BF609 EVSK FinBoard
Dual Core SPI File Creation and Programming Instructions

26Mar13

I. Environment

1. CCES Version 1.0.1.1
2. Windows XP, Vista, 7 OS
3. Dual core application, for this example it is .DotCountApp
4. HDMI capable monitor
5. FinBoard Rev. B HW setup
6. ADI EZ-Kit installation (or the bf609_w25q32bv_dpia.dxe driver).
7. Two batch files: Make_LDR.bat & Burn_DotSPI.bat

II. Setup

1. Install CCES Version 1.0.1.1 in the default path of: C:\Analog Devices\CrossCore Embedded Studio 1.0.1

2. Using the CCES import command, install the .DotCountApp into a workspace within CCES. Note: the .DotCountApp is a dual core application and therefore two projects will show up in CCES under Project Explorer. They are named .DotCountApp_Core0 and .DotCountApp_Core1

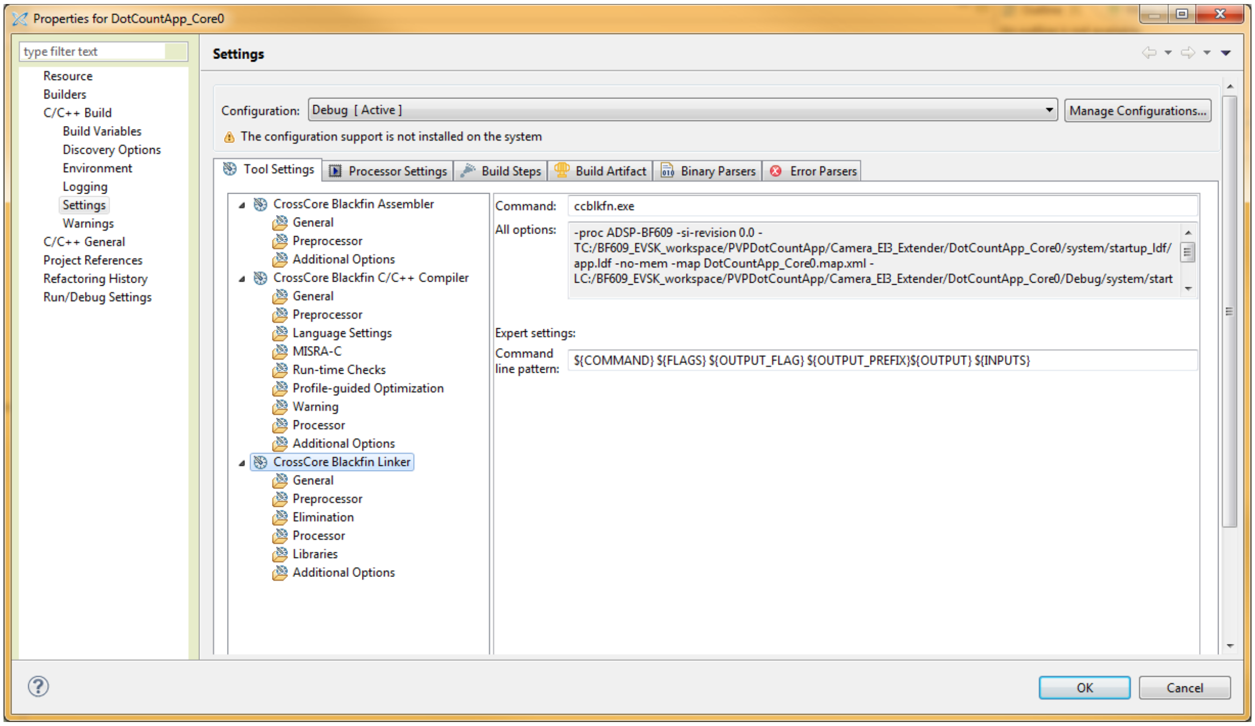
3. *NOTE: It is assumed the .DotCountApp application is functional within the CCES environment. To verify this attach the FinBoard via ICE-100B emulator and right click on the application. Select "Run As" option and the application should turn on the HB LEDs and perform video passthrough.*

III. Creating the .dxe files

1. Right click on the .DotCountApp_Core0 file project in the Project Explorer.

2. Select “Properties” on the drop down menu.
   i. Note: If you do not have the ICE-100B emulator attached to the FinBoard you will get the error “no processor found”. You can still create the .dxe files without the processor present.

3. Select C/C++ Build & then:
   i. Click – Settings tab
   ii. Check the Configuration Window – should read “debug [Active]”
   iii. Under Tool Settings tab select CrossCore Blackfin Linker > Processor: this should read “Allow indirect” and “Internal SRAM Only”
   iv. Click on “Build Artifact” tab. The Artifact Type should be “Executable” and the extension should be “dxe”.
   v. The default settings for Core1 should not need to be modified.
   vi. Click Apply or OK on the window.



Configuration: Debug [Active]

⚠ The configuration support is not installed on the system

Command: ccblkfn.exe

All options:

```
-proc ADSP-BF609 -si-revision 0.0 -
TC:/BF609_EVSK_workspace/PVPDotCountApp/Camera_EI3_Extender/DotCountApp_Core0/system/startup_ldf/
app.ldf -no-mem -map DotCountApp_Core0.map.xml -
LC:/BF609_EVSK_workspace/PVPDotCountApp/Camera_EI3_Extender/DotCountApp_Core0/Debug/system/start
```

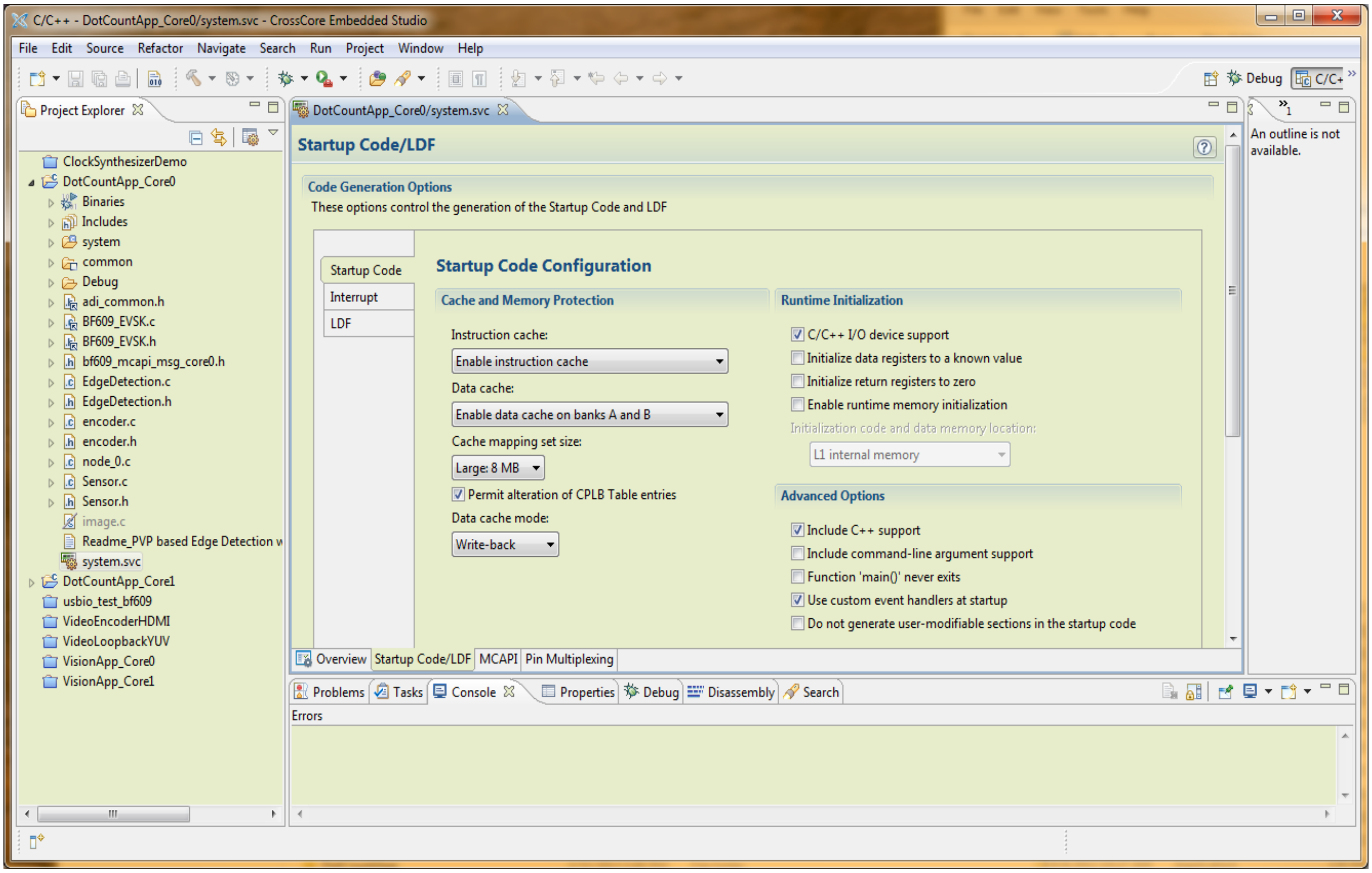
Expert settings:

Command line pattern: ${COMMAND} ${FLAGS} ${OUTPUT_FLAG} ${OUTPUT_PREFIX}${OUTPUT} ${INPUTS}

Figure 1 - Tool Settings Screen

4. In the Core0 Project Explorer window, double click the “system.svc” icon.

5. Select the “Startup Code/LDF” tab in the middle of the popup window, as below:



Startup Code/LDF

Code Generation Options

These options control the generation of the Startup Code and LDF

Startup Code Configuration

Cache and Memory Protection

Instruction cache: Enable instruction cache

Data cache: Enable data cache on banks A and B

Cache mapping set size: Large: 8 MB

☑ Permit alteration of CPLB Table entries

Data cache mode: Write-back

Runtime Initialization

☑ C/C++ I/O device support

☐ Initialize data registers to a known value

☐ Initialize return registers to zero

☐ Enable runtime memory initialization

Initialization code and data memory location: L1 internal memory

Advanced Options

☑ Include C++ support

☐ Include command-line argument support

☐ Function 'main()' never exits

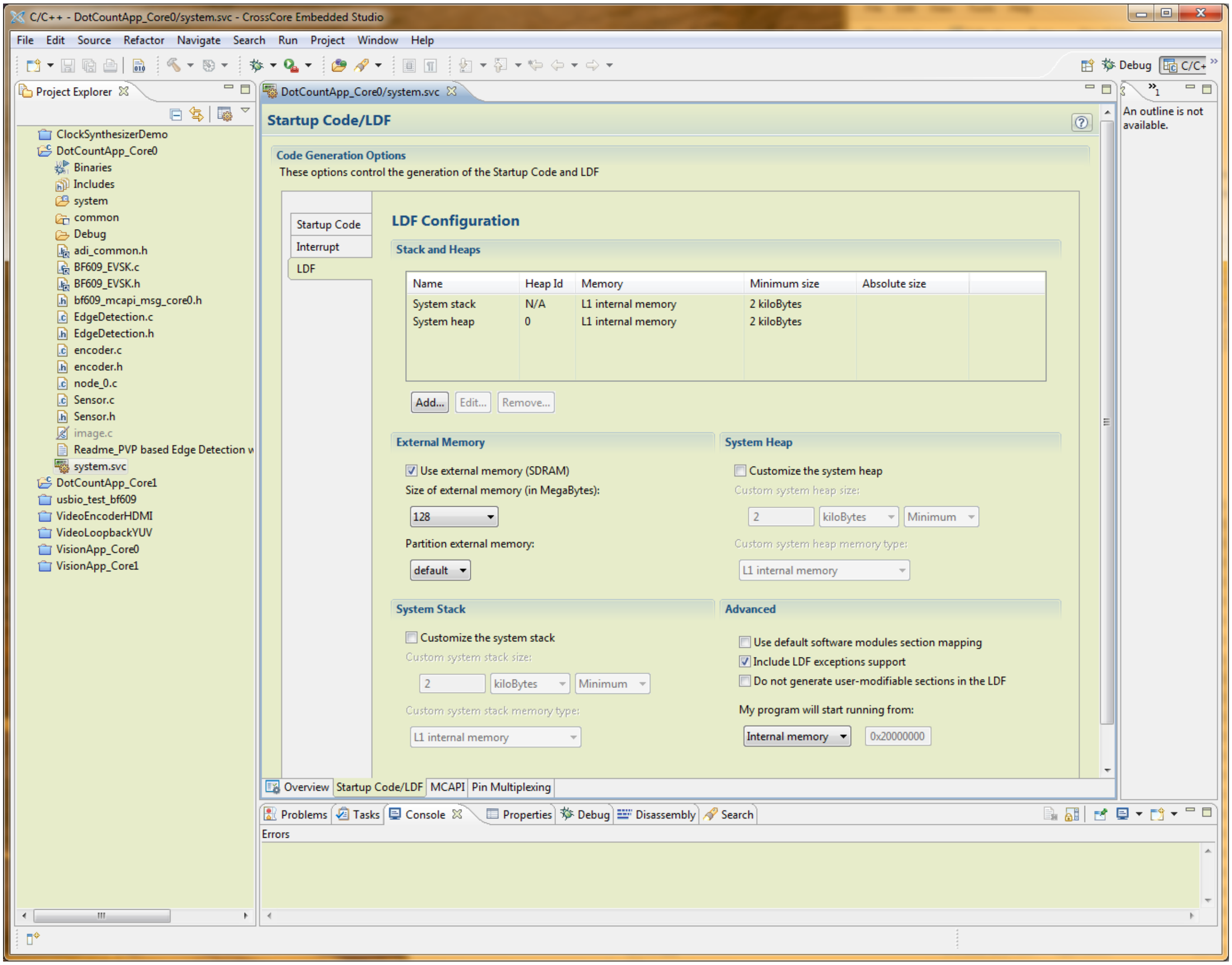
☑ Use custom event handlers at startup

☐ Do not generate user-modifiable sections in the startup code

Figure 2 - Startup Code/LDF Screen

6. Check to ensure the following attributes are selected:
    i. Enable Instruction Cache
    ii. Enable data cache on banks A and B
    iii. Large: 8 MB
    iv. Permit alteration of CPLB Table entries
    v. Write-back data cache modes

7. Next select the LDF window and check the following:



Startup Code/LDF

Code Generation Options

These options control the generation of the Startup Code and LDF

LDF Configuration

Stack and Heaps

Name	Heap Id	Memory	Minimum size	Absolute size
System stack	N/A	L1 internal memory	2 kiloBytes	
System heap	0	L1 internal memory	2 kiloBytes	

External Memory

☑ Use external memory (SDRAM)

Size of external memory (in MegaBytes): 128

Partition external memory: default

System Heap

☐ Customize the system heap

Custom system heap size: 2 kiloBytes Minimum

Custom system heap memory type: L1 internal memory

System Stack

☐ Customize the system stack

Custom system stack size: 2 kiloBytes Minimum

Custom system stack memory type: L1 internal memory

Advanced

☐ Use default software modules section mapping

☑ Include LDF exceptions support

☐ Do not generate user-modifiable sections in the LDF

My program will start running from: Internal memory 0x20000000

Figure 3 - LDF window setup

8. Check to ensure the following attributes are selected:
    i. Use external memory (SDRAM)
    ii. Partition external memory: default

9. In the Project Explorer window, right click on the .DotCountApp_Core0 application and select “build project”. This will create a .dxe for each of the cores. These .dxes will be used in the next step to create the loader file.

IV: Creating a dual core .LDR file for SPI memory

1. Create a directory where you would like to place your programming files. In this example the files are located away from the CCES project directory.

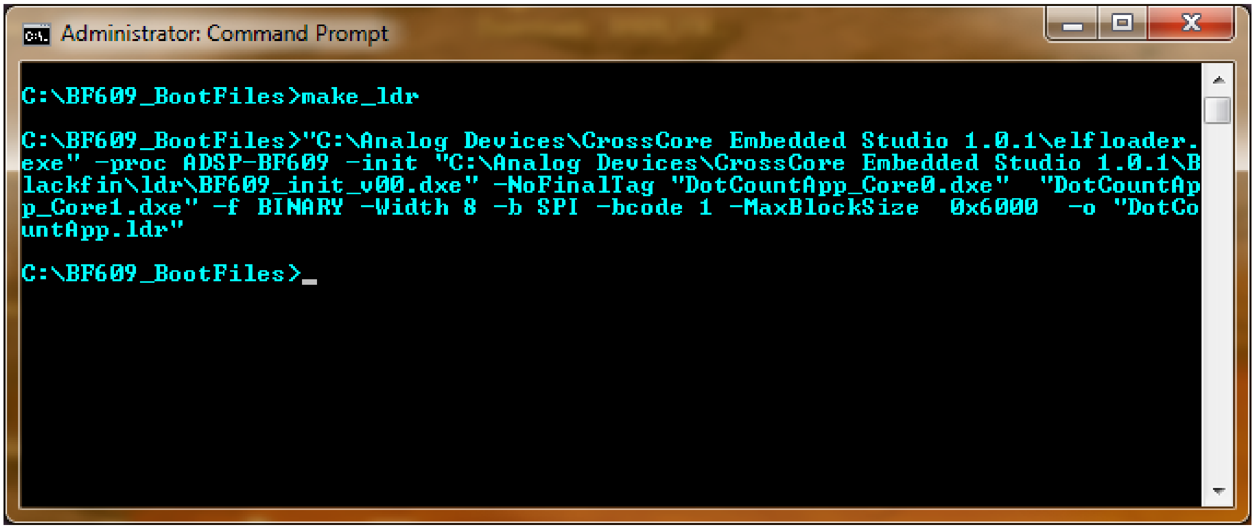
   The current path is: C:\BF609_BootFiles. This document will assume this directory, but you can choose whichever directory is most fitting for your environment. You will have to modify the batch files to meet your environment.

2. Place the following batch files in the C:\BF609_BootFiles directory:
   i. Make_LDR.bat
   ii. Burn_DotSPI.bat

   Note: See Appendix A for the batch file listings.

3. We are going to execute Make_LDR bat to create the programmable dual core loader file for the FinBoard's SPI memory.
   i. Open a Command Prompt window using Administrator privileges. Right click on the Command Prompt icon in the Start Menu and select "Run as Administrator". This will keep the window open after the batch file has executed so you can see what the tool outputs.

   ii. At the command prompt, type "Make_LDR" and press Enter. The .DotCountApp.ldr file will be created in the default directory of C:\BF609_BootFiles. Note the "BINARY", "Width", "bcode" and "MaxBlockSize" attributes in the batch file.



```
C:\BF609_BootFiles>make_ldr

C:\BF609_BootFiles>"C:\Analog Devices\CrossCore Embedded Studio 1.0.1\elfloader.
exe" -proc ADSP-BF609 -init "C:\Analog Devices\CrossCore Embedded Studio 1.0.1\B
lackfin\ldr\BF609_init_v00.dxe" -NoFinalTag "DotCountApp_Core0.dxe"  "DotCountAp
p_Core1.dxe" -f BINARY -Width 8 -b SPI -bcode 1 -MaxBlockSize  0x6000  -o "DotCo
untApp.ldr"

C:\BF609_BootFiles>_
```

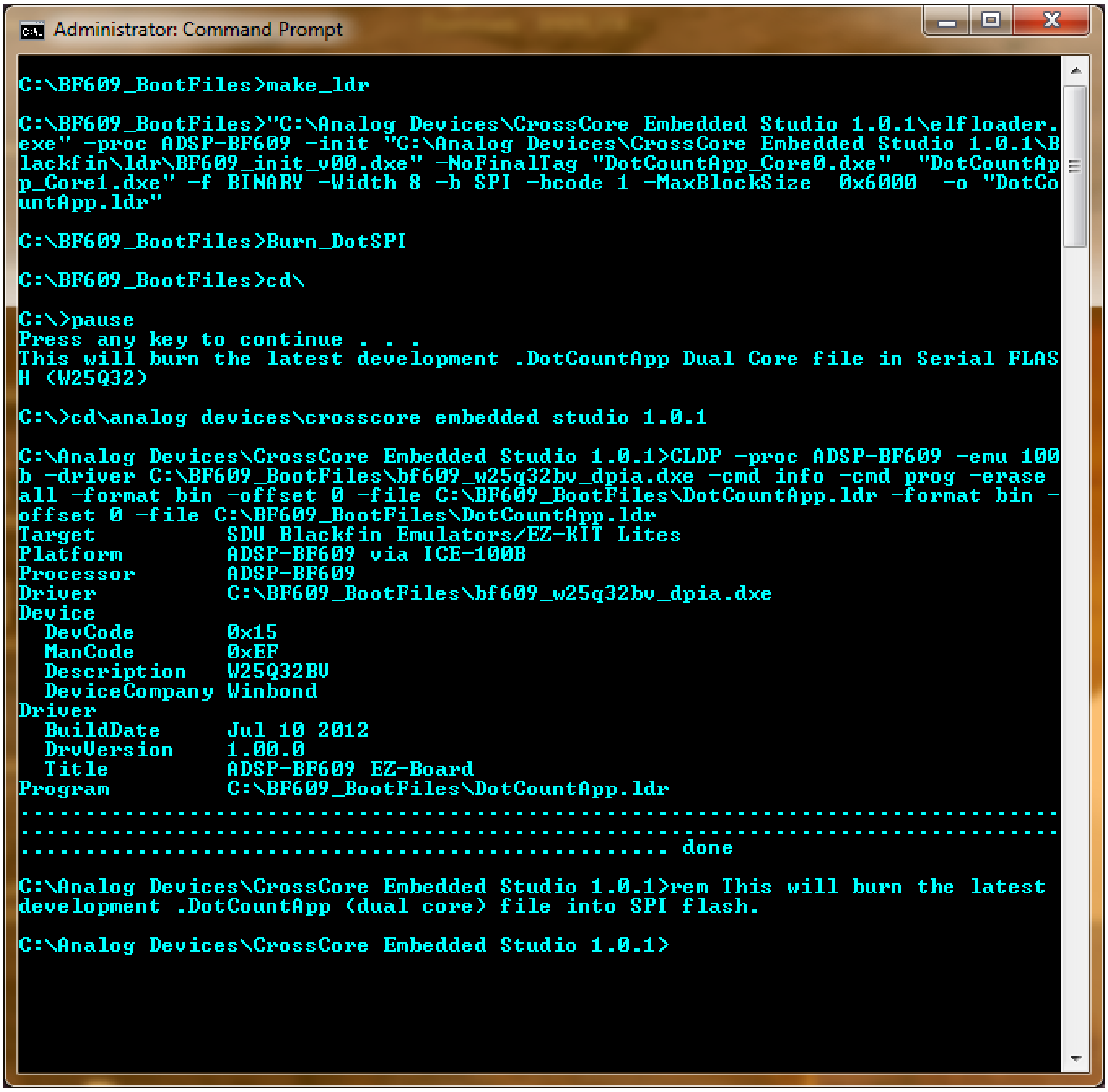
Figure 4 - Make_LDR command line window

V: Programming the .LDR file into SPI memory

1. On the FinBoard make sure the boot memory select switch SW 1.1 is in the “up” position (closed). The remaining switches should be in the “down” position (open).

2. Make sure the ***bf609_w25q32bv_dpia.dxe*** file is in the C:\BF609_BootFiles directory. This is the SPI programming driver file. You can get this file from the Analog Devices EZ-Kit directory or online at the Avnet Egnyte\Devonshire\board directory.

3. At the Command Prompt and in the C:\BF609_BootFiles directory, type “Burn_DOTSPI” and press Enter. Follow the command line prompts. It will take approximately 80 seconds to program the SPI.



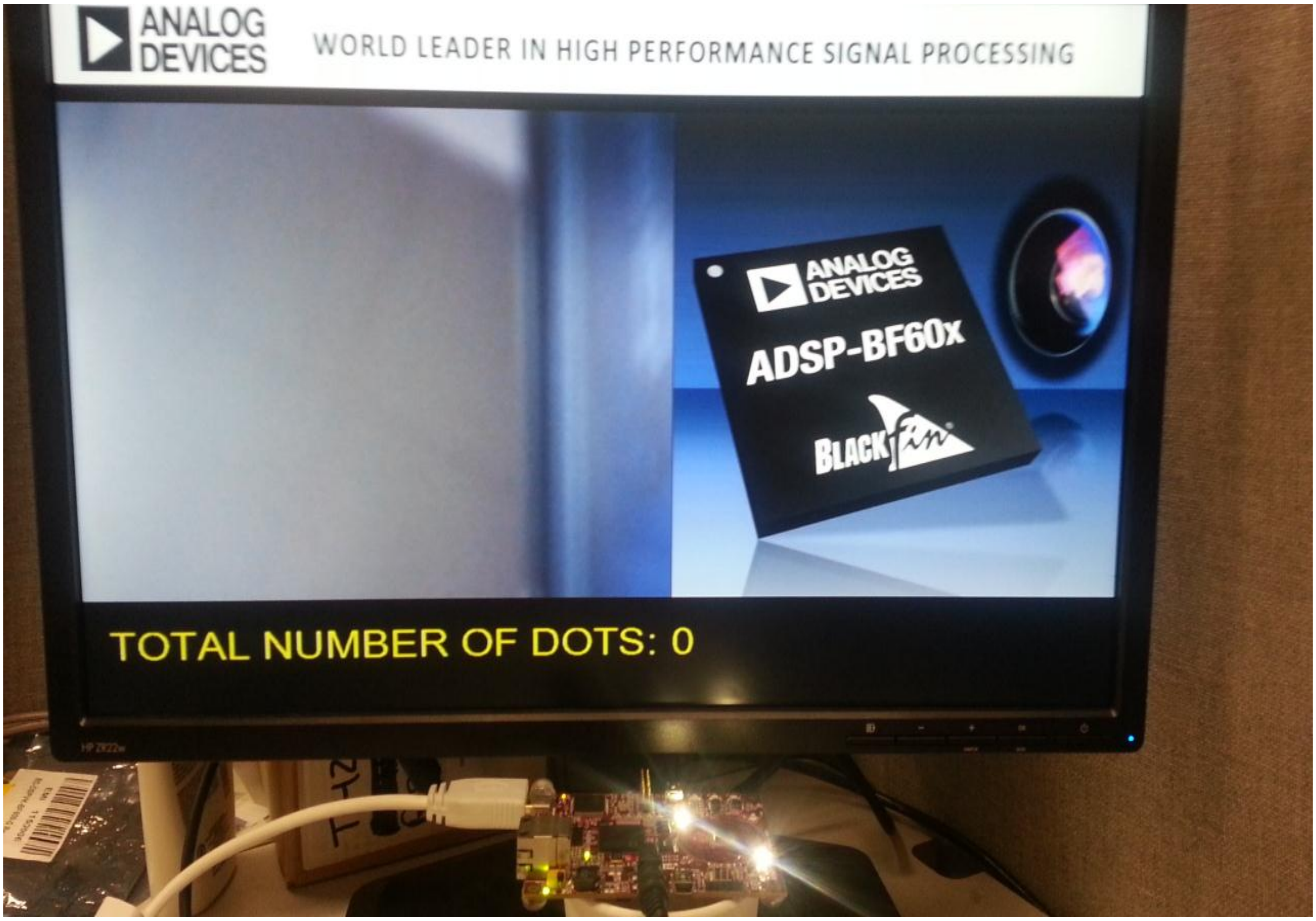
```
C:\BF609_BootFiles>make_ldr

C:\BF609_BootFiles>"C:\Analog Devices\CrossCore Embedded Studio 1.0.1\elfloader.
exe" -proc ADSP-BF609 -init "C:\Analog Devices\CrossCore Embedded Studio 1.0.1\B
lackfin\ldr\BF609_init_v00.dxe" -NoFinalTag "DotCountApp_Core0.dxe"  "DotCountAp
p_Core1.dxe" -f BINARY -Width 8 -b SPI -bcode 1 -MaxBlockSize  0x6000  -o "DotCo
untApp.ldr"

C:\BF609_BootFiles>Burn_DotSPI

C:\BF609_BootFiles>cd\

C:\>pause
Press any key to continue . . .
This will burn the latest development .DotCountApp Dual Core file in Serial FLAS
H (W25Q32)

C:\>cd\analog devices\crosscore embedded studio 1.0.1

C:\Analog Devices\CrossCore Embedded Studio 1.0.1>CLDP -proc ADSP-BF609 -emu 100
b -driver C:\BF609_BootFiles\bf609_w25q32bv_dpia.dxe -cmd info -cmd prog -erase
all -format bin -offset 0 -file C:\BF609_BootFiles\DotCountApp.ldr -format bin -
offset 0 -file C:\BF609_BootFiles\DotCountApp.ldr
Target          SDU Blackfin Emulators/EZ-KIT Lites
Platform        ADSP-BF609 via ICE-100B
Processor       ADSP-BF609
Driver          C:\BF609_BootFiles\bf609_w25q32bv_dpia.dxe
Device
  DevCode       0x15
  ManCode       0xEF
  Description   W25Q32BV
  DeviceCompany Winbond
Driver
  BuildDate     Jul 10 2012
  DrvVersion    1.00.0
  Title         ADSP-BF609 EZ-Board
Program         C:\BF609_BootFiles\DotCountApp.ldr
................................................................................
................................................................................
................................................... done

C:\Analog Devices\CrossCore Embedded Studio 1.0.1>rem This will burn the latest
development .DotCountApp (dual core) file into SPI flash.

C:\Analog Devices\CrossCore Embedded Studio 1.0.1>
```

Figure 5 - Dual Core .LDR programmed in SPI

4. After the SPI is programmed power-cycle the FinBoard.

5. Two High Bright LEDs (D28 and D30) should illuminate. Depress pushbutton switch SW4 to change LED intensity from 0 to 7, thus verifying the load is successful.

6. If attached, the camera's image should appear on the HDMI monitor along with the ADI BlackFin banner.



TOTAL NUMBER OF DOTS: 0

Appendix A – File Listings:

NOTE: Ensure there are no spaces in this file listing other than what is present. Also note the highlighted areas that must be set for the application to properly program.

1 - Make_LDR.bat listing:

```
"C:\Analog Devices\CrossCore Embedded Studio 1.0.1\elfloader.exe" -proc ADSP-BF609 -init "C:\Analog
Devices\CrossCore Embedded Studio 1.0.1\Blackfin\ldr\BF609_init_v00.dxe" –NoFinalTag
"DotCountApp_Core0.dxe"  "DotCountApp_Core1.dxe" -f BINARY -Width 8 -b SPI -bcode 1 -
MaxBlockSize 0x6000  -o "DotCountApp.ldr"
```

2 - Burn_DotSPI.bat listing:

```
cd\
pause
@echo This will burn the latest development .DotCountApp Dual Core file in Serial FLASH (W25Q32)

cd\analog devices\crosscore embedded studio 1.0.1
CLDP -proc ADSP-BF609 -emu 100b -driver C:\BF609_BootFiles\bf609_w25q32bv_dpia.dxe -cmd info -
cmd prog -erase all -format bin -offset 0 -file C:\BF609_BootFiles\DotCountApp.ldr -format bin –offset
0 -file C:\BF609_BootFiles\DotCountApp.ldr

rem This will burn the latest development .DotCountApp (dual core) file into SPI flash.
```