



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Méréstechnika és Információs Rendszerek Tanszék

DSPController

Gyors kezdési útmutató

ÁLTALÁNOS INTERFÉSZ ILLESZTÉSE JELFELDOLGOZÓ
PROCESSZORHOZ

v1.0

Készítette

Simon Tibor

Konzulens

dr. Orosz György

2014. január 21.

Tartalomjegyzék

1. Bevezetés	2
2. DSPController csomag tartalma	3
2.1. Dokumentációk	3
2.1.1. Teljes dokumentáció [2]	3
2.1.2. Gyors kezdési útmutató	4
2.2. Példa projektek	4
2.2.1. Bare Minimum App	4
2.2.2. Demo App	4
2.3. Források	4
2.3.1. API	4
2.3.2. Firmware	5
2.3.3. Hardware	5
3. Beüzemelés	6
3.1. Hardver előkészítése	6
3.2. Példa projektek használata	7
4. VisualDSP++ környezet sajátosságai	8
4.1. Gyakran előforduló hibák	8
5. Saját projekt írása	9
Irodalomjegyzék	10

1. fejezet

Bevezetés

Jelen dokumentum célja, hogy segítse a felhasználót az első lépések megtételében a DSP-Controller rendszer használatához.

Ismertetésre kerül a DSPController csomag tartalma, a rendszer beüzemelése lépésről-lépésre, a fejlesztői környezet sajátosságai, valamint a sikeres beüzemelés utáni teendők.

Amennyiben a dokumentum áttanulmányozása után bármilyen további kérdés merülne fel, úgy emailben a *tiborsimon@tibor-simon.com* címen lehet tanácsot kérni.

2. fejezet

DSPController csomag tartalma

A DSPController rendszerhez magán a hardveren kívül még hét további digitális dokumentum tartozik.

DSPController csomag digitális tartalma		
Dokumentáció	Dokumentáció [2]	github.com/tiborsimon/DSPController-Bundle
	Kezdési útmutató	github.com/tiborsimon/DSPController-Bundle
Példaprojekt	Bare Minimum App	github.com/tiborsimon/DSPController-BareMinimumApp
	Demo App	github.com/tiborsimon/DSPController-DemoApp
Forrás	API	github.com/tiborsimon/DSPController-API
	Firmware	github.com/tiborsimon/DSPController-Firmware
	Hardware	github.com/tiborsimon/DSPController-Hardware

DSPController csomag hardverei	
Áramkör	DSPController interfész eszköz
Kiegészítő	DSPController tápkábel

2.1. Dokumentációk

2.1.1. Teljes dokumentáció [2]

A teljes dokumentáció tartalmazza azokat az ismereteket, amelyekre szükség van a DSPController rendszer teljeskörű használatához. A következő főbb fejezeteket tartalmazza:

Hoszt egység	Alapvető információk a SHARC DSP kártyáról.
DSPController interfész	DSPController hardveres áttekintés.
SPI protokoll	A hoszt és az interfész egység közötti kommunikáció leírása.
DSPController API	DSPController API teljes dokumentációja.
Példa projektek	Elérhető, fejlesztést segítő minta alkalmazások.

A rendszer alapvető használatához nem szükséges a teljes dokumentum elolvasása, de legalább az alábbi fejezetek tartalmával tisztában kell lenni:

- DSPController API
- Példa projektek

<https://github.com/tiborsimon/DSPController-Bundle>

2.1.2. Gyors kezdési útmutató

Az éppen olvasott dokumentum. Segítséget nyújt a fejlesztés megkezdésében.

<https://github.com/tiborsimon/DSPController-Bundle>

2.2. Példa projektek

A SHARC jelprocesszoron háttérben futó mechanizmusok elfedésére, a példa projektek egy magasabb szintű felületet mutatnak a felhasználó felé, amiben mindeössze három fájl szerkesztésével képesek jelfeldolgozási feladatok megoldására.

2.2.1. Bare Minimum App

A legalapvetőbb, működéshez minimálisan szükséges felhasználói kódot tartalmazó példa projekt. Ez lehet a kiindulás minden későbbi DSPControllert használó alkalmazásnak.

<https://github.com/tiborsimon/DSPController-BareMinimumApp>

2.2.2. Demo App

Szemléltető jellegű, jóval összetettebb alkalmazás, ami bemutatja a DSPControllert használó alkalmazások tipikus felépítését.

<https://github.com/tiborsimon/DSPController-DemoApp>

2.3. Források

A közölt forráskódok és gyártáshoz szükséges fájlok csupán szemléltető jellegűek. Használatukra, és módosításukra a legritkább esetben van csak szükség.

2.3.1. API

A DSPController API forráskódja tartalmazza a DSPController interfész eszközzel való kommunikációhoz szükséges mechanizmusok implementációját. Specifikusan az Analog Devices ADSP-21364 fejlesztőkártyához van portolva, más eszközökön használva módosítani kell a környezetspecifikus kódrészleteket. Új projektek kezdéséhez a mintaprojektek tartalmaznak minden szükséges összetevőt, nem kell törődni a DSPController API kézzel való hozzáadásával.

<https://github.com/tiborsimon/DSPController-API>

2.3.2. Firmware

A DSPControlleren futó firmware kódjának a használata és a DSPController interfész eszköz újraprogramozása is rendkívül ritka. A forráskód nyilvános közlése szemléltető jellegűnek tekinthető.

<https://github.com/tiborsimon/DSPController-Firmware>

2.3.3. Hardware

A DSPController interfész egység gyártásához szükséges összes fájlt tartalmazó csomag.

<https://github.com/tiborsimon/DSPController-Hardware>

3. fejezet

Beüzemelés

3.1. Hardver előkészítése

A programozás megkezdése előtt össze kell építeni a hardvert.

Szükséges eszközök

- Analog Devices ADSP-21364 SHARC fejlesztőkártya
- DSPController interfész eszköz
- DSPController tápkábel
- Tápegység adapter
- USB kábel (A és B csatlakozókkal)
- PC (VisualDSP++ 4.5 fejlesztőkörnyezettel)

Hardver összeállítása és beüzemelése

1	Csatlakoztassuk a DSPController tápkábelét az interfész egységhez, ha még nincs rádugva a fehér csatlakozón keresztül (ha már rajta van, nem is célszerű levenni).
2	A SHARC kártyáról szedjük le az esetleg rajta lévő kiegészítő kártyát.
3	Csatlakoztassuk a DSPControllert a SHARC kártyához, pontosan pozicionálva a rögzítő csavarokat és a 90 tűskés csatlakozót.
4	Rögzítsük a DSPControllert a rögzítőcsavarokra csavart távtartókkal.
5	Kössük be a DSPController tápkábelét a SHARC kártya tápcsatlakozójába.
6	Csatlakoztassuk a tápegység adaptert a DSPController tápkábeléhez, majd dugjuk be az adaptert egy fali csatlakozóba. Megjelenik az üdvözlő képernyő.
7	Ha még nincs bekapcsolva, kapcsoljuk be a számítógépet, amit használni fogunk a fejlesztéshez.
8	Az USB kábellel csatlakoztassuk a SHARC kártyát a számítógéphez.
9	Várjunk addig, amíg a kártyán az USB MONITOR LED-je világítani kezd.
10	Indítsuk el a VisualDSP++ 4.5 fejlesztőkörnyezetet.
11	A környezetnek fel kell ismernie a kártyát. Ha ez megtörtént, kezdődhet a fejlesztés.

3.2. Példa projektek használata

A rendelkezésre álló példa projektek egy biztos pontot nyújtanak a felhasználónak, amiből könnyebb az új projektek megkezdése.

A következő részekben a Bare Minimum App példa projektet fogjuk használni. Az eljárás menete csupán a projekt letöltési helyében különbözik a Demo App esetében.

Példaprojekt letöltése

1	Látogassunk el Bare Minimum App GitHub-os oldalára: https://github.com/tiborsimon/DSPController-BareMinimumApp
2	A jobb oldalon található oszlop alján kattintsunk a Download ZIP gombra.
3	A letöltött fájlt csomagoljuk ki tetszőleges helyre.

Példa projekt használata

1	Az előző blokkban ismertetett módon üzemeljük be a hardvert.
2	A megnyitott VisualDSP++ fejlesztőkörnyezetben nyissuk meg a letöltött példa projektet.
3	Fordítsuk le a projektet (<i>F7</i>).
4	Sikeres fordítás után a projekt rögtön feltöltődik a kártyára és megkezdődik a futását, majd rögtön meg is áll a main.c fájl belépési pontjánál. A futást kézzel kell újra beindítani (<i>F5</i>).
5	A futás megkezdése után a DSPController LCD képernyőjén az üdvözlő képernyőt felülírja a példaprojekt.

4. fejezet

VisualDSP++ környezet sajátosságai

Az Analog Devices fejlesztői környezete hajlamos nem várt módon viselkedni esetenként. Az ilyen hibák az első alkalommal megtévesztőek lehetnek.

4.1. Gyakran előforduló hibák

<i>Hiba</i>	<i>Lehetséges megoldás</i>
Nincs kapcsolat a hardverrel	Mentsük el a projektet, és lépünk ki a fejlesztői környezetből. Áramtalanítsuk a kártyát, majd újra csatlakoztassuk az adapteréhez. A kártyán lévő USB monitor LED folyamatos világítása után indítsuk be újra a VisualDSP++ környezetet.
Továbbra sincs kapcsolat a hardverrel	Lehetséges, hogy a <i>Debug Session</i> beállítás elállítódott szimulációra. Állítsuk vissza a csatlakoztatott kártyának megfelelő <i>Debug Session</i> -t.
A program futása folyamatosan megáll indokolatlanul egy helyen	A fejlesztői környezetet túl sokat használtuk. Megpróbálhatjuk kiszedni az összes hardveres és szoftveres breakpoint-ot, de ez az esetek nagyon kevés hányadában segít csak. Érdemes az első pontban ismertetett teljes újratezdés.
Debuggolás közben léptetésnél lefagy a környezet	Ne használjunk léptetést debuggolás alatt. Használjunk helyette több breakpoint-ot.

További információkért az Analog Devices használati útmutatóját érdemes használni [1].

5. fejezet

Saját projekt írása

Miután beüzemeltük a hardveres és a szoftveres egységeket, ideje, hogy hozzálassunk a saját feladatunk megvalósításának. Ehhez érdemes elolvasni a *Teljes dokumentáció* 6. Példa projektek fejezetét, ami részletesen tárgyalja a projekt felépítését, és a módosítandó forrásfájlok leírását.

A fejlesztés során szükségünk lehet a *Teljes dokumentáció* 5. fejezetére, ami a DSPController API dokumentációja.

Módosítandó fájlok

INTERFACE.c	Felhasználói interfészt megvalósító program.
DSP.c	Jelfeldolgozási algoritmus.
GLUE.h	Kapcsolat az interfész és a DSP kód között.

Saját projekt írásának lépései

1	Kívánt funkciókat kiszolgáló jelfeldolgozási algoritmus tervezése.
2	Vezérlési paraméterek összegyűjtése.
3	Vezérlési paraméterek létrehozása globális <i>GLUE</i> változóként a GLUE.h fájlban.
4	Jelfeldolgozó algoritmus implementálása a DSP.c fájlban.
5	Kívánt funkciókat lefedő menürendszer tervezése.
6	Megtervezett menürendszer állapotgépbe foglalása.
7	Paraméterek előállításának tervezése az interfészen.
8	Állapotgép impelementálása az INTERFACE.c fájlban.
9	<i>GLUE</i> változó értékadásának implementálása az INTERFACE.c fájlban.
10	Alkalmazás tesztelése, esetleges hibák javítása.

Irodalomjegyzék

- [1] Analog Devices. *VisualDSP++ 5.0 Users Guide*. AD, August 2007. URL: http://www.analog.com/static/imported-files/software_manuals/719705850_ug.pdf.
- [2] Simon Tibor. *DSPController Teljes Dokumentáció*. BME MIT, January 2014. URL: <https://github.com/tiborsimon/DSPController-Bundle>.