

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1237

**UGRADNJA OPERACIJSKOG SUSTAVA
uCLINUX NA DSP PROCESORE PORODICE
BLACKFIN**

Nader Salameh

Zagreb, lipanj 2010

Sadržaj

Sadržaj.....	1
Uvod.....	2
1. ADSP-BF537 EZ-KIT Lite	3
2. Učitavanje U-Boot programa na pločicu	6
2.1. Metode <i>bootanja</i>	6
2.2. Memorijska mapa ADSP BF537 EZ-KIT LITE.....	7
2.3. Serijsko spajanje računala i uređaja	10
2.4. Učitavanje U-Boot-a.....	12
2.4.1. Programiranje flash-a pomoću JTAG-a.....	12
2.4.2. Programiranje flash-a pomoću UART-a.....	15
3. Učitavanje slike uCLinux-a	18
3.1. Učitavanje linux slike pomoću TFTP servera	18
3.1.1. Postavljanje TFTP servera.....	18
3.1.2. Učitavanje slike uCLinux-a.....	21
3.2. Učitavanje linux slike pomoću <i>Multimedia</i> kartice.....	22
3.2.1. Spajanje kartice na pločicu	22
3.2.2. Učitavanje slike uCLinux-a.....	26
4. Pokretanje jednostavne linux aplikacije	28
4.1. Pisanje i <i>kompajliranje</i> aplikacije.....	28
4.2. Učitavanje i pokretanje aplikacije preko mreže	30
Zaključak.....	31
Literatura.....	32
Naslov, sažetak i ključne riječi	33
Title, summary, keywords	34

Uvod

Cilj završnog rada je istražiti postupke ugradnje operacijskog sustava u procesore ugradbenih računalnih sustava radi lakšeg upravljanja pločicom, te mogućnosti korištenja raznih programa kako bi se procesor mogao u potpunosti iskoristiti. U ovom slučaju kao platforma će se koristiti ADSP-BF537 EZ-KIT Lite firme Analog Devices, dok će operacijski sustav biti uCLinux, grana Linux jezgri specijalizirana za mikrokontrolere. Kako bi se postigao konačni cilj, potrebno je napraviti glavne tri točke koje će ujedno biti tri glavne cjeline završnog rada:

1. učitavanje *U-Boot* programa koji će služiti za pokretanje operacijskog sustava
2. učitavanje komprimirane slike operacijskog sustava preko mogućih medija
3. prevođenje, učitavanje i pokretanje jednostavne Linux aplikacije na ADSP-BF537 EZ-KIT razvojnom modulu

Učitavanje U-Boot programa provodit će se pomoću određenih alata, predviđenih za pisanje u memoriju različitih razvojnih modula. Nakon učitavanja U-Boot programa komprimiranu sliku potrebno je učitati sa jednog od više medija, kojima može raspolagati platforma ADSP-BF537 EZ-KIT Lite. Konačno pomoću pokretanja jednostavne Linux aplikacije provjerava se uspješnost učitavanja operacijskog sustava na pločicu. Sve tri cjeline detaljno će biti opisane u sljedećim poglavljima.

Za uspješno provođenje svih predviđenih ciljeva potrebno je upoznati platformu s kojom će se rukovati, te će se tako u nastavku govoriti i o najbitnijim značajkama ADSP-BF537 EZ-KIT Lite razvojnog modula.

1. ADSP-BF537 EZ-KIT Lite

Porodica Blackfin procesora utjelovljuje novi tip ugrađenog procesora dizajniranog posebno kako bi se suočio sa računalnim zahtjevima i ograničenjima snage današnjih ugrađenih, audio, video i komunikacijskih aplikacija. Oni donose novosti u performansama obrade signala i energetske učinkovitost unutar RISC (*reduced instruction set computing*) programskog modela.

Blackfin procesori podržavaju MISC (*media instruction set computing*) arhitekturu. Ova arhitektura je rezultat spajanja RISC-a, medijskih funkcija, i DSP (*digital signal processing*) karakteristika. Blackfin procesori nude obradu signala u mikroprocesorskom okruženju.

ADSP-BF 537 procesor jedan je u nizu procesora porodice BlackFin, koji nudi značajne visoke performanse, malu snagu ali i jednostavnost korištenja svih pogodnosti koje taj procesor sadrži.

Razvojni modul je dizajniran da bude u službi zajedno sa VisualDSP++ razvojnom okolinom kako bi se mogle testirati mogućnosti ADSP-BF537 Blackfin procesora. VisualDSP++ razvojna okolina omogućuje :

- stvaranje, sastavljanje, prevođenje i povezivanje programa napisanih u programskom jeziku C++ i jeziku C
- učitavanje, pokretanje, zaustavljanje te postavljanje zaustavnih točaka u programima
- čitanje i pisanje podatkovne i programske memorije
- čitanje i pisanje s registara jezgre i periferija

Za ovaj završni rad najviše će se koristiti upravo čitanje i pisanje sa podatkovne i programske memorije kako bi se u memoriju razvojnog modula postavio U-Boot.

Pristup ADSP-BF537 procesoru sa osobnog računala omogućen je preko USB porta ili opcionalnog JTAG emulatora. USB sučelje daje neograničen pristup ADSP-BF537 procesoru i periferijama pločice.

Elementi i svojstva pločice :

- Analog Devices ADSP-BF537 Blackfin procesor
 - o performanse jezgre do 600 MHz
 - o performanse vanjske sabirnice do 133 MHz
 - o 182-pinsko mini-BGA kućište
 - o kristal od 25 MHz
- sinkrona dinamička memorija sa izravnim pristupom (SDRAM)
 - o MT48LC32M8 – 64 MB (8M x 8-bit x 4 bloka) x 2 integrirana kruga
- Flash memorija
 - o 4 MB (2M x 16-bits)
- Analogno audio sučelje
 - o AD1871 96 kHz analogno-digitalni koder (ADC)
 - o AD1854 96 kHz digitalno-analogni koder (DAC)
 - o 1 ulazni stereo utikač
 - o 1 izlazni stereo utikač
- Ethernet sučelje
 - o 10-BaseT (10Mbit/s) i 100-BaseT (100 Mbit/s)
- Ethernet MAC (Media Acces Controller)
 - o uređaj SMSC LAN83C185
- CAN (Controller Area Network) sučelje
 - o Philips TJA1041 CAN primopredajnik velike brzine
- UART (Universal asynchronous reciever/transmitter)
 - o ADM3202 RS-232 izlazno pojačalo/odašiljač
 - o DB9 ženski konektor
- Diode koje zrače svjetlost (LED)
 - o 10 LE dioda: 1 za uključivanje/isključivanje (zelena), 1 za reset pločice (crvena), 1 za USB (crvena), 6 općenamjenskih (žute), 1 USB prikaznik (žuta)
- Tasteri
 - o 5 tastera: 1 reset, 4 programibilnih zastavica sa debounce logikom
- Ostale značajke
 - o JTAG ICE 14-pinsko zaglavlje

U prethodnom tekstu navedene su samo osnovne informacije o razvojnom modulu ADSP-BF537 EZ-KIT Lite. Sve ostale informacije o pločici, koje će se ticati završnog rada, biti će detaljno objašnjene u daljnjim poglavljima, gdje će za realiziranje ciljeva određenih završnim radom, biti potrebno malo dublje zaviriti u pločicu. U nastavku poglavlja priložena je slika pločice ADSP-BF537 EZ-KIT Lite (Slika 1).



Slika 1- ADSP-BF537 EZ-KIT Lite

2. Učitavanje U-Boot programa na pločicu

U-Boot je program za podizanje sustava (*bootanje*), koji je obično spremljen u flash memoriju ugrađenog sustava. U-Boot je u stanju učitavati datoteke sa različitih periferija, pa se tako može učitavati preko, serijske veze, *ethernet* mrežne veze ili *flash* memorije. Često se događa situacija, gdje je U-Boot program koji se automatski učitava pri paljenju pločice, te tako služi za automatsko pokretanje drugih aplikacija, kao što će to u ovom radu biti uCLinux jezgra. U-Boot također može *parsirati* mnoge oblike sustava datoteka (*file system*) na različitim tipovima uređaja.

2.1. Metode *bootanja*

Blackfin procesori podržavaju različite metoda učitavanja i pokretanja U-Boota, ovisno o specifikacijama procesorske varijante. Sve varijante koje podržava Boot ROM na integriranom krugu, podržava i U-Boot. Ograničenje koje ima Boot ROM, na integriranom krugu, ne mora nužno biti otežavajuća okolnost. Naime, kada se postavi i pokrene U-Boot, ostale aplikacije ali i operacijski sustavi mogu se učitavati sa svih periferija koje podržavaju Blackfin procesor i U-Boot. To znači da se na nekim Blackfin procesorima, gdje se Boot ROM pokreće samo sa paralelnog ili serijskog flash-a, U-Boot može pokretati druge aplikacije sa mreže, iz NAND-a, sa tvrdog diska, sa kompaktne flash kartice, USB spremnika ili bilo koje druge periferije/uređaja. U-Boot može prevoditi neke formate datotečnih sustava (*file system*) što znači da se aplikacija/jezgra može spremiti unutar EXT2/3, JFFS2, ili FAT32 particije.

Boot ROM na procesoru podiže jedan format slike koji je kod Blackfina LDR datoteka. U-Boot *build process* automatski ostvaruje format datoteke (obično nazvan u-boot.ldr) koristeći *ldr-utils* paket koji je dio Blackfin *toolchain-a*. U ovom završnom radu, ldr datoteka koja sadrži sliku U-Boot-a već je napravljena i stavljena na CD koji se dobije uz pločicu ADSP-BF537 EZ-KIT Lite, pa je nema potrebe generirati preko *ldr-utils* paketa. Sve što je potrebno napraviti je učitati ldr datoteku na uređaj sa kojeg mislimo *bootat*, a to će detaljno biti objašnjeno u potpoglavlju **Učitavanje U-Boot-a**.

Kako bi se provjerilo je li U-Boot ispravno učitano na flash, koristit ćemo serijsku vezu između računala i pločice preko RS-232 priključka, stoga će se prije samog programiranja flash-a, opisati postupak uspostavljanja veze između računala i uređaja.

2.2. Memorijska mapa ADSP BF537 EZ-KIT LITE

Prije samog učitavanja U-Boot-a, potrebno je odrediti gdje ćemo ga učitati, stoga je neophodno poznavanje memorijske mape razvojnog modula ADSP BF537 EZ-KIT LITE.

ADSP-BF537 procesor ima internu SRAM (*Static Random Access Memory*) koja može poslužiti kao instrukcijska ili podatkovna memorija. Pločica sadrži 2 tipa vanjske memorije, SDRAM i flash. Veličina SDRAM-a je 64 MBajta (32M*16 bita). Veličina flash memorije je 4 MBajta (2M*16 bita). Detaljne informacije dostupne su u **tabeli 1**.

Tabela 1 - Memorijska mapa

	Početna adresa	Završna adresa	Sadržaj
Vanjska memorija	0x0000 0000	0x03FF FFFF	SDRAM blok 0
	0x2000 0000	0x200F FFFF	ASYNČ memorijski blok 0
	0x2010 0000	0x201F FFFF	ASYNČ memorijski blok 1
	0x2020 0000	0x202F FFFF	ASYNČ memorijski blok 2
	0x2030 0000	0x203F FFFF	ASYNČ memorijski blok 3
	0x203F 0000		MAC adresa
	sve ostale lokacije		neiskorišteno
Unutarnja memorija	0xFF80 0000	0xFF80 3FFF	Podatkovni blok A SRAM 16 KB
	0xFF80 4000	0xFF80 7FFF	Podatkovni blok A SRAM/CACHE 16 KB
	0xFF90 0000	0xFF90 7FFF	Podatkovni blok B SRAM 16 KB
	0xFF90 4000	0xFF90 7FFF	Podatkovni blok B SRAM/CACHE 16 KB
	0xFFA0 0000	0xFFA0 7FFF	Instrukcijski blok A SRAM 32 KB
	0xFFA1 0000	0xFFA1 3FFF	Instrukcijski blok B SRAM 16 KB
	0xFFA0 8000	0xFFA0 BFFF	Instrukcijska SRAM/CACHE 16 KB
	0xFFB0 0000	0xFFB0 0FFF	<i>Scratch pad</i> SRAM 4 KB
	0xFFC0 0000	0xFFDF FFFF	Sustavski MMRs 2 MB
	0xFFE0 0000	0xFFFF FFFF	Jezgreni MMRs 2 MB
	sve ostale lokacije		rezervirane

Za pohranu U-Boot-a, SDRAM neće poslužiti jer se nakon svakog reseta brišu svi podaci sa njega, te ćemo zbog toga koristiti flash memoriju. Veličina flash memorije određena je prekidačem SW6 pomoću kojeg se odabire koliko MB je dostupno korisniku. Podrazumijevano je da su sve sklopke postavljene na ON. Svaka sklopka odgovara za jedan MB.

Osim prekidača SW6, za učitavanje U-Boot-a važan je i rotacijski prekidač SW16. On naime određuje s koje memorije će se provoditi *bootanje*. Podrazumijevana vrijednost je *bootanje* sa flash memorije, što će ujedno biti i vrijednost postavljena u ovom radu. Detalji o prekidaču SW16 nalaze se u **tabeli 2**.

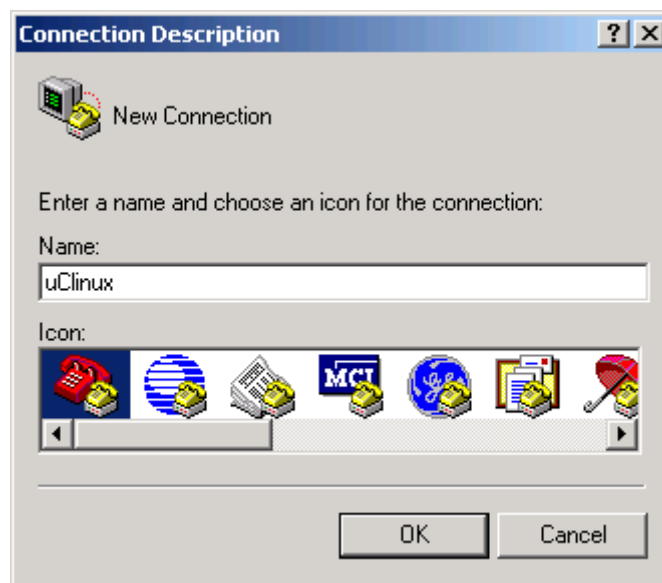
Tabela 2 - prekidač SW16

SW16 pozicija	Način <i>bootanja</i>
0	Pokreni sa 16-bitne vanjske memorije
1	<i>Bootaj</i> sa 16-bitne flash memorije
2	Rezervirano
3	<i>Bootaj</i> sa SPI memorije
4	<i>Bootaj</i> sa SPI domaćina
5	<i>Bootaj</i> sa serijske TWI memorije
6	<i>Bootaj</i> sa TWI domaćina
7	<i>Bootaj</i> sa UART domaćina

2.3. Serijsko spajanje računala i uređaja

Kao to je već prije navedeno, za vezu između računala u razvojnog modula ADPS BF537 EZ-KIT lite koristit ćemo serijsku vezu ostvarenu preko RS-232 kabla. Terminal koji će se koristiti je popularan *Hyper Terminal* koji obično dolazi u paketu sa Windowsima.

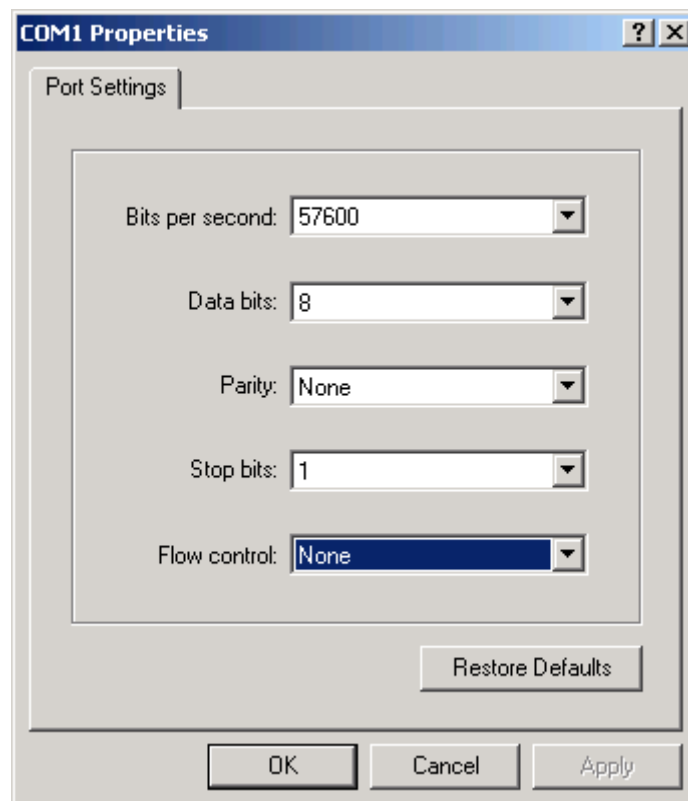
Program se obično nalazi pod *Start>Program files> Accessories> Communication> Hyper Terminal*, pa se i odavde pokreće. Pri pokretanju se pojavljuje prozor *Connect Description*, gdje se pod polje Name upiše bilo koja vrijednost npr. uCLinux (**Slika 2**). Sljedeći prozor koji se pojavljuje je *Connect to*. Ovdje se u polju *Connect using* treba izabrati priključak koji se koristi npr. COM 1 (**Slika 3**). Konačno, pojavljuje se prozor *COMx properties*, gdje se odabire brzina od 57600 bita po sekundi, 8 podatkovnih bitova bez pariteta, jedan zaustavni bit te se upravljanje tokom postavljuje kao *None* (**Slika 4**). Nakon ovog koraka veza bi trebala biti uspostavljena .



Slika 2



Slika 3



Slika 4

2.4. Učitavanje U-Boot-a

Flash memorija na razvojnom modulu ADSP-BF537 EZ-KIT Lite bi trebala imati već postavljen U-Boot. No ako imamo loš ili zastarjeli U-Boot, ili ako smo slučajno izbrisali U-Boot potrebno je ponovno učitati U-Boot na pločicu.

Danas su najraširenije dvije osnovne metode za učitavanje U-Boot-a a to su :

1. Programiranje flash-a pomoću JTAG-a
2. *Bootanje* pomoću UART-a

Svaka od metoda učitavanja sadrži nekoliko rješenja za učitavanje U-Boot-a. U ovom radu objasniti će se obje metode, svaka sa jednim rješenjem.

2.4.1. Programiranje flash-a pomoću JTAG-a

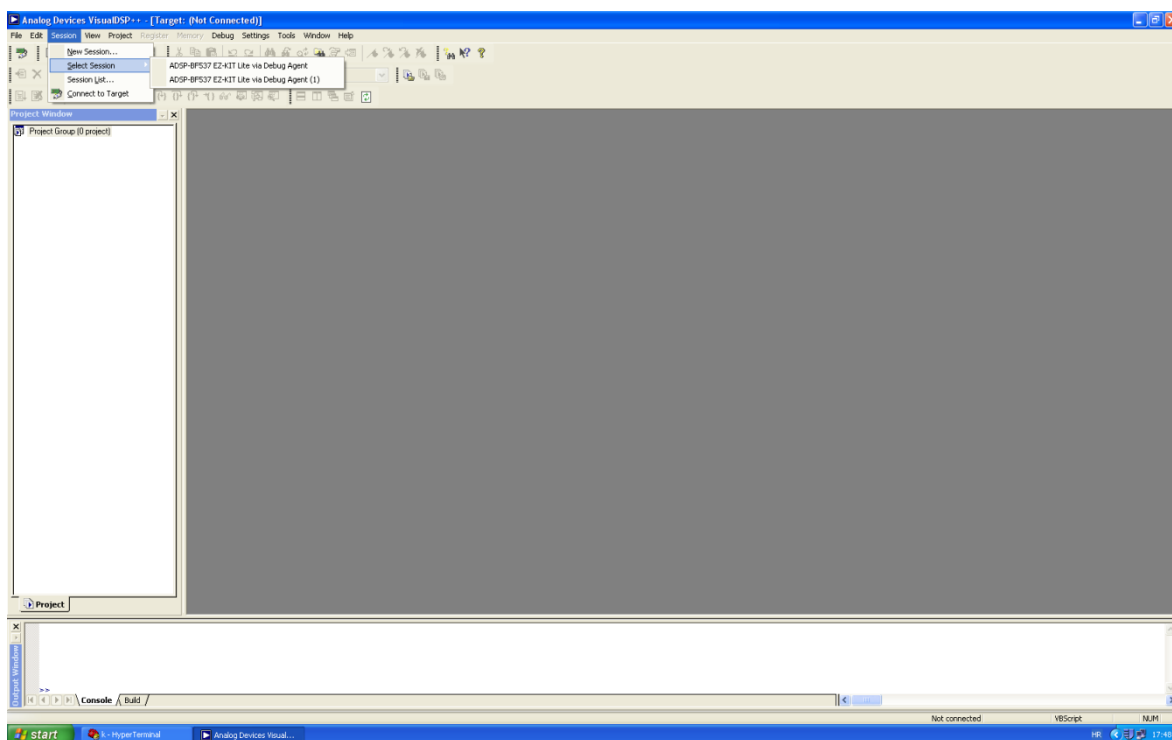
Kao što je već ranije napomenuto, za postavljanje U-Boot-a na flash preko JTAG-a, postoji više različitih rješenja, a u nastavku su navedeni najpopularniji:

1. gnICE (Linux, Windows, Mac OS X)
2. gnICE+ (linux, windows, Mac OS X)
3. IGLOO *Parallel port* JTAG (Linux)
4. JTAG *Blue* (Linux)
5. Visual DSP++ ADI ICE (Windows)

Ovdje ćemo opisati samo zadnje rješenje, koje se koristi isključivo na Windowsima.

Kako bi počeli sa programiranjem, potrebno je najprije instalirati program Visual DSP++. Program se nalazi na CD-u koji smo dobili sa razvojnim modelom ADSP BF-537 EZ-KIT Lite. Konkretna verzija programa na kojoj će se objasniti postupak je 5.0.

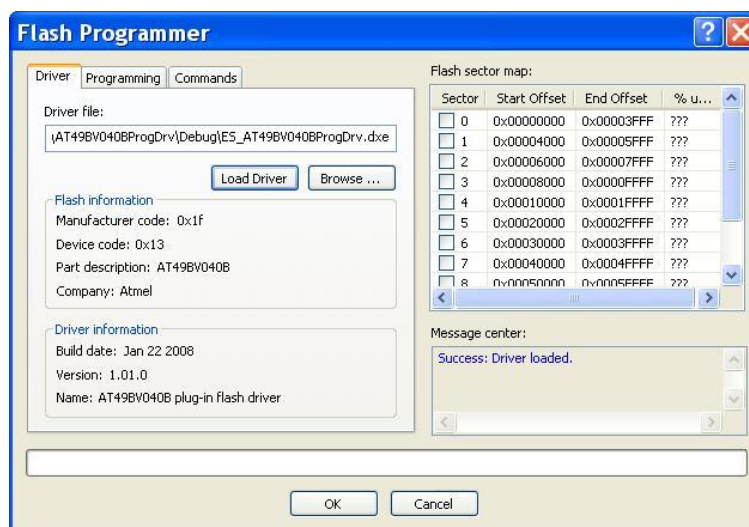
Nakon što je program uspješno instaliran, potrebno je spojiti uređaj sa osobnim računalom i prekidač SW16 postaviti na 1. Uređaj na sebi sadrži priključak USB-a tipa B, dok računalo sadrži tip A, pa će se veza najlakše ostvariti preko USB kabela. Sada se pokreće program, u izborniku se odabire Session, te se nalazi željeni uređaj (**Slika 5**).



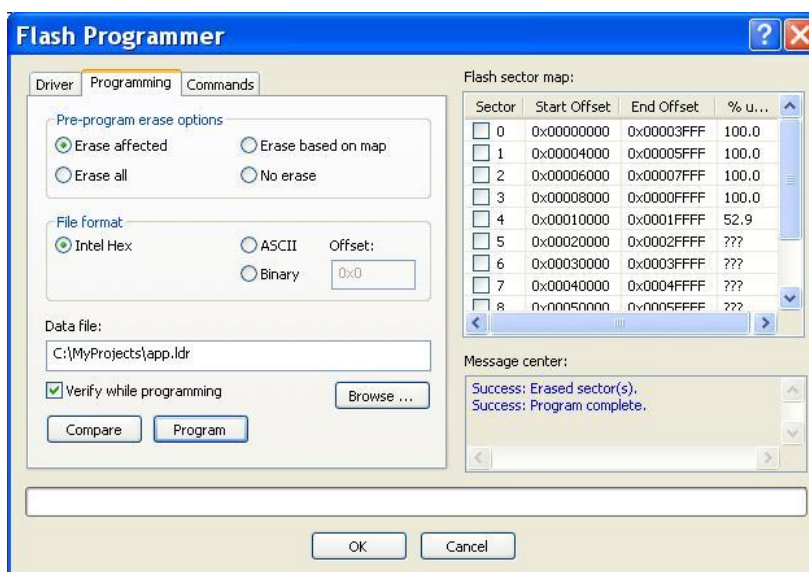
Slika 5 – Visual DSP++

Sljedeći korak je pokretanje *Flash Programmer-a*, koji se nalazi u izborniku *Tools*. Klikom na *Flash Programmer* otvara se novi izbornik i kartica *driver* kako je prikazano na **slici 6**. Učitava se *driver* koji je najbolji za pločicu koja se koristi, te stisnite gumb *Load driver* (u slučaju manjka prikladnog *driver-a* isti se može pronaći na netu). *Message center* u donjem desnom uglu, ukoliko se driver uspješno učitao, javlja poruku "Success: Driver loaded". Pritiskom na karticu *Programming* ulazimo u novi izbornik prikazan na **slici 7**. U polju *Pre-program erase options* uključuje se opcija *Erase all*, kako bi se izbrisao cijeli sadržaj koji je do tada postao na flash-u. Format datoteke koje ćemo učitati potrebno je pripremiti u *binary* formatu pa se uključuje i ta opcija. Kako je već prije napomenuto u radu, Boot ROM u ovom slučaju podiže sliku U-Boot-a formata ldr,

pa je potrebno staviti tu datoteku kao *Data file*. Ldr datoteka U-Boot-a nalazi se na CD-u koji se dobije uz pločicu, a najnovije verzije mogu se naći na stranici <http://blackfin.uclinux.org/gf/>. Poželjno je uključiti opciju *Verify while programing*, kako ne bi došlo do greške prilikom programiranja flash-a. Nakon ovih podešavanja, konačno uključujemo programiranje pritiskom na gumb *Program*. Ukoliko se U-Boot uspješno učitao u Message centru dolazi poruka "Success: Erased sector(s)." te "Success: Program complete" . Provjeru možemo obaviti pomoću opcije *Compare* koja uspoređuje flash i ldr datoteku.



Slika 6 – Flash Programmer (Driver)

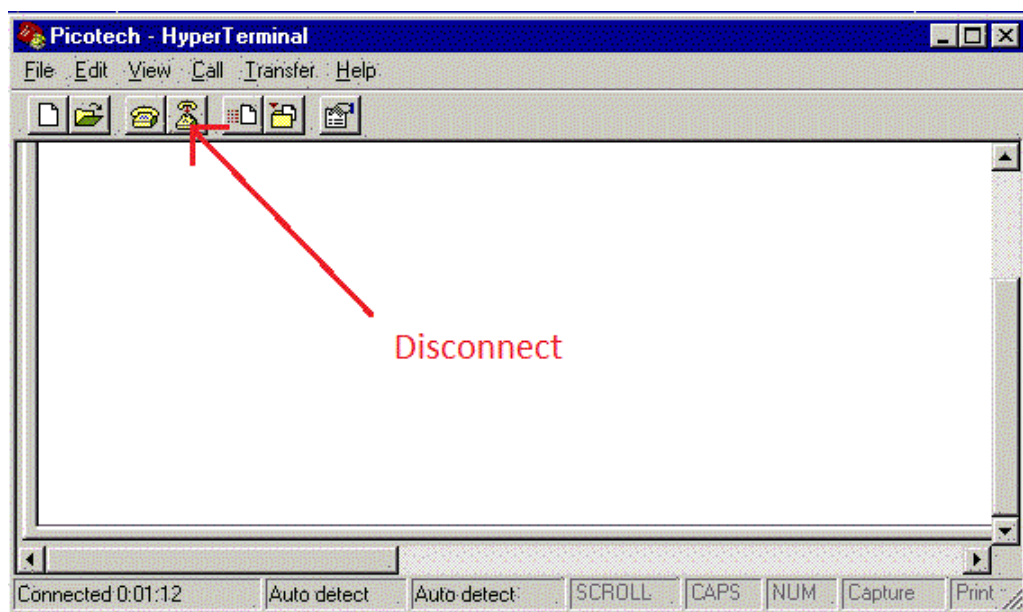


Slika 7 – Flash Programmer (Programming)

2.4.2. Programiranje flash-a pomoću UART-a

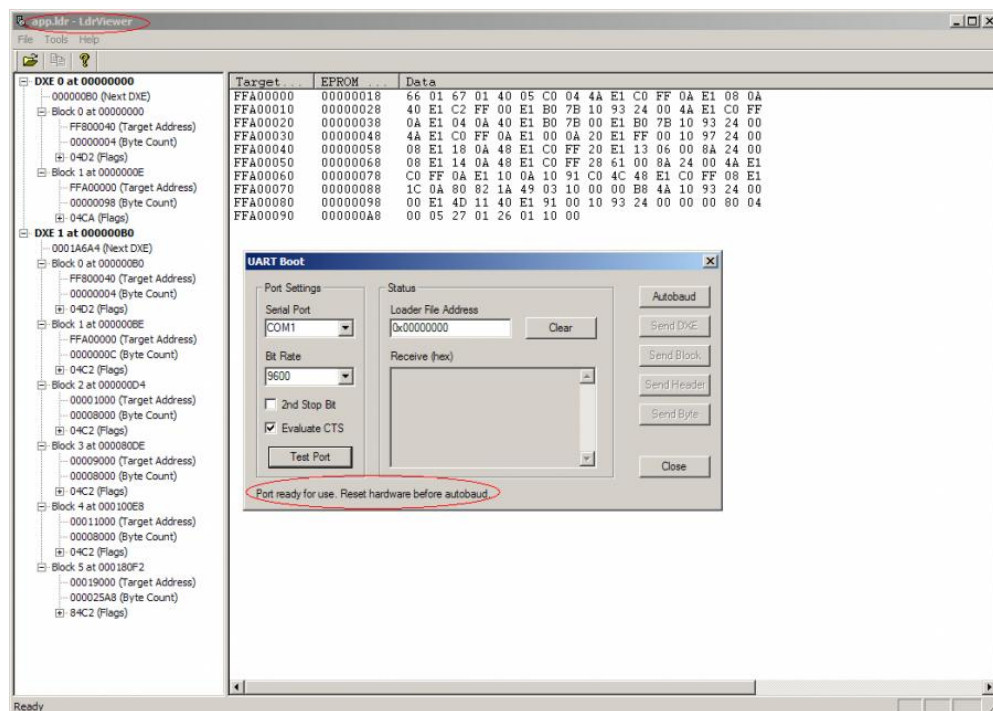
Postoji više načina učitavanja U-Boot-a preko UART-a, no ovdje će se opisati samo jedan postupak (za Windows-e) preko programa *Ldr viewer*. Za početak, potrebno je instalirati program te prebaciti prekidač SW16 u poziciju 1 ako to već nije napravljeno.

U poglavlju **Serijsko spajanje računala i uređaja**, objašnjeno je kako bi veza računala i uređaja trebala biti postavljena, pa se pretpostavlja da je to već učinjeno. Prije pokretanja samog programa *Ldr viewer*-a, u Hyper terminalu-u veza sa uređajem se mora prekinuti, pritiskom na gumb *Disconnect* (**Slika 8**).



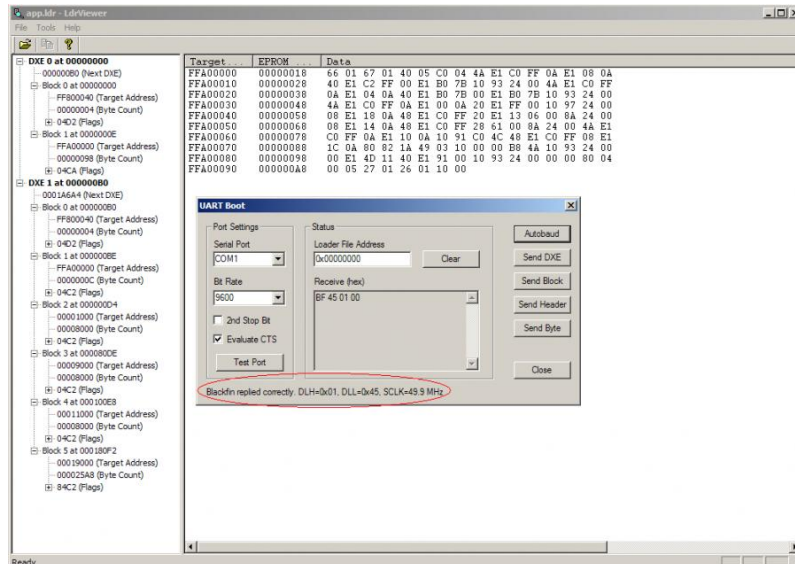
Slika 8 - Hyper terminal

Sada kada je veza prekinuta pokreće se Ldr viewer. U izborniku Tools > UART Boot odabiremo ldr datoteku koju želimo spremati. Posebnu pažnju treba posvetiti tome da će se učitati ldr datoteka koja će se pokretati sa flash-a a ne sa UART domaćina (prekidač SW16 u poziciji 1). Brzina prijenosa podataka namješta se na 115200 bauda, a serijski port se odabire kao u poglavlju **Serijsko spajanje računala i uređaja**. Pritiskom na gumb *Test Port button* trebala bi se pojaviti poruka *Port ready for use. Reset hardware before autobaud.* kao na **slici 9**.



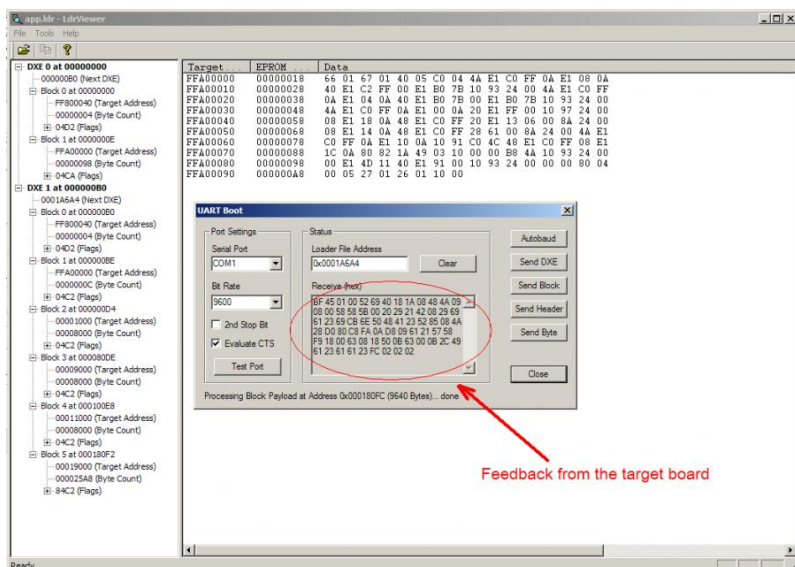
Slika 9 - Ldr viewer (testiranje priključka)

Kao što je program upozorio, potrebno je resetirati uređaj, a nakon toga se pritišće gumb *Autobaud*. U slučaju uspješnosti trebali bi dobiti poruku da je uređaj odgovorio: *Blackfin replied correctly* (Slika 10).



Slika 10 - Ldr viewer (Autobaud)

Zadnji korak koji predstoji je poslati datoteku koja je već otvorena pritiskom na gumb *Send DXE*. Odgovor programa trebao bi izgledati kao na slici 11. Ovime bi se U-Boot trebao učitati na flash, a provjera se obavlja tako da ugasimo Ldr viewer i pritisnemo gumb *Connect* na Hyper terminalu. Nakon stiskanja tipke ENTER trebala bi se pojaviti poruka tipa *bfin>*.



Slika 11 - Ldr viewer (povratna informacija)

3. Učitavanje slike uCLinux-a

Sada kada je U-Boot postavljen u flash memoriju preostaje učitati komprimiranu sliku operacijskog sustava uCLinux, na pločicu. U ovom radu objasnit će se dva načina za učitavanje Linux jezgre na uređaj:

1. Učitavanje slike preko *TFTP servera*
2. Učitavanje slike preko dodatne flash memorije multimedijske kartice

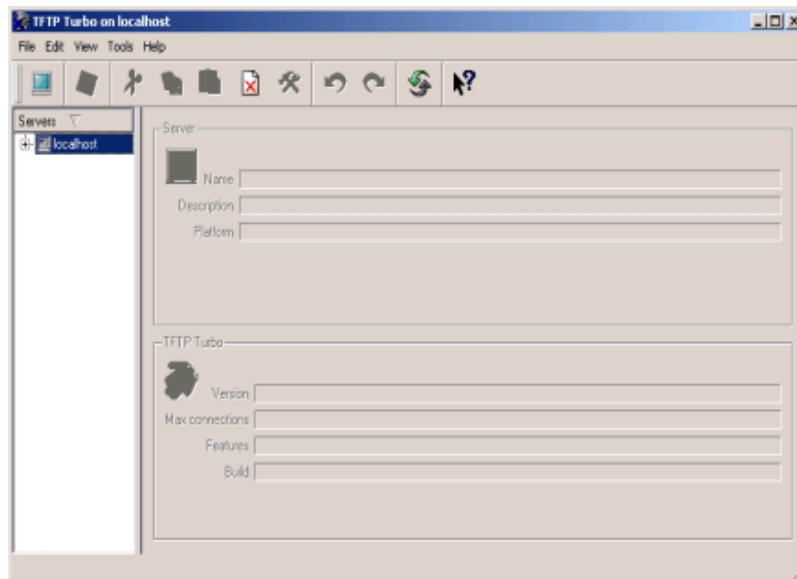
3.1. Učitavanje linux slike pomoću TFTP servera

Dakle preostaje, učitati komprimiranu sliku uCLinux-a, preko TFTP servera na pločicu. Nužan preduvjet za to je postavljanje TFTP servera, koji će detaljno biti opisan u sljedećem poglavlju **Postavljanje TFTP servera**.

3.1.1. Postavljanje TFTP servera

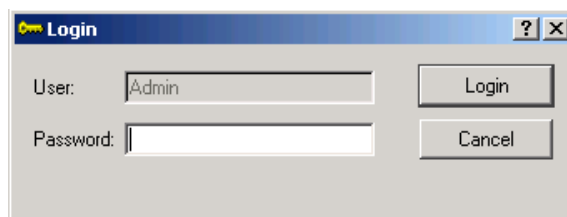
U ovom poglavlju objasnit će se kako postaviti TFTP server, preko programa *TFTP Turbo*, koji se pokreće na Windowsima.

Za početak potrebno je instalirati program i pokrenuti ga. Praćenjem instrukcija za instalirati program, odbire se potpuna instalacija. Sada kada je instalacija gotova potrebno je pokrenuti TFTP Turbo, čiji glavni prozor bi trebao izgledati kao na **slici 12**.



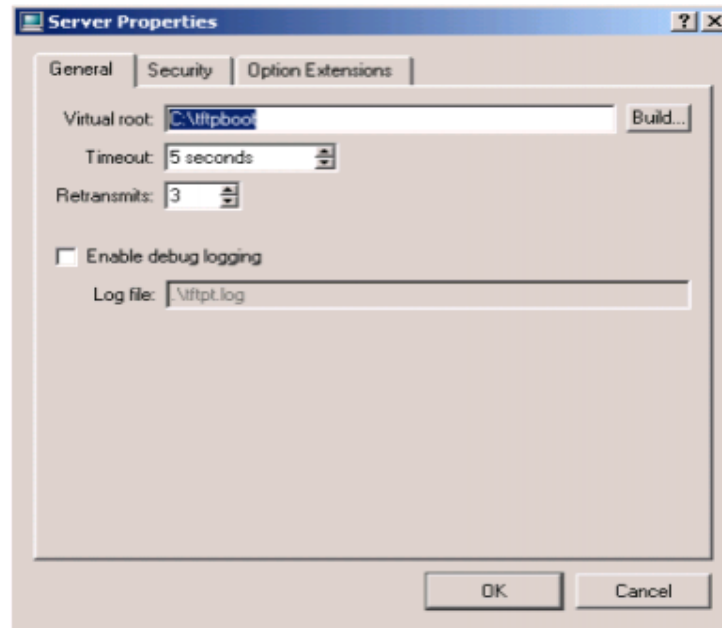
Slika 12 - TFTP Turbo (glavni prozor)

Dvostrukim klikom na *localhost* u stupcu *Server* pojavljuje se sljedeći prozor (Slika 12).



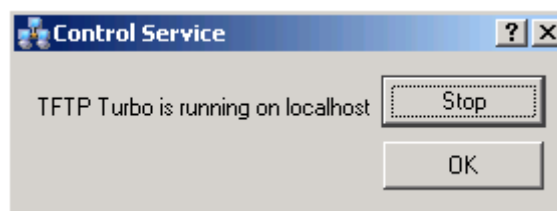
Slika 13 - TFTP Turbo (Login)

Polje *Password* ostaje prazno, te se samo odabire opcije *Login*. Sljedeći korak je odabrati u izborniku *Edit* polje *Properties*, dok *localhost* i dalje ostaje označen. Otvara se prozor *Server Properties* koji izgleda kao na **slici 14**.



Slika 14 - TFTP Turbo (Server Properties)

U polju Virtual root upisuje se staza do direktorija koji će se koristiti kao glavni direktorij, gdje će se smjestiti slika uCLinux-a. Kako bi provjerili radi li TFTP server, u izborniku *Tools* odabire se *Control Service*. Ukoliko poruka kaže "*TFTP Turbo is running on localhost*" samo se odabere opcija OK, a u koliko poruka kaže "*TFTP Turbo is not running on localhost*" odabire se Start pa OK (**Slika 15**).



Slika 15 - TFTP Turbo (Control Service)

3.1.2. Učitavanje slike uCLinux-a

Nakon što se TFTP konekcija, uspješno uspostavila, vrijeme je za prebacivanje slike uCLinux-a na pločicu.

Za početak potrebno je konfigurirati umrežavanje u U-Bootu pomoću sljedećih naredbi:

```
bfin> dhcp
Using MAC Address 00:E0:22:FE:47:7B
BOOTP broadcast 1
DHCP client bound to address 192.168.1.99
bfin> setenv serverip 192.168.1.1
bfin> saveenv
```

Naredbom *setenv* postavljaju se varijable okruženja (*serverip* i *ipadress*), a naredba *saveenv* se koristi kako bi se varijable spremile nakon reseta. Drugi način za postavljanje ovih varijabla je:

```
bfin> setenv serverip 192.168.1.1
bfin> setenv ipaddr 192.168.1.99
```

Sada preostaje samo preuzeti sliku uCLinux-a sa glavnog direktorija koji je ranije odabran (**Postavljanje TFTP servera**), i kopirati ga u memoriju namjesto koje želimo.

```
bfin> tftp 0x1000000 uImage
bfin> bootm 0x1000000
```

Kako je flash memorija premala za U-Boot i za sliku uCLinux-a (4 MB flash-a), slika se stavlja u SDRAM, te se tako nakon svakog reseta ove naredbe moraju ponoviti kako bi se opet sa TFTP servera preuzela slika. Umjesto stalnog samostalnog utipkavanja moguće je i sva automatizirati tako da se odmah nakon reseta traži slika sa TFTP-a te automatski pokreće:

```
bfin> setenv tftp1 "tftp 0x1000000 uImage ; bootm 0x1000000"  
bfin> saveenv
```

Pomoću naredbe *setenv* i *saveenv*, u varijablu *tftp1* spremili smo postupak spremanja slike u memoriju, te *bootanja* nakon toga.

Varijabla se pokreće naredbom *run*, a automatiziranje se postavljanjem *auto boot* varijable.

```
bfin> run tftp1  
bfin> setenv bootcmd run tftp1
```

3.2. Učitavanje linux slike pomoću *Multimedia* kartice

Kao što je već ranije navedeno u flash memoriju ne stane U-Boot i slika uCLinux-a. Jedan od načina učitavanja slike linux-a je taj da se uređaj poveže preko TFTP servera, no to rješenje nije najpraktičnije. Drugi način učitavanja komprimirane slike, je taj da se flash memorija proširi pomoću Multimedia kartice (Multimedia card, MMC). Na taj način U-Boot bi se direktno spremio u flash na pločici, a komprimirana slika uCLinux-a spremila bi se na MMC.

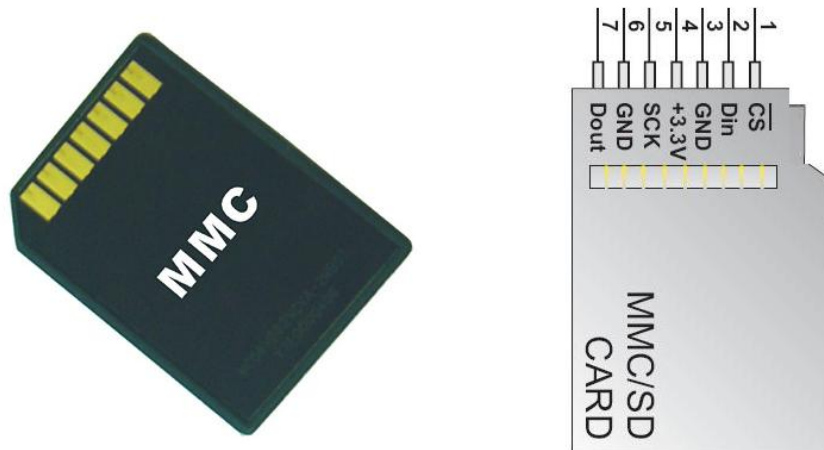
3.2.1. Spajanje kartice na pločicu

Multimedijska kartica spojit će se na pločicu preko serijskog perifernog sučelja (SPI). Kartica se ubacuje u adapter te se preko pinova adaptera koji su spojeni na pinove kartice, povezuje sa ADSP BF537 EZ-KIT Lite razvojnim modulom. Kako bi se kartica točno spojila potrebno je proučiti SPI *pinout*, te *pinout* kartice.

MMC sadrži 7 pinova a njihove uloge i položaj na kartici prikazani su **tabelom 3** te **slikom 16**.

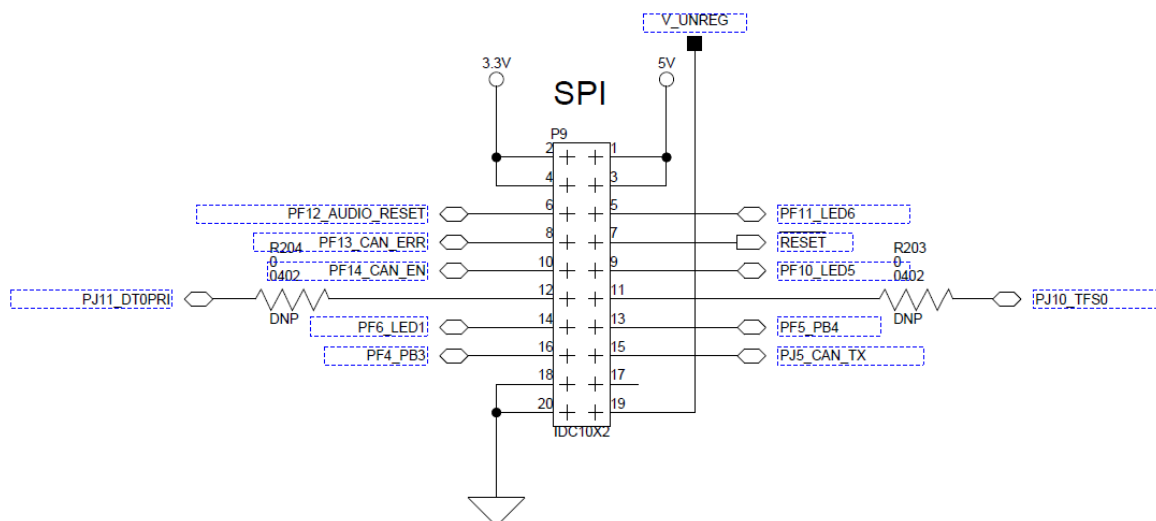
Tabela 3 - MMC pinout

Pin	Ime pina	Funkcija signala
1	CS*	Chip select
2	SI	Serijski ulaz
3	GND	Uzemljenje
4	Vcc	Napajanje
5	SCK	Serijski takt
6	NC	Bez spajanja
7	SO	Serijski izlaz



Slika 16 - MMC

SPI sučelje pločice sadrži 20 pinova, čija je shema prikazana na **slici 17**.



Slika 17 - SPI pinout

Procesor ukupno ima 48 ulazno/izlaznih signala opće namjene (GPIO), koji su rašireni uzduž tri skupine priključaka (PF, PG i PH). Pinovi su multifunkcionalni i ovise o procesorskim postavkama. U **tabeli 4** bit će navedeni samo oni pinovi koji su potrebni za spajanje sa multimedijском karticom.

Tabela 4 - SPI pinovi za MMC

Procesorski pin	Ostale funkcije	EZ-KIT Lite funkcija
PF 5	TMR4/SPI_SSEL5	Push button (SW10)
PF 11	SPI_MOSI	LED (LED6)
PF 12	SPI_MISO	Audio reset
PF 13	SPI_SCK	CAN ERR

Dakle, sada kada su prikazani pinout-i za SPI pločice i za MMC, potrebno ih je spojiti. Pin 1 multimedijske kartice, spaja se na PF 5, koji je 13. pin SPI-a. Pin 2, povezuje se s pinom PF 11, (*Master Output Slave Input*, MOSI) koji je 5. priključak po redu na serijskom sučelju. Pinovi 3 i 4 su pinovi napajanja te se spajaju na uzemljenje te na Vcc, tj. pinove 18-20 (*Ground*) i 2-4 (Vcc). Pin 5 spaja se PF 13 (SPI_SCK) tj. pin pod rednim brojem 8 na SP sučelju. Pin broj 6 se ne dira, dok na pin 7 spajamo *Master Input Slave Output* (PF 12, MISO), koji je pin broj 6. Ovime bi kartica trebala biti spremna za upotrebu.

Provjera uspješnog spajanja kartice na ADSP BF537 EZ-KIT Lite provest će se tako da spojimo uređaj na Hyper terminal te upišemo naredbu *mmcinit*. U slučaju dobrog spajanje dobivaju se sljedeće poruke na terminalu:

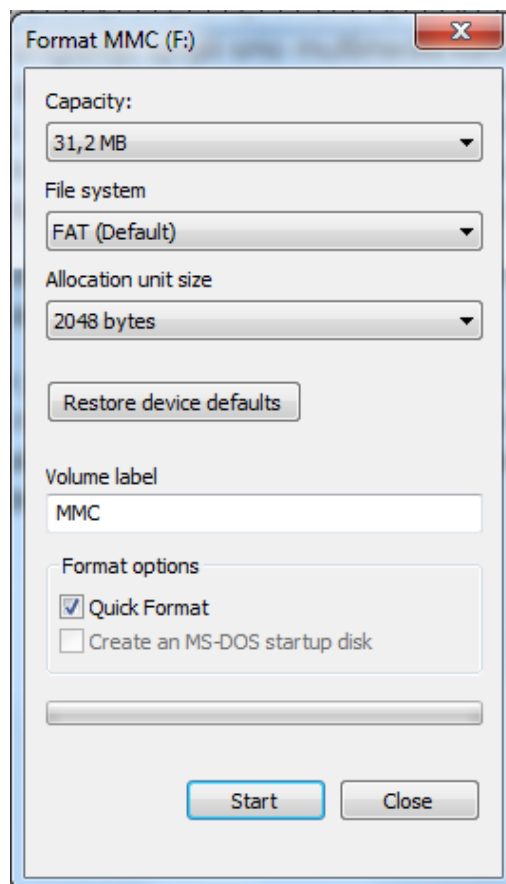
```
bfin> mmcinit
CID information:
Manufacturer ID:      03
OEM/Application ID:  5344
Product name:        SD256
Product Revision:    8.0
Product Serial Number: 544862406
Manufacturing Date:  07/08
SD Card detected (RCA 32769)
CSD information:
CSD structure version: 1.0
Card command classes: 5f5
Max trans speed: 25MHz
Read block length:   512
Write block length:  512
Card capacity:       252968960 bytes
```

3.2.2. Učitavanje slike uLinux-a

U prethodnom poglavlju spojili smo multimedia karticu na uređaj te ju je on pročitao. Sada je potrebno na karticu staviti komprimiranu sliku uLinux-a i učitati je na platformu. Da bi mogli koristiti karticu na ADSP BF537 EZ-KIT Lite razvojnom modulu, kartica će biti formatirana a File System na njoj bit će FAT16. Tek nakon formatiranja slika Linux-a bit će prebačena na karticu.

Formatiranje kartice može se obaviti i na Linux-u i na Windows-ima, a ovdje će biti opisan postupak formatiranja na Windows-ima.

Formatiranje je jednostavno, te sve što je potrebno napraviti je desnim klikom na karticu kliknuti na Format. Tada se pojavljuje prozor prikazan na slici 18. Postavke se ostavljaju kakve jesu, dok se samo za File System odabire FAT (u većini slučajeva to je određeno po defaultu), te se opcija Quick Format može isključiti. Klikom na Start obaviti će se formatiranje.



Slika 18 - Formatiranje kartice

Sljedeći korak je stavljanje slike uCLinux-a na karticu, umetanje kartice u adapter te učitavanje slike na pločicu.

Postupak započinjemo naredbom *fatinfo* koja će nam ispisati podatke o FAT-u uređaja spojenog na serijsko sučelje (*File allocation table*).

```
bfin> help fatinfo
fatinfo <interface> <dev[:part]>
    - print information about filesystem from 'dev' on 'interface'
bfin> fatinfo mmc
Interface: MMC
Device 0: Vendor: Man 035344 Snr 2079f0c6 Rev: 8 0 Prod: SD256
Type: Hard Disk
Capacity: 241.2 MB = 0.2 GB (494080 x 512)
Partition 1: Filesystem: FAT16 "
```

Naredbom *fatls* izlistavaju se sve datoteke i direktoriji na kartici.

```
bfin> help fatls
fatls <interface> <dev[:part]> [directory]
    - list files from 'dev' on 'interface' in a 'directory'
bfin> fatls mmc 0:1
6349262 uimage

1 file(s), 0 dir(s)
```

Slijedi učitavanje komprimirane slike uCLinux-a na memorijsku adresu 0x1000000, s koje će U-Boot učitavati operacijski sustav.

```
bfin> help fatload
fatload <interface> <dev[:part]> <addr> <filename> [bytes]
    - load binary file 'filename' from 'dev' on 'interface'
    to address 'addr' from dos filesystem
bfin> fatload mmc 0:1 0x1000000 /uimage
reading /uimage

6349262 bytes read
```

Konačno pokrećemo naredbu *bootm* kako bi pokrenu uCLinux.

```
bfin> bootm
## Booting kernel from Legacy Image at 01000000 ...
Image Name:   Linux-2.6.28.10-ADI-2009R1
Created:      2009-06-10   6:06:18 UTC
Image Type:   Blackfin Linux Kernel Image (gzip compressed)
Data Size:    6349198 Bytes =  6.1 MB
Load Address: 00001000
Entry Point:  0027d49c
Verifying Checksum ... OK
Uncompressing Kernel Image ... OK
Starting Kernel at = 0027d49c
```

4. Pokretanje jednostavne linux aplikacije

Zadnja velika cjelina u ovom završnom radu je pokretanje Linux aplikacije. Linux aplikaciju učitati će se preko *ftp servera*. Aplikacija će biti pisana u programskom jeziku C, ali će se za potrebe pokretanja sa pločice prevesti u oblik pogodan za operacijski sustav uCLinux. Svi navedeni koraci objasniti će se u sljedećim poglavljima.

4.1. Pisanje i kompajliranje aplikacije

Aplikacija koja će se pokretati biti će jako jednostavna, a cilj će joj biti ispisati poruku na ekran. Kao i u svakom primjeru za nekakav ispis ili testiranje rada sustava, poruka će glasiti "Hello world!". Aplikacija izgleda ovako :

```
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

Program je pisan u programskom jeziku C te će se spremirati pod nazivom "hello.c" gdje nastavak .c označava da se radi o C aplikaciji.

Sljedeći korak je kompajliranje aplikacije pomoću Blackfin GNU toolchain-a, tj. prevođenje programa u oblik razumljiv uCLinux-u. Za početak potrebno je osigurati da se *bfin-uclinux-gcc* nalazi u trenutnom direktoriju, a ispitivanje ćemo provesti pomoću naredbe *bfin-uclinux-gcc --version*. Ukoliko je sve u redu pojaviti će se sljedeće:

```
rgetz@imhotep:~> bfin-uclinux-gcc --version
bfin-uclinux-gcc (ADI-trunk/svn-3343) 4.3.3
Copyright (C) 2008 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Ukoliko se pojavi sljedeća poruka potrebno je, pozicionirati se na pravu lokaciju:

```
rgetz@imhotep:~> bfin-uclinux-gcc --version
bash: bfin-uclinux-gcc: command not found
```

Nakon što je sve spremno, kreće se sa kompajliranjem programa pomoću naredbe *bfin-uclinux-gcc --version*.

```
rgetz@imhotep:~/src> bfin-uclinux-gcc -Wl,-elf2flt hello.c -o hello
```

Naredba će osim datoteke *hello* stvoriti i datoteku *hello.gdb* koja služi za ispravljanje pogrešaka tj. *debugiranje*.

4.2. Učitavanje i pokretanje aplikacije preko mreže

Mnogo je načina da se preko mreže učita aplikacija, no ovdje će se opisati postupak učitavanja preko tftp servera, kojeg smo koristili i kao varijantu za učitavanje U-Boot-a.

Prije mrežnog učitavanja aplikacije potrebno je podesiti mrežne postavke pločice. To se može napraviti ili ovako :

```
root> dhcpd &  
root> ifconfig eth0
```

gdje se pomoću naredbe dhcpd namješta varijabla eth0 ili ovako:

```
root> ifconfig eth0 192.168.0.1 up
```

gdje se ručno postavlja adresa domaćina preko kojeg ćemo prenijeti aplikaciju.

Kako bi mogli prenijeti aplikaciju na pločicu, jedan od zahtjeva je da se aplikacija hello prekopira u glavni direktorij programa TFTP Turbo o kojemu je već ranije bilo riječi. Nakon toga koristimo naredbu *tftp* :

```
root> tftp -g -r hello 192.168.0.1
```

Pomoću naredbe *chmod* mijenjamo postavke datoteke *hello*:

```
root:/tmp> chmod 777 hello
```

I konačno slijedi pokretanje aplikacije:

```
root:/tmp> ./hello  
Hello, World
```


Zaključak

U ovom radu analizirali su se različiti načini ugradnje operativnog sustava na pločicu ADSP-BF537 EZ-KIT Lite firme Analog Devices. Cijeli rad oblikovan je u tri glavne cjeline, a to su učitavanje U-Boot-a na pločicu, učitavanje slike uCLinux-a te ispitivanje rada operacijskog sustava pokretanjem jednostavne aplikacije.

Pokazalo se kako postoji nekoliko varijacija ugradnje operacijskog sustava, ovisno o domaćinu sa kojega se uzimala komprimirana slika, načinu spajanja pločice na domaćina, operacijskom sustavu domaćina te programskim alatima. Na početku je ulogu domaćina za prebacivanje U-Boot-a preuzelo računalo, koje je koristilo operacijski sustav Windows. Pločica se spajala sa računalom ili preko serijskog RS-232 kabla ili preko USB kabla, ovisno o programima preko kojih smo učitavali podatke u memoriju sustava. Prilikom postupka učitavanja same slike operacijskog sustava na pločicu, pokušala se, osim računala iskoristiti i multimedijaska kartica, gdje je ideja bila da se operacijski sustav automatski "diže" sa kartice. Jedini problem koji se mogao izvoditi točno na određen način je pisanje i "kompajliranje" aplikacije, ali se opet pri njezinom prebacivanju na pločicu moglo birati između više opcija.

Jedina prepreka koja je ostala nesavladana je učitavanje slike uCLinux-a preko multimedijske kartice. Razlog tomu je najvjerojatnije forma naredbi preko koje je bilo potrebno učitati podatke sa kartice na pločicu. Kako je sama pločica prepoznala karticu, forma naredbi nameće se kao jedini logični krivac za neuspjeh, no usprkos tomu kao jedna od alternativa, poslužio je tftp server preko kojeg su se podatci prebacili na pločicu.

Može se zaključiti kako je postupak ugradnje operacijskog sustava na razvojni sustav ADSP-BF537 EZ-KIT Lite, složeniji proces, pogotovo za osobe koje se prvi put susreću sa ovakvom vrstom pločice. Unatoč tome veliki broj mogućnosti koje pruža pločica, mnoštvo dokumentacije o pločici na internetu, te popularnost same pločice, čine postupak učitavanja operacijskog sustava na pločicu bitno jednostavnijim.

Literatura

1. Robin Getz: Blackfin Koop, ožujak 2010., *Linux Blackfin*, <http://blackfin.uclinux.org/gf/>, 17. Ožujka 2010.
2. Vapier: bootloaders:u-boot [Open Source @ Analog], studeni 2009., *[[bootloaders:u-boot]]*, <http://docs.blackfin.uclinux.org/doku.php?id=bootloaders:u-boot>, 17. ožujka 2010.
3. Vapier: uclinux_on_blackfin [Open Source @ Analog], ožujak 2010., *[[uclinux_on_blackfin]]*, http://docs.blackfin.uclinux.org/doku.php?id=uclinux_on_blackfin, 1. travnja 2010.
4. Vapier: downloading_to_the_target [Open Source @ Analog], ožujak 2010., *[[downloading_to_the_target]]*, http://docs.blackfin.uclinux.org/doku.php?id=uclinux_on_blackfin, 1. Travnja 2010.
5. Vapier: bootloaders:u-boot:booting_methods [Open Source @ Analog], rujan 2009., *[[bootloaders:u-boot:booting_methods]]*, http://docs.blackfin.uclinux.org/doku.php?id=bootloaders:u-boot:booting_methods, 5. travnja 2010.
6. Vapier: setting_up_a_ftp_server [Open Source @ Analog], kolovoz 2009., *[[setting_up_a_ftp_server]]*, http://docs.blackfin.uclinux.org/doku.php?id=setting_up_a_ftp_server, 27. travnja 2010.

Naslov, sažetak i ključne riječi

Naslov

Ugradnja operacijskog sustava uCLinux na DSP procesore porodice Blackfin

Ključne riječi

ADSP, blackfin, BF537, uCLinux, U-Boot, tftp, multimedijaska kartica, flash

Sažetak

Cilj ovoga rada bio je ugradnja operacijskog sustava uCLinux na pločicu ADSP-BF537 EZ-KIT Lite firme Analog Devices. Ugradnja se sastojala od tri dijela: učitavanja U-Boot-a, učitavanje uCLinux-a te pokretanje jednostavne aplikacije kojom bi provjerili rad operacijskog sustava. Postupak učitavanja U-Boot-a i uCLinux-a, su objašnjeni na dva načina, dok je pokretanje aplikacije realizirano na jedan način. Jedne od najčešćih metoda za komunikaciju i rad sa pločicom su preko tftp servera i serijske RS-232 veze. U ovom radu su se najčešće koristile upravo kombinacije tih dviju metoda.

Title, summary, keywords

Title

Installing the operating system uCLinux on the DSP procesors of the Blackfin family

Keywords

ADSP, blackfin, BF537, uCLinux, U-Boot, tftp, multimedia card, flash

Summary

The aim of this study was installation of the operating system uCLinux on ADSP-BF537 EZ-KIT Lite board from the Analog Devices company. Installation consisted of three parts: loading the U-Boot, loading the uCLinux and runing a simple application that would check the work of the operating system. The procedure of loading the U-Boot and uCLinux, were explained in two ways, while the running of application was realized in one way. One of the most common methods for communicating and working with the bords are by the TFTP server and serial RS-232 connection. In this work the combination of those two methods, was most commonly used .