

## Tutorial zmiany oprogramowania w NBox – ADB-5800s serial BSKAxxxxxxx

Soft max 4.7ca 17 data aktualizacji 09.2009, jest to wersja FINALNA dla BSKA i bootowania z PENDRIVE, HDD lub NAND (pamięć wewnętrzna)

Polecam też zapoznanie się z Tutorialem v4, v5.2 link na końcu.

### **PRZECZYTAJ CAŁY PORADNIK ZANIM ZACZNIESZ COŚ ROBIĆ!!! CAŁY!!!**

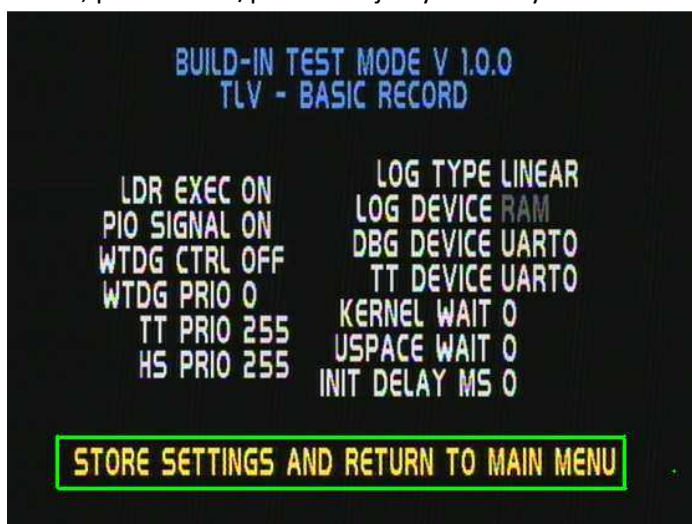
1. Najpierw trzeba odpalić w tunerze Menu Serwisowe. W tym celu:  
Przyciskamy przycisk **MENU** na front panelu i włączamy tuner do prądu – pojawi się napis **tEst** i diody zaczną mrugać po ok. 5-6sek. Puszczaemy na 1sek i znów trzymamy, jak pojawi się napis **dL** wciskamy po kolei: **< OK > OK OK <** pojawi nam się takie okno



2. Wpisujemy z pilota kod: **BACK ' 3 ' OK ' 7 ' EPG ' 4 ' 9 ' CH.DOWN**
3. Pojawi nam się Menu Serwisowe



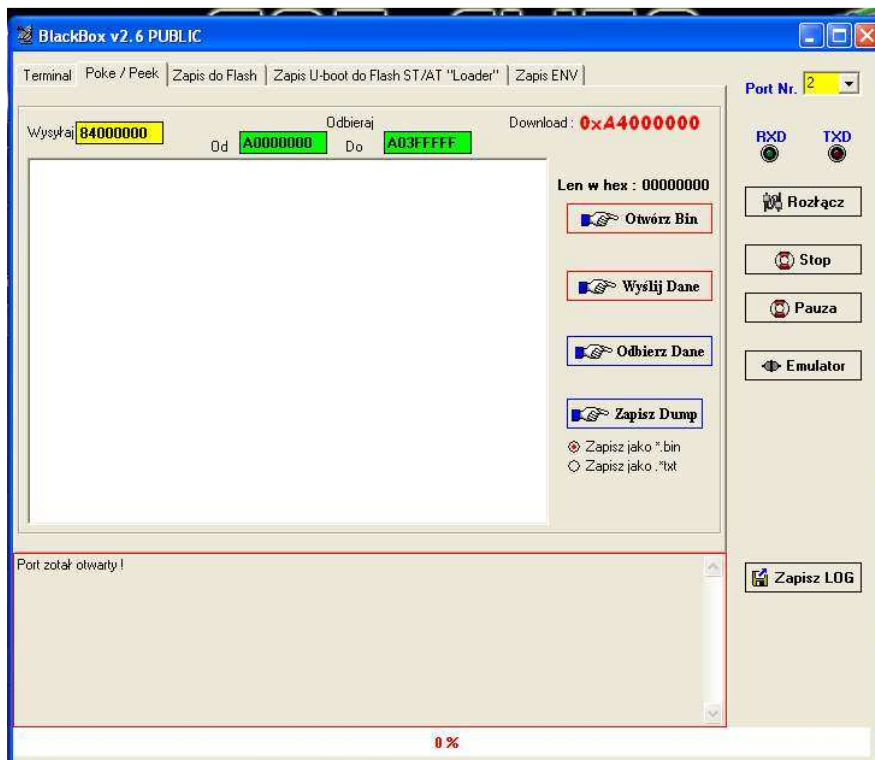
4. Wybieramy opcję **2. TLV – BASIC RECORD** i ustawiamy parametry jak na zdjęciu poniżej, poczym schodzimy na sam dół (tam gdzie jest **IGNORE CHANGES AND RETURN TO MAIN MENU**), potem lewo/prawo i dajemy szukamy **STORE** i zatwierdzamy OK.



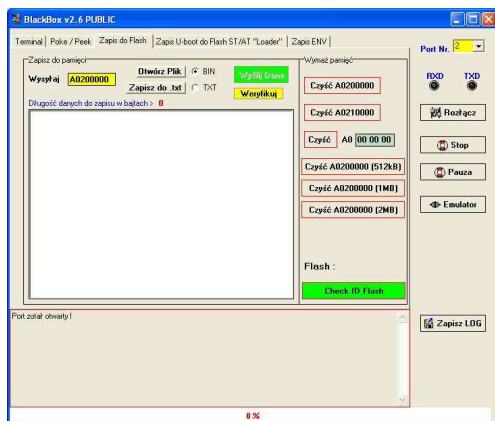
5. Jesteśmy znów w głównym menu na samym dole lewo/prawo i szukamy opcji **EXIT AND START LOADER**.



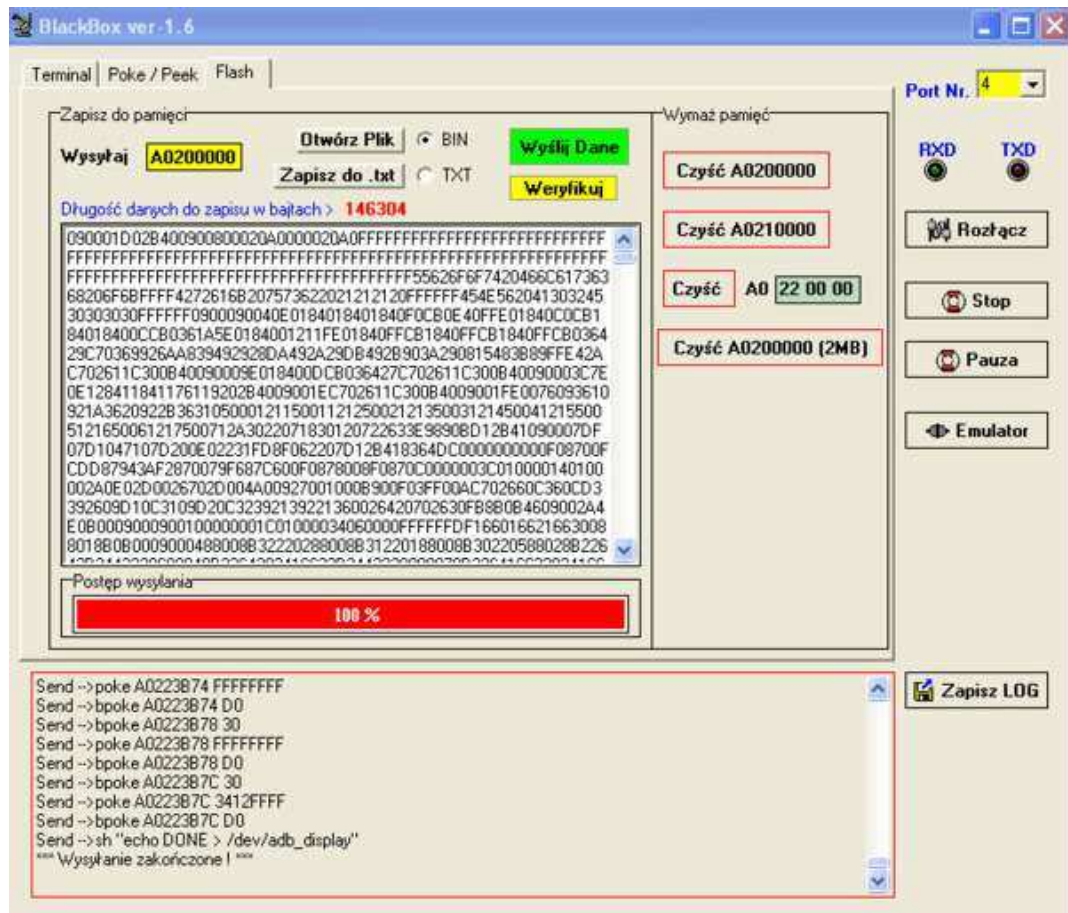
6. Nbox, normalnie się uruchomi i pojawi nam się ekran aktywacyjny bądź paski poziomego sygnału z satelity. Podłączamy kabelek USB<->TTL. Odpalamy program **BlackBox 2.6 public** wybieramy port, na którym mamy zainstalowany interfejs i klikamy Połącz.
7. Otwieramy zakładkę **Poke/Peek** i próbnie odczytujemy jakieś dane z dowolnego sektora, co by sprawdzić czy poprawnie działa komunikacja. Próbujemy kilka razy i porównujemy dane, jeśli za każdym razem mamy to samo to jest ok.



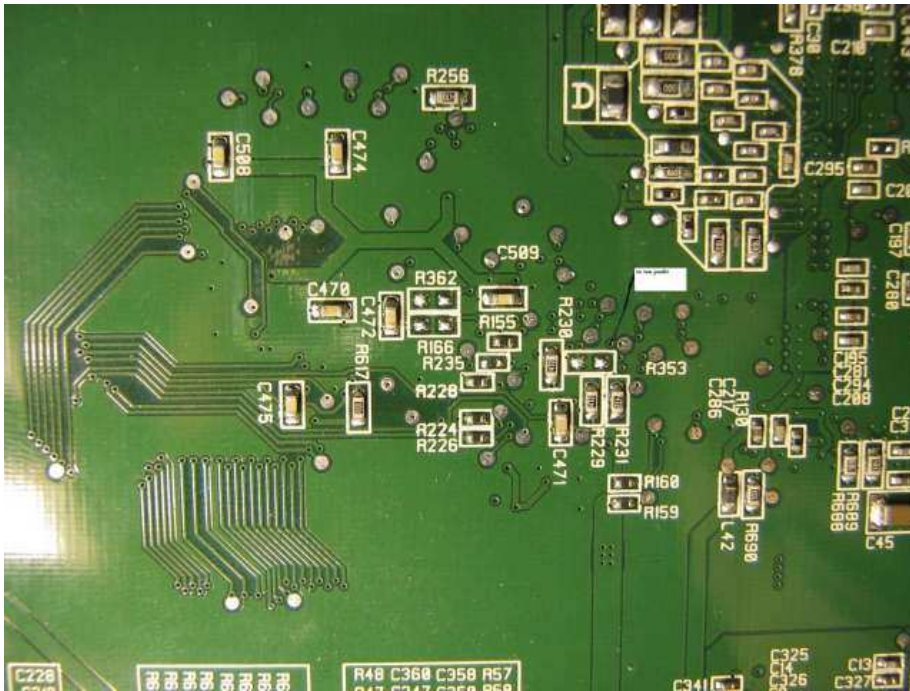
8. Otwieramy zakładkę **Zapis do Flash** i klikamy **Check Flash ID** program sam określi jaki mamy flash ST czy AT i wybierze dla nas formę zapisu.
9. W tej samej zakładce klikamy po kolei **Czyść A0200000**, **Czyść A0210000** i **Czyść A0220000** (ten musimy ręcznie dopisać) to trwa moment.
10. Teraz Otwórz Plik (BIN) i wybieramy uBoot od kolegi pewnego forum (wielkie dzięki mu za to) „*u-boot A020 Dla Rs232\_Zmiana\_01\_2011\_OK.bin*” i klikamy **Wyslij dane**, proces może potrwać od **30 do 90min!!!** Dlaczego nie wrzucamy od razu właściwego uBoot, bo w trakcie programowania czasem mogą pojawić się błędy. Właściwy uBoot dogramy później.



11. Po zakończeniu procesu programowania pojawi się komunikat w programie „**Wysyłanie zakończone**”, a na front panelu napis **dOnE**



12. Następnie odłączamy zasilanie rozkręcamy do końca tuner, obracamy płytę główną i lutujemy zworę jak na zdjęciu poniżej, z pkt oznaczonego do 3V3 złącza debug.



### 13. Składamy tuner

Odpalamy **HyperTerminal** -> Nowe Połączenie (nazwa dowolna np. nbox) -> Połącz używając COMx -> Liczba bitów na sekundę 115200; Bit dany 8; Parzystość brak; Bit stopu 1;

Sterowanie przepływem: brak

Podłączamy kabelek USB do tuner i włączamy go do prądu (wyświetlacz będzie martwy póki co).

Po tym powinien nam się zgłosić w terminalu nasz tuner.

*Board: STb7100-Reference (MB442) [29-bit mode]*

*U-Boot 1.3.1 (Aug 20 2010 - 05:28:08) - stm24\_0051*

*DRAM: 64 MiB*

*NOR: 4 MiB*

*\*\*\* Warning - bad CRC, using default environment*

*In: serial*

*Out: serial*

*Err: serial*

*MB442>*

Teraz wpisujemy po kolei:

**loady** <enter>

w tym momencie wybierasz transfer -> Wyślij plik -> protokół YModem i tam wskazujesz gdzie masz plik z uBoot w naszym przypadku jest to „**u-boot-m28w320fst\_053\_11052011.bin**” jeśli mamy pamięć ST – FST, FSB lub ATmel, lub „**u-boot-m28w320fsu\_053\_11052011.bin**” jeśli mamy pamięć ST– FSU, idzie dość szybko ok. 2min. Jak się wgrało to wpisujemy

**protect off A0200000 A024FFFF** <enter>

potem

**erase A0200000 A024FFFF** <enter>

następnie

**cp.b 84000000 A0200000 245D0** <enter>

#### 14. Restartujemy NBox'a (wyłączyć z prądu)

Na wyświetlaczu pojawi się napis **boot**. Teraz dogrywamy komendy ENV. Po ponownym restarcie tuner zgłasza się tak:

*Board: Nbox [29-bit mode] by FREEBOX*

*U-Boot 1.3.1 (May 11 2011 - 17:17:33) - stm23\_0053*

*DRAM: 128 MiB*

*NOR: 4 MiB*

*\*\*\* Warning - bad CRC, using default environment*

*In: serial*

*Out: serial*

*Err: serial*

*NBOX>*

Programujemy same komendy ENV dla obsługi tuner z PENDRIVE bez HDD:

**set bootargs 'console=ttyAS0,115200 root=/dev/sda1 rw mem=128m  
coprocessor\_mem=4m@0x10000000,4m@0x10400000 rootdelay=6  
init=/bin/devinit'** <enter>

**setenv bootcmd 'usb reset;ext2load usb 0:1 a5000000  
/boot/uImage;bootm a5000000'** <enter>

**saveenv** <enter>

15. Przygotowanie Enigmy2 na pendrive, opcje są dwie, albo korzystamy z Linuxa i formatujemy pendrive pod **ext2** i rozpakowujemy plik „**bska\_v2.tar.gz**” na przygotowany wcześniej pendrive, albo korzystamy z programu HD Clone 3.9 po Windowsem.

Instalujemy program HD Clone na dysku. Zakładamy katalog **C:\HDClone Images** i do niego wrzucamy to co powstanie z rozpakowania np. „**bska\_v2\_4GB.zip**” czyli katalog „**bska\_v2\_4GB.img**” a w nim 4 pliki i 2 ukryte. Na upload (na końcu linki) są image E2 i Neutrino dla konkretnych pojemności pendrive.

Odpalamy program, wybieramy źródło **\*.IMG** wybieramy miejsce docelowe **DRIVE** - nasz pendrive o odpowiedniej pojemności . Klikamy next chwilę czekamy i po ok. 1min. mamy gotowe E2 na usb.

16. Wkładamy pendrive z E2 do dekodera. Jeśli wszystko zostało poprawnie wykonane, to po chwili na wyświetlaczu pojawi się napis **boot**, potem **nBoX**, potem **E2**, na końcu **LoAd** i zegarek z godziną **20:00**, która przy poprawnym ustawieniu satelity zsynchronizuje się. W przypadku Neutrino pojawiać się będzie napis **boot**, potem **Go n**, na końcu **LoAd** i zapali się nr kanału, w trybie StandBy zapali się godzina.

17. Skręcamy tuner i wszystko, cieszymy się z Linux'a w naszym NBox'ie

## Obsługa HDD przez złącze SATA

1. Odkręcamy ponownie obudowę naszego NBoxa, odnajdujemy złącze miejsce na płycie pod złącze SATA.

Są dwa typy płyt ze złączem przewlekany - foto poniżej



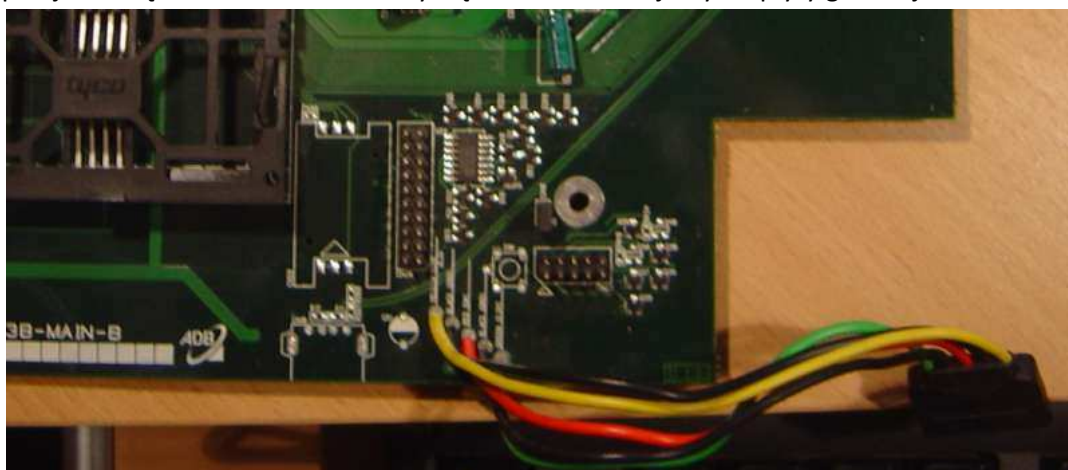
A także ze złączem powierzchniowym – foto poniżej



Nie ma to znaczenie dla pracy tunera, trzeba to złącze uzupełnić jak również kondensatory SMD **100nF** C91, C92, C93, C94, które są obok złącza po lewej stronie.

Złącze SATA można kupić nowe, aczkolwiek lepiej wydłubać z jakiejś starej płyty od komputera, czy konsoli. Można też oczywiście polutować bezpośrednio kabelek SATA do płyty głównej tunera.

2. Uzupełnić należy również zasilanie SATA dla naszego dysku, w tym celu kupujemy przejściówkę Molex-SATA obcinamy złącze Molex i lutujemy do płyty głównej.



Przewód 5 PINowy nie jest wymagany (zielony kabelek jest zbędny ale jak macie to można przylutować). Lutujemy kabelki wg. opisów kolorów na laminacie: żółty, czarny, czerwony, czarny.

3. Montaż i umocowanie dysku pozostawiam własnej inwencji.
4. Podłączamy nasz tuner bez PENDRIVE do **HyperTermial** i po kolei:

Board: Nbox [29-bit mode] by FREEBOX

U-Boot 1.3.1 (May 11 2011 - 17:17:33) - stm23\_0053

```
DRAM: 128 MiB
NOR:   4 MiB
In:    serial
Out:   serial
Err:   serial
Hit any key to stop autoboot: 0
(Re)start USB...
USB:   scanning bus for devices... 1 USB Device(s) found
       scanning bus for storage devices... 0 Storage Device(s)
found
** Bad partition 1 **
## Booting image at a5000000 ...
Bad Magic Number
nbox>
```

Programujemy same komendy ENV dla obsługi tuner z PENDRIVE bez HDD:

```
set bootargs 'console=ttyAS0,115200 root=/dev/sda1 rw mem=128m
coprocessor_mem=4m@0x10000000,4m@0x10400000 rootdelay=6
init=/bin/devinit' <enter>
```

```
setenv bootcmd 'usb reset;ext2load usb 0:1 a5000000
/boot/uImage;bootm a5000000' <enter>
```

```
saveenv <enter>
```

5. **UWAGA**, jeśli nie mamy do tej pory zainstalowanego HDD to, zrób to, bo tuner bez HDD po tych zmianach nie wystartuje! Dysk sformatowany musi być na **ext2**.
6. Idąc dalej wgrywamy Enigma2 v3.5 (lub nowszą) jest to pierwsza wersja, która radzi sobie z obsługą dysku SATA.
7. Jak Enigma2 odpali łączymy się, przez FTP z tunerem, szukamy pliku „etc/fstab” edytujemy go za pomocą NotePad++ (czy inny) i dodajmy wpis

```
/dev/sda1          /hdd                auto
defaults,errors=remount-ro,noatime,nodiratime          0 0
```

Po restarcie, zalogować się przez telnet i utworzyć katalogi:

```
mkdir /hdd/movies  
mkdir /hdd/picture  
mkdir /hdd/music
```

8. Teraz tuner będzie nam się odpalał z automatu z obsługą HDD i nie trzeba go będzie za każdym razem ręcznie montować.



### Image odpalane z NAND

1. Formatujemy nowy PENDRVIE pod Windows FAT32 (może być 128MB).
2. Podłączamy nasz tuner z do **HyperTermial** i po kolei:

Board: Nbox [29-bit mode] by FREEBOX

U-Boot 1.3.1 (May 11 2011 - 17:17:33) - stm23\_0053

```
DRAM: 128 MiB  
NOR: 4 MiB  
In: serial  
Out: serial  
Err: serial  
Hit any key to stop autoboot: 0  
nbox>
```

**loady** <enter>

Wybieramy plik ***u-boot\_nand\_fst.bin*** lub ***u-boot\_nand\_fsu.bin*** w zależności od tego jaki mamy typ kości.

**protect off A0200000 A024FFFF** <enter>

**erase A0200000 A024FFFF** <enter>

**cp.b 84000000 A0200000 32ed8** <enter>

3. Wyłączamy i włączamy ponownie tuner. Jeśli wszystko ok. zgłosi nam się nowy uBoot

Board: Nbox [29-bit mode] by FREEBOX

U-Boot 1.3.1 (Jun 11 2011 - 08:23:06) - stm23\_0053

DRAM: 128 MiB

NOR: 4 MiB

NAND: 64 MiB

\*\*\* Warning - bad CRC, using default environment

In: serial

Out: serial

Err: serial

nbox>

Wpisujemy po kolei:

**nand bad** <enter> (jeśli wyskoczy 0 to jest ok)

**nand scrub** <enter> jak zapyta czy yes czy no naciskamy **y** i <enter>

4. Wyłączamy tuner z prądu, na pendrive FAT32 wrzucamy plik **update.img** z konfiguracją multiboot.

Wciskamy przycisk POWER na Front Panelu i włączamy tuner do prądu. Po chwili pojawi nam się **boot** a następnie **PROG** jeśli wszystko przebiegnie ok pojawi się napis **—OK—**.

5. Wyłączamy znów tuner i na naszego pendrive FAT32 wrzucamy pliki **update.img rootfs.img kernel.img** (czyli cały soft do tunera) – przykładowo **v53\_bska\_e2\_nand\_11\_06\_2011**.

Wciskamy przycisk POWER na Front Panelu i włączamy tuner do prądu. Po chwili pojawi nam się **boot** a następnie **PROG**:

- jeżeli będzie jakiś problem z ładowaniem w/w pliku na wyświetlaczu pojawi się napis **FAIL**

- podczas aktualizacji będą wyświetlane komunikaty:

**L--1** - ładowanie pliku kernel.img do pamięci

**S--1** - zapis kernel.img do NANDa

**E--1** - jeżeli pojawi się błąd w fazie L--1

**L--2** - ładowanie pliku rootfs.img do pamięci

**S--2** - zapis rootfs.img do NANDa  
**E--2** - jeżeli pojawi się błąd w fazie L—2  
 jeśli wszystko przebiegnie ok pojawi się napis **—OK—**.

6. Wyłączamy tuner z prądu, włączamy ponownie i naciskamy strzałkę gór/ dół  
 Pojawiają nam się opcje bootowania  
 USBA – start image z USB (tuner bez HDD)  
 USBB – start image z USB (tuner z HDD)  
 HDD1 – start image z HDD  
 HDD2 – start image z HDD  
 HDD3 – start image z HDD  
 NAND – start image z pamięci wewnętrznej - NAND  
 NFS – start image podmontowane na dysku sieciowym

Po wybraniu opcji naciskamy przycisk OK na FP.

I to koniec, tuner odpala nam się z pamięci wewnętrznej, a my mamy dostępny wolny port USB.

## TO CO WAŻNE

Proponuję zawsze wrzucać przez Black Box najpierw uBoot **u-boot A020 Dla Rs232\_Zmiana\_01\_2011\_OK.bin** lub uBoot od @niebieski20 **u-boot i2c Niebieski wersja 2.bin**, ponieważ jest tak wiele interfejsów czasem zdarzają się błędy więc lepiej wrzucić w/w uBoot i za jego pomocą wrzucić docelowy.  
 Są 3 typy pamięci **ST: FST, FSB. FS** i jeden typ pamięci **AT** – do niego wrzucamy zawsze uBoot **FST**

W celu identyfikacji pamięci ST , gdy już mamy wgrany uBoot wpisujemy w HT: **fli** lub **flinfo**

nbox> **fli**

```
Bank # 1: CFI conformant FLASH (16 x 16) Size: 4 MiB in 32 Sectors
Intel Standard command set, Manufacturer ID: 0x20, Device ID: 0x0C
Erase timeout: 8192 ms, write timeout: 1 ms
Buffer write timeout: 1 ms, buffer size: 8 bytes
```

Sector Start Addresses:

|          |    |          |    |          |    |          |   |          |
|----------|----|----------|----|----------|----|----------|---|----------|
| A0000000 | R0 | A0020000 | R0 | A0040000 | R0 | A0060000 | E | A0080000 |
| A00A0000 |    | A00C0000 | E  | A00E0000 | E  | A0100000 |   | A0120000 |
| A0140000 | E  | A0160000 | E  | A0180000 |    | A01A0000 |   | A01C0000 |
| A01E0000 | E  | A0200000 |    | A0220000 |    | A0240000 |   | A0260000 |
| A0280000 |    | A02A0000 |    | A02C0000 | E  | A02E0000 | E | A0300000 |
| A0320000 | E  | A0340000 |    | A0360000 | E  | A0380000 | E | A03A0000 |
| A03C0000 |    | A03E0000 | R0 |          |    |          |   |          |

**0x0A** – FST – wrzucamy uBoot FST  
**0x0B** – FSB – wrzucamy uBoot FST  
**0x0C** – FSU – wrzucamy uBoot FSU

**Wrzucanie nowego uBoot'a którego nie ma w/w opisie, na co zwrócić uwagę:**

Board: Nbox [29-bit mode] by FREEBOX

U-Boot 1.3.1 (Jun 12 2011 - 20:09:26) - stm23\_0053

```
DRAM: 128 MiB
NOR: 4 MiB
NAND: 64 MiB
In: serial
Out: serial
Err: serial
Hit any key to stop autoboot: 0
nbox> loady
## Ready for binary (ymodem) download to 0x84000000 at 115200
bps...
CCxyzModem - CRC mode, 2(SOH)/204(STX)/0(CAN) packets, 4
retries
## Total Size = 0x00032f18 = 208664 Bytes
nbox> protect off A0000000 A005FFFF
... done
Un-Protected 3 sectors
nbox> erase A0000000 A005FFFF

... done
Erased 3 sectors
nbox> cp.b 84000000 A0000000 32f18
Copy to Flash
..... done
nbox>
```

Opis komend:

**loady** - ładowanie do RAM pliku bin protokołem yModem pod adres 84000000  
**protect off A0000000 A005FFFF** - odbezpiecza sektory od A0000000 do A005FFFF  
**erase A0000000 A005FFFF** - czyści sektory w zakresie jak wyżej  
**cp.b 84000000 A0000000 32F18** - kopiuje plik binarny z RAM z adresu 84000000 do A0000000 o wielkości 32F18

Wartość zaznaczona na czerwono jest różna dla każdego uBoot'a, więc nie można bezmyślnie kopiować poleceń. Jest to przykładowy log programowania nowego uBoot, w typ przypadku do sektora A0000000.

Ile wyczyścić sektorów przykładowo uBoot zajmuje 208 664 byte. Odpalamy kalkulator w Windows wpisujemy **208 664** zamieniamy na HEX = **32F18**.

W kalkulatorze dodajemy A0200000+32F18 = A0232F18 czyli czyścimy od A0200000 do A023FFFF

**protect off A0200000 A023FFFF** <enter>

**erase A0200000 A023FFFF** <enter>

**cp.b 84000000 A0200000 32f18** <enter>

Po uwagach jednego z kolegów z forum sat-elita opis został zmodyfikowany. Zalecane jest programowanie kilku uBootów do różnych sektorów (najlepiej co 4). Zabezpieczamy się przed przypadkowym skasowaniem uBoot, a także uszkodzenia pkt. zworki.

Np. A0040000, A0080000, A0100000, A0140000, A0180000, A0200000, A0240000 itd.

Stan wysoki (3,3V) można wtedy podać na inną nóżkę pamięci i dzięki temu tuner w przypadku skasowania nie tego co trzeba, albo uszkodzenia zworki można odratować bez skrobania kleju.

Przykładowy zapis – log:

Board: Nbox [29-bit mode] by FREEBOX

U-Boot 1.3.1 (Jun 11 2011 - 08:23:06) - stm23\_0053

DRAM: 128 MiB

NOR: 4 MiB

NAND: 64 MiB

In: serial

Out: serial

Err: serial

Hit any key to stop autoboot: 0

nbox> loady

## Ready for binary (ymodem) download to 0x84000000 at 115200 bps...

CxyzModem - CRC mode, 2(SOH)/172(STX)/0(CAN) packets, 3 retries

## Total Size = 0x0002af38 = 175928 Bytes

nbox> protect off A0180000 A01AFFFF

... done

Un-Protected 3 sectors

nbox> erase A0180000 A01AFFFF

... done

Erased 3 sectors

nbox> cp.b 84000000 A0180000 2af38

Copy to Flash

..... done

nbox> protect off A0100000 A013FFFF

.... done

Un-Protected 4 sectors

nbox> erase A0100000 A013FFFF

```

.... done
Erased 4 sectors
nbox> cp.b 84000000 A0100000 2af38
Copy to Flash
..... done
nbox> protect off A0140000 A017FFFF
.... done
Un-Protected 4 sectors
nbox> erase A0140000 A017FFFF

.... done
Erased 4 sectors
nbox> cp.b 84000000 A0140000 2af38
Copy to Flash
..... done
nbox>

```

I sprawdzamy czy zapisało poprawnie:

```

nbox> go a0100000

## Starting application at 0xA0100000 ...

Board: STb7100-Reference (MB442) [29-bit mode]

U-Boot 1.3.1 (Nov 11 2010 - 01:21:23) - stm24_0047

DRAM: 128 MiB
NOR: 4 MiB
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
Hit any key to stop autoboot: 0
MB442 niebieski20> go a0180000
## Starting application at 0xA0180000 ...

Board: STb7100-Reference (MB442) [29-bit mode]

U-Boot 1.3.1 (Nov 11 2010 - 01:21:23) - stm24_0047

DRAM: 128 MiB
NOR: 4 MiB
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
Hit any key to stop autoboot: 0
MB442 niebieski20> go a0140000
## Starting application at 0xA0140000 ...

Board: STb7100-Reference (MB442) [29-bit mode]

U-Boot 1.3.1 (Nov 11 2010 - 01:21:23) - stm24_0047

DRAM: 128 MiB
NOR: 4 MiB
*** Warning - bad CRC, using default environment

```

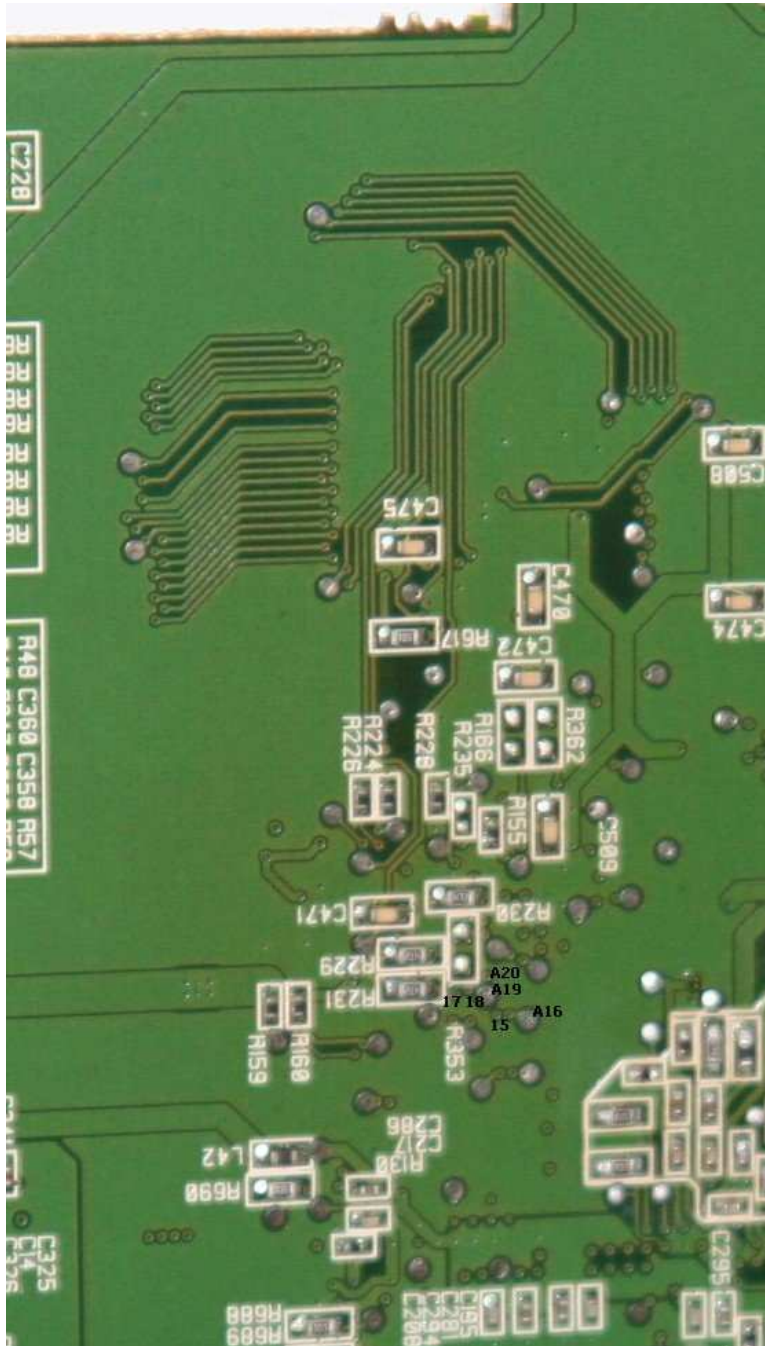
```
In: serial  
Out: serial  
Err: serial  
Hit any key to stop autoboot: 0  
MB442 niebieski20>
```

I oczywiście pkt. zapasowych zworek:

A20 = A0200000

A19 = A0100000

A18 = A0080000



Mam nadzieję, że jest to dość proste dla osób nie wtajemniczonych w programowanie.

## I na koniec przykładowy log z ładowania image do wewnętrznej pamięci tunera – NAND

U-Boot 1.3.1 (Jun 17 2011 - 21:16:40) - stm23\_0053

```
DRAM: 128 MiB
NOR:   4 MiB
NAND:  64 MiB
In:     serial
Out:    serial
Err:    serial
Hit any key to stop autoboot: 0
(Re)start USB...
USB:   scanning bus for devices... 1 USB Device(s) found
       scanning bus for storage devices... 0 Storage Device(s) found
** Bad partition 1 **
## Booting image at a5000000 ...
Bad Magic Number
nbox> nand bad
```

```
Device 0 bad blocks:
nbox> nand scrub
```

```
NAND scrub: device 0 whole chip
Warning: scrub option will erase all factory set bad blocks!
        There is no reliable way to recover them.
        Use this command only for testing purposes if you
        are sure of what you are doing!
```

```
Really scrub this NAND flash? <y/N>
Erasing at 0x3ffc000 -- 100% complete.
OK
nbox>
```

Board: Nbox [29-bit mode] by FREEBOX

U-Boot 1.3.1 (Jun 17 2011 - 21:16:40) - stm23\_0053

```
DRAM: 128 MiB
NOR:   4 MiB
NAND:  64 MiB
In:     serial
Out:    serial
Err:    serial
(Re)start USB...
USB:   scanning bus for devices... 2 USB Device(s) found
       scanning bus for storage devices... 1 Storage Device(s) found
reading update.img
```

```
450 bytes read
## Executing script at 84000000
reading kernel.img
.....
.....
.....
```

```
2321785 bytes read
device 0 whole chip
nand_unlock: start: 00000000, length: 67108864!
NAND flash successfully unlocked
```

```
NAND erase: device 0 offset 0x3c00000, size 0x400000
Erasing at 0x3ffc000 -- 100% complete.
OK
```

```
NAND write: device 0 offset 0x3c00000, size 0x236d79
```

```
Writing data at 0x3e36c00 -- 100% complete.
```

2321785 bytes written: OK

reading rootfs.img

```
50528256 bytes read
device 0 whole chip
nand_unlock: start: 00000000, length: 67108864!
NAND flash successfully unlocked

NAND erase: device 0 offset 0x0, size 0x3c00000
Erasing at 0x3bfc000 -- 100% complete.
OK

NAND write: device 0 offset 0x0, size 0x3030000

Writing data at 0x302fe00 -- 100% complete.
50528256 bytes written: OK
```

**Wszelkie nowości w sprawie NBox'a na <http://sat-elita.net.pl/forum/>**

Nie jestem autorem żadnego z tych programów, ani też rozwiązań, które tu zostały podane, zebrałem to tylko w jedną całość, gdyż sam też miałem z tym sporo problemów. Odpowiedzialność za szkody, które wyrządzisz bierzesz sam na siebie, czytaj dwa razy zanim zaczniesz. Jak nie czujesz się na siłach, by to zrobić, zleć to komuś, kto się czuje by to wykonać poprawnie i nie zepsuć sprzętu. Podziękowania dla kolegów z forum **sat-elita.net.pl**

**Konstantynopolitańczykkowianeczkasześć**

**Niezbędne linki:**

Od tego wszystko się zaczęło -> <http://freebox.lamerek.com>

Najnowsze IMAGE do tunerów -> <http://freebox.lamerek.com>

Najnowsze uBoot do tunerów -> <http://freebox.lamerek.com>

Polecam też IMAGE do nBox od PKT&B4Team -> <http://www.pkteam.pl>

Wszelkie info i nowości na forum w dziale NBox HD -> <http://sat-elita.net.pl/forum/>

Niezbędne uBoot, narzędzia -> <http://www.megaupload.com/?d=MC6SPVVH>

HD Clone 3.9.3 -> <http://www.megaupload.com/?d=KUBJ2RQ8>

Tutorial v4 – w sumie warto się zapoznać -> <http://www.megaupload.com/?d=COM810WQ>

Tutorial v5.2 – też warto się zapoznać -> <http://www.megaupload.com/?d=T2OXL54D>