BME-MIT

Beágyazott rendszerek fejlesztése laboratórium – DSP fejlesztési technológiák

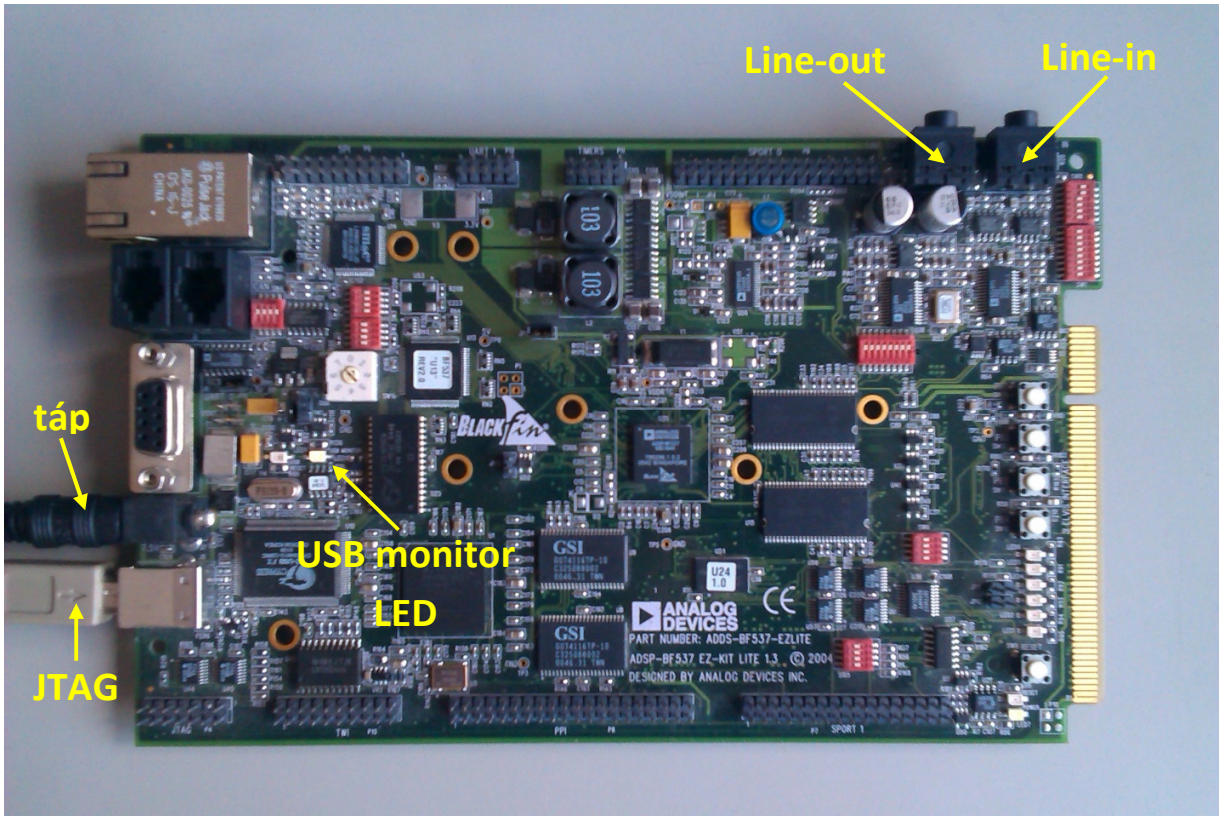
Bevezető mérési feladatok a fejlesztőeszközök megismeréséhez

Orosz György

2015

Fejlesztőkártya és fejlesztőkörnyezet előkészítése

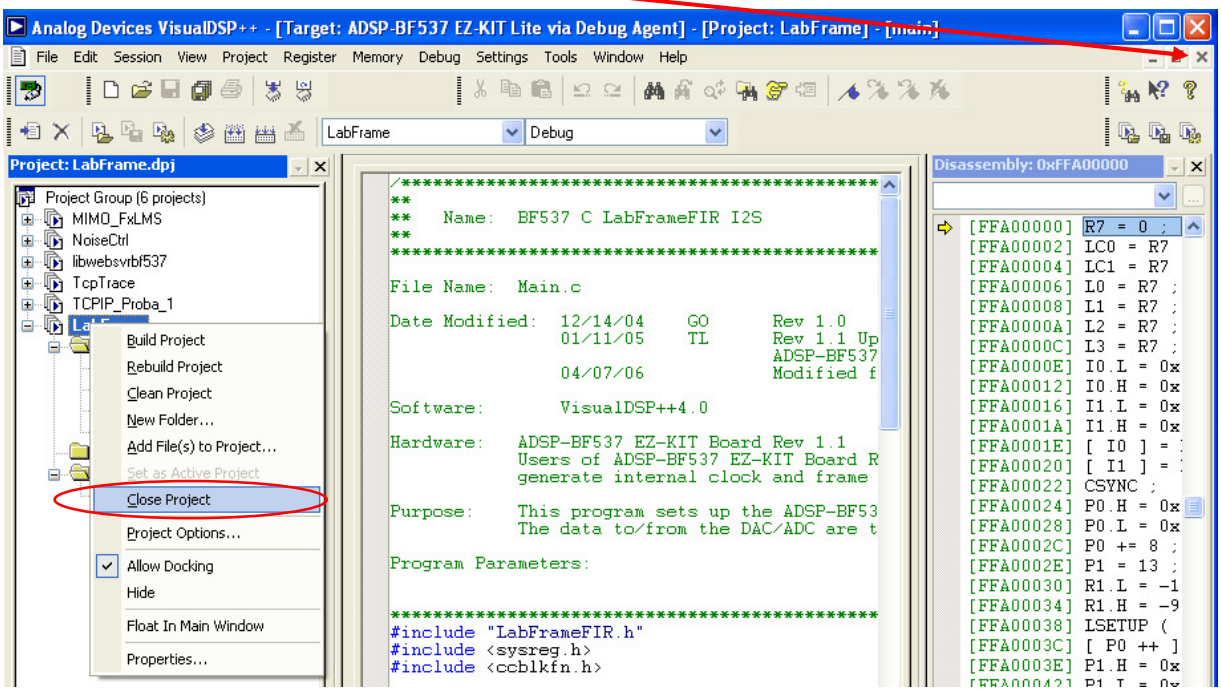
- Csatlakoztasd a DSP-kártyát a PC-hez. **Figyelem!** Először a tápcsatlakozót kell bedugni, és csak aztán az USB-s JTAG-et. Ez a sorrend a későbbiekben is érvényes! (például lefagy a program, és a kártyát áramtalanítani kell).
- Várd meg, amíg az „USB monitor” LED kigyullad! (felépül a kapcsolat a PC és a kártya között)



- Indítsd el a PC-n a fejlesztői környezetet: Start menü → Programs → Analog Devices → VisualDSP++ 5.0 → VisualDSP++ Environment

  Fontos, hogy az 5.0-ás verziót indítsuk el. Szánjunk időt arra, hogy kiválasszuk a programot, és ne az asztalon található első kósza ikonra kattintsunk, ezzel a későbbiekben sok kellemetlenségtől kímélhetjük meg magunkat!

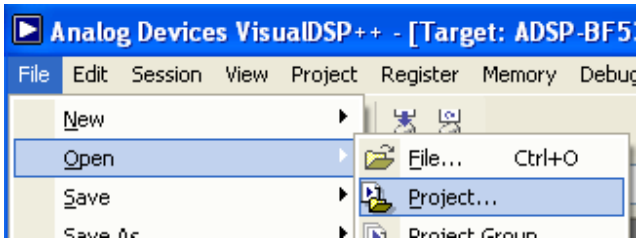
- Zárjuk be az összes megnyitott projektet és ablakot! (később sok félreértés forrásai lehetnek)
    - Projekt bezárása: a bal oldali Project ablakban jobb gombbal a bezárandó projektre kattintunk, ott a „Close Project” menüpontot választjuk (lásd ábra).
    - Ablakok bezárása: a jobb fölső sarokban található „x” gombbal zárhatjuk be az ablakokat.

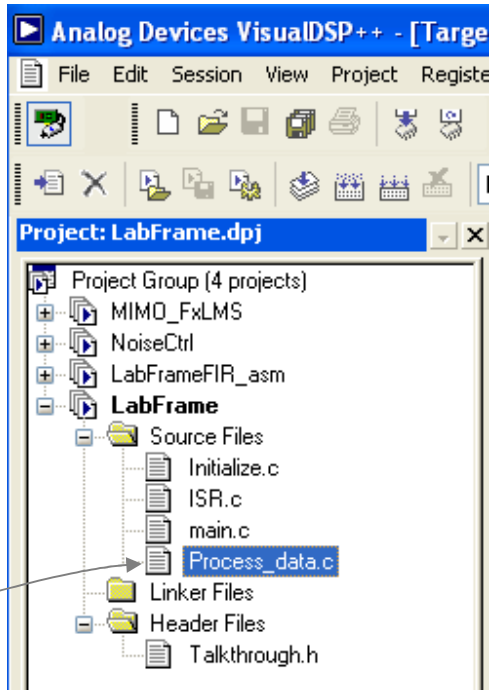


- Ellenőrizzük, hogy a kártyát felismerte-e a fejlesztőkörnyezet: a címsorban megjelenik-e az „ADSP-BF537 EZ-KIT Lite via Debug Agent” felirat (lásd fent). Ha nem, akkor a Session menü→Select session menüpontjából válasszuk ki.
- Hozzunk létre egy saját könyvtárat (lehetőleg, ami jól azonosítja a csoportot – a pl. Csoprot10 és hasonlók nem jó)!
- Töltsük le a tárgy honlapjáról a mintaprojektet! Útvonal: tanszéki honlap → Oktatás → Tantárgyak → Mesterképzés (MSC) → Rendszerarchitektúrák laboratórium → Jegyzetek → „Minta project az első méréshez - zip”. Csomagold ki a fájlt!
- Analóg jelek csatlakoztatása jack – BNC kábelekkel (Line-in és Line-out csatlakozók)
    - **Figyelem:**
        - A kártyán lévő feliratokat alaposan figyeljük meg (Line-in/Line-out), nehogy két kimenetet összekössünk!
        - A jelgenerátoron az egyik -20 dB gomb legyen benyomva, ellenkező esetben túl nagy jelet adhatunk a kártyára! (jelgenerátor: max. 20 V_{pp}, DSP kártya megengedett bemeneti jelszintje: 3 V_{pp})
    - A „Line out” kimenetet csatlakoztasd az oszcilloszkóphoz (mindkét csatornát).
    - A „Line in” bemenetet csatlakoztasd a jelgenerátorhoz (egyelőre mindegy melyiket, majd szükség esetén megcseréljük). Állíts be szinusz jelet!

Egyszerű projekt megnyitása

- Nyisd meg a kicsomagolt könyvtárban a LabFrame.dpj projektet! Projekt megnyitása: File menü → Open → Project…, itt ki kell választani a saját könyvtárban a megfelelő fájlt!





- A Project ablakban található LabFrame projekt Source Files könyvtárában található a Process_data.c fájl, amely a jelfeldolgozó eljárást tartalmazza:
  - void Process_Data(void) függvény minden új mintavétel során meghívásra kerül.
  - Az iChannel0RightIn és iChannel0LeftIn változók tartalmazzák a mintavételezett adatokat.
  - Az iChannel0RightOut és iChannel0LeftOut tartalmazzák a kiadott adatokat.
- Fordítsuk le a projektet! A fordítás során a lefordított kód be is töltődik a kártyára. Fordítási lehetőségek:
  - F7 billentyűvel *vagy*
  - Projekt menü → Build (Rebuild all) *vagy*
  - a  gombra kattintva.
- Futtassuk a projektet! (a kártyára töltött kód automatikusan Halt állapotba kerül fordítás után) Program futtatása:
  - F5 *vagy*
  - Debug menü → Run *vagy*
  - a  gombra kattintva.

  Futás leállítása:
  - Shift + F5 *vagy*
  - Debug menü → Halt *vagy*

- a  gombra kattintva.

- A projekt egyszerűen a jobb csatornán található jelet kiteszi a jobb és bal csatornára, ezzel tesztelhetjük a rendszert, és meghatározhatjuk, hogy melyik a jobb oldali bemenet. Amennyiben a program futtatását követően nem jelenik meg semmi az oszcilloszkóp képernyőjén, cseréljük meg a jelgenerátorra csatlakoztatott bemeneteket. Jegyezd meg, hogy melyik a jobb oldali bemenet!

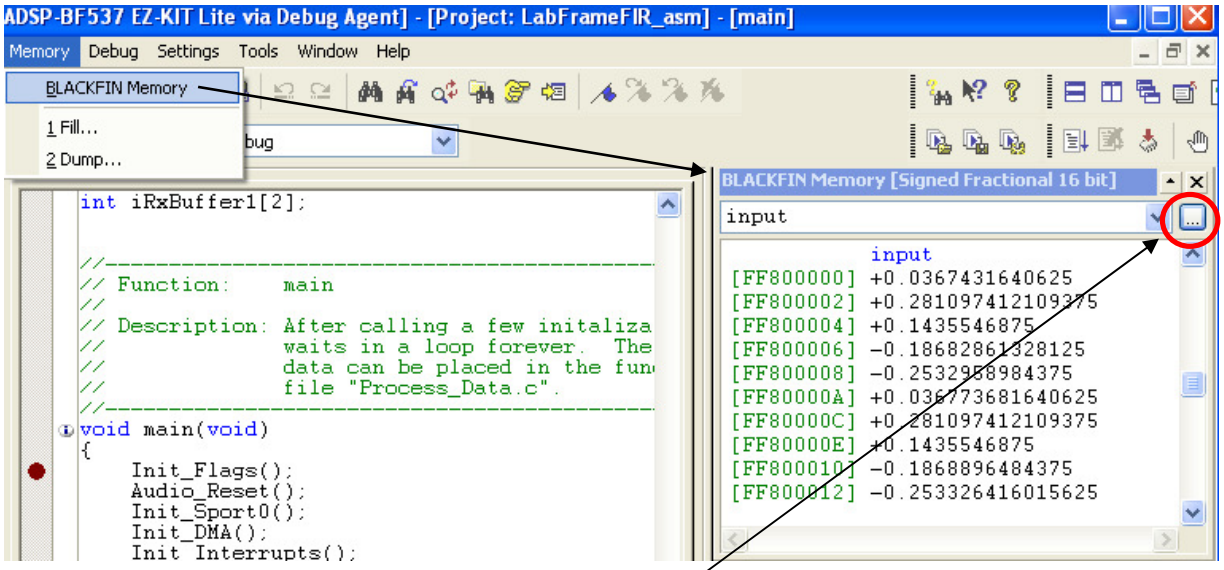
FIR szűrő assembly rutinnal

- Nyisd meg a LabFrameFIR_3.dpj projektet! A rendszer egy FIR szűrőt valósít meg: a jobb oldali csatornán beérkező jelet szűri. A bal kimeneti csatornán a szűretlen, jobb kimeneti csatornán a szűrt jel látható.
- Tanulmányozd és értsd meg a kódot! Megjegyzések:
    - A iChannel0RightIn >> 8 és out <<8 parancsokban a 8 bites shiftelés azért van, mert a DSP-n 16 bittel dolgozunk, míg az AD/DA átalakítók 24 bitesek, ezt később sem feledd!
    - A szűrést a conv_asm függvény végzi.
- Fordítsd le a projektet és futtasd!
- Mérd meg a szűrő jellegzetes töréspontí frekvenciáit jelgenerátor és oszcilloszkóp segítségével, és vázold fel az amplitúdómenetet! Milyen típusú szűrőről van szó? Hasonlítsd össze az ideális szűrővel, amely a DSP-n implementálásra került: MATLAB-ban az fdatool parancs hatására megnyíló ablakban a bandpass.fda fájlt kell megnyitni.

Debug funkciók megismerése

Változók írása és olvasása a memóriában.

- A Memory menüpont BLACKFIN Memory menüpontra kattintva megjelenik egy „BLACKFIN Memory” feliratú ablak.

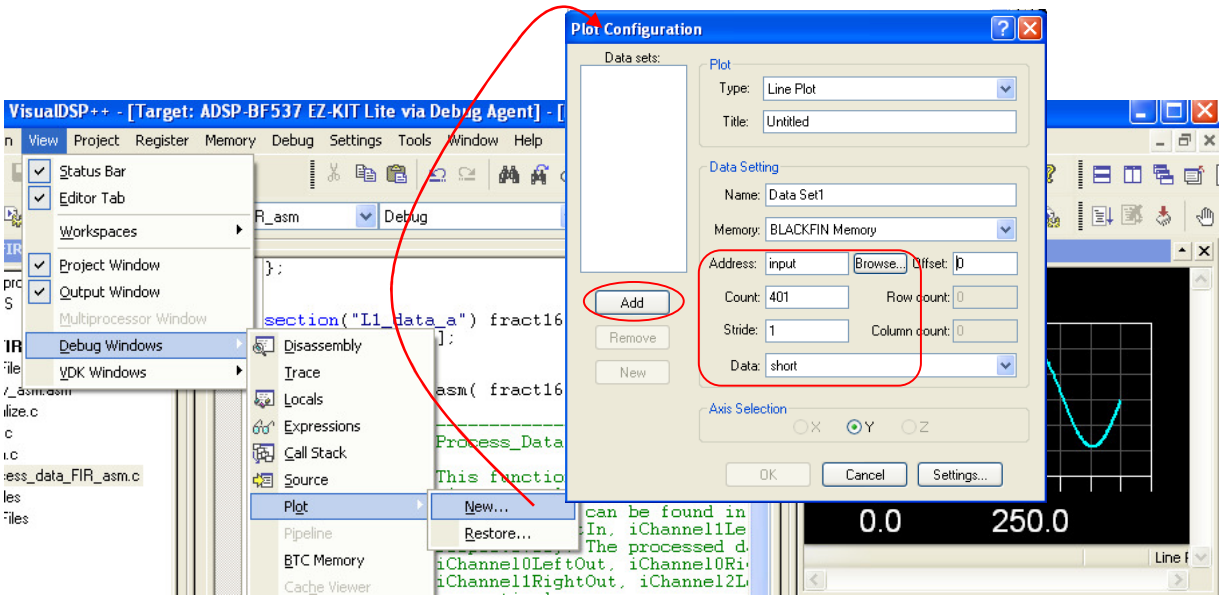


- Az ábrán pirossal bekeretezett gombra kattintva kiválaszthatjuk, hogy melyik változót szeretnénk megtekinteni (válasszuk ki pl. az input tömböt).
- Jobb gombbal valamelyik adatra kattintva a Select Format menüpontban kiválaszthatjuk a megjelenítés formátumát (16 bites fix pontos tört számok esetén „Signed Fractional 16 bit”-et kell választani, egyébiránt igazodni kell a megjelenített változó formátumához).

- Egy adatra kétszer kattintva módosítható az adat. Az új értéket beírva, ENTER megnyomásával érvényesíthetjük, ekkor a DSP memóriájában is átírásra kerül.
- A funkciókat csak a DSP Halt állapotában lehet használni (a program futását le kell állítani).
- A program egyes fázisaiban a változókat úgy érdemes vizsgálni, hogy brakepointokat helyezünk el a kódban (az adott kódsor mellett balra a szürke sávra kétszer kattintva). A lépésenkénti végrehajtás általában elég lassú, használata nem javasolt. A fejlesztői környezet lehetővé teszi a változók megváltozására történő programmegszakítást, ezt nem használjuk.

Tömbök grafikus megjelenítése.

- A View menü → Debug windows → Plot → New … almenükön keresztül megjelenő ablakkal határozhatjuk meg a kirajzolandó tömb paramétereit, lásd alább.



- A Browse (ablak közepén) gombbal kiválaszthatjuk a változót. Legyen pl. az input változó.
- A Count mezőbe beírhatjuk a megjelenítendő adatok számát. Legyen ez 401!
- A stride a lépés méretét adja meg, ez legyen 1.
- A Data legördülő listában megadhatjuk a formátumot. Ez legyen short!
- A bal oldali Add gombbal aktiválhatjuk a beállított adatokat, és az OK gomb hatására megjelenik az ablak.
- A funkciókat csak a DSP Halt állapotában lehet használni (a program futását le kell állítani).
- Állítsunk be olyan mintavételi frekvenciát, ahol a mintavételezés koherens (teljes periódus van az input bufferben) és ahol nem. Honnan lehet látni a buffer cirkularitását?

- Írjunk át egy adatot a memóriában az input tömbben, és nézzük meg a grafikus megjelenítőben, hogy valóban megtörtént-e a változás!
- Vizsgáljuk meg a szűrőegyütthatókat (coefs változó)!

Futásiidő-analízis

- A Tools menü → Statistical Profiling → New Profile almenüben nyissunk meg egy új futási idő statisztikát!
- Indítsuk el a program futását, és elemezzük az egyes funkciók futási idejét! Jegyezzük fel a conv_asm függvény százalékos futási idejét!
- Vagy a bemenő adatokat (input) vagy a szűrőegyütthatókat (coefs) tedd át egy másik section-be (`section("L1_data_a")` ból `section("L1_data_b")`-be vagy fordítva).
- Nézd meg újra a százalékos futási időket! Mekkora a különbség? (érdemes a statisztikát törölni az újbóli futtatás előtt: jobb gomb a statisztikára és „Clear profile”)

FIR szűrő megvalósítása C nyelven

- Nyisd meg a LabFrameFIR_2.dpj projektet!
- Értelmezd, hogy hogyan valósítjuk meg C nyelvben a cirkuláris buffert! Milyen függvényeket használunk összeadásra és szorzásra? Miért?
- Mérd meg a jelfeldolgozó rutin százalékos futási idejét!
- A Project menü → „Project Options …” almenüjét válasszuk ki. A megnyíló ablakban a Compile → General beállításoknál az „Additional options:” szövegmezőből töröljük ki a -force-circbuf opciót!
- Fordítsuk le újra a projektet és nézzük meg a futási idő statisztikát! Hogyan változott a futási idő és miért (milyen új függvények jelentek meg)?
- Érdemes visszaállítani a -force-circbuf opciót.
- Végezz el néhány összeadást és szorzást a normál * és + operátorokkal, illetve a beépített könyvtári függvényekkel! Többféle szituációra is teszteld és összegezd a tapasztalatokat!