содержащихся в файле config.tar.gz.

Примеры:

Содержимое файла config.tar.gz, применяемого у одного из операторов:

```
file_list.txt
files_to_delete.txt
index.html
mtconf.xml
prov_settings.xml
```

Содержимое файла files_to_delete.txt:

```
/sdk/qt/STBGUI_PLUGIN/appstore/libappstore.so
/sdk/qt/STBGUI_PLUGIN/minute/libminute.so
```

После перепрошивки приставки файлы, указанные в files_to_delete.txt, будут удалены.

Содержимое файла file_list.txt:

```
/sdk/config/config_files/prov_settings.xml
/sdk/config/config_files/rootcert.pem
/sdk/config/config_files/mtconf.xml
/sdk/config/config_files/index.htm
/home/user/index.htm
```

После перепрошивки приставки файлы, перечисленные в file_list.txt, будут скопированы из архива по заданным путям.

Необходимо помнить, что автоматическое обновление применяется только к прошивке, расположенной в рабочем разделе. Прошивка расположенная на резервном разделе не изменяется.

Для предотвращения смены прошивки на невалидную или обновления на прошивку стороннего оператора используются секретные ключи и подписи прошивок.

1. Создаем секретный ключ `secret.key.pem`:

```
«openssl genrsa -out secret.key.pem -des3 -rand /var/log/messages 4096»
```

2. По секретному ключу создается открытый ключ `pub.key.pem`:

```
«openssl rsa -in secret.key.pem -out pub.key.pem -pubout»
```

3. По секретному ключу и файлу прошивки создаем цифровую подпись:

```
«openssl dgst -sha1 -sign secret.key.pem -out nv10ximg_date_ver.sha1  
nv10ximg_date_ver»
```

Пункты 1-2 делаются всего 1 раз.

Пункт 3 надо выполнять каждый раз для новой прошивки заново.

4. На сервере HTTP нужно организовать следующие файлы и директории:

- `current` – файл, содержащий строку с текущей версией прошивки;
- `nv10ximg_date_ver.xml` – файл, содержащий пары комбинаций "издатель - цифровая подпись";
- `keys.xml` – файл, содержащий пары комбинаций "издатель - открытый ключ";
- `nv10ximg_date_ver` – файл прошивки;
- `nv10ximg_date_ver.sha1` – цифровая подпись к данной прошивке;
- `keys/pub.key.pem` – открытый ключ.

Обновление приставки:

- приставка скачивает открытый ключ (`pub.key.pem`), цифровую подпись к прошивке (`nv10ximg_date_ver.sha1`), прошивку (`nv10ximg_date_ver`) и проверяет совпадение;
- если в составе прошивки отсутствует ключ, этого достаточно для прошивки новой версии на плату;
- если в составе прошивки есть ключ – проверяется совпадение этого внутреннего ключа с `pub.key.pem`. Обновление ПО происходит только в случае совпадения.

Примеры:

Пример файла current:

```
nv10ximg_110711_0.0.3
```

Пример файла nv10ximg_date_ver.xml:

```
<?xml version="1.0" encoding="UTF-8" ?>
<firmware>
  <version>0.0.3</version>
  <date>110711</date>
  <image>nv10ximg_110711_0.0.3</image>
  <critical>>false</critical>
<signatures>
  <signature>
    <publisher>Eltex</publisher>
    <filename>nv10ximg_110711_0.0.3.sha1</filename>
  </signature>
</signatures>
</firmware>
```

Пример файла keys.xml:

```
<?xml version="1.0" encoding="UTF-8" ?>
<keys>
  <key>
    <owner>Eltex</owner>
    <filename>keys/pub.key.pem</filename>
  </key>
</keys>
```

Примечания.

1. Поля publisher (в nv10ximg_date_ver.xml) и owner (в keys.xml) должны совпадать (пример приведен для издателя Eltex).
2. Все комбинации "ключ - цифровая подпись" в этих файлах должны относиться к одной и той же версии прошивки.
3. Поле «critical» указывает критическое обновление или нет (критическое обновление выполняется принудительно и не требует согласия пользователя).

При использовании режима смены ПО с использованием мультикаста, прошивка циклически транслируется провайдером в заранее предопределённых мультикастовых группах. Полная прошивка состоит из двух частей: ядра и файловой системы, которые транслируются в разных мультикастовых группах.

Для обеспечения возможности смены ПО с использованием мультикаста, необходимо произвести настройку DHCP-сервера, т.к. приставка получает нужный адрес мультикастовой группы и номер порта в параметрах, получаемых от DHCP-сервера через механизм «vendor specific».

Для смены ядра используется идентификатор "eltexNV10xkboot". Для смены файловой системы используется идентификатор "eltexNV10xfsboot".

2. Порядок настройки автономного multicast-сервера обновлений

Для запуска сервера обновлений понадобится компьютер x86 под управлением ОС Linux, файлы содержащие ядро и файловую систему для смены прошивки, конфигурационные файлы для DHCP и для программы «mcastbootd», осуществляющей трансляцию файлов прошивки.

1. Необходимо откорректировать файл dhcp.conf, подставив в него необходимые значения диапазона адресов, номеров портов и т.п.
2. Необходимо откорректировать файл mcastbootd.conf, содержащий имя транслируемого файла, IP-адрес и номер порта (должны совпадать со значениями, указанными в файле dhcp.conf), установить размер пакета и скорость передачи данных.
3. Запустить DHCP, используя команду «dhcpd -cf dhcp.conf».
4. Запустить трансляцию, используя команду «mcastbootd -D -C mcastbootd.conf».

После этого начнётся трансляция в сеть указанных в файле mcastbootd.conf прошивок по записанным в конфигурации адресам. При подключении приставки к сети, в случае установления режима обновления с использованием мультикаста, произойдёт запрос к DHCP-серверу сначала с параметром «eltexNV10xkboot», и, в случае получения от сервера адреса мультикастовой группы и номера порта, запуск процедуры обновления ядра. После обновления ядра процедура повторится для файловой системы.

