FinBoard™

Avnet's Blackfin BF609
Embedded Vision Starter Kit

# Getting Started Guide

Version 1.0

AVNET® electronics marketing

# Revision History

| Version | Description | Date |
|---|---|---|
| 1.0 | Initial release | April 5, 2013 |

# Table of Contents

Revision History....................i.
Table of Contents....................ii.
Table of Figures....................iii.
FinBoard BF609 Embedded Vision Starter Kit....................iv.
About this Guide....................- 1 -
Additional Resources....................- 1 -
Support for FinBoard....................- 1 -
Support for Blackfin BF609 Processor....................- 1 -
Design Tutorials....................- 1 -
Introduction....................- 2 -
Hardware Overview....................- 2 -
What's Inside the Box....................- 5 -
Requirements....................- 5 -
Software....................- 5 -
Hardware....................- 6 -
Installing Software....................- 6 -
CrossCore Embedded Studio Installation....................- 7 -
BDTI Dice Dot Counting Demo Installation....................- 7 -
FinBoard Board Support Package (BSP)....................- 7 -
Licensing....................- 7 -
ADZS-ICE-100B Emulator Driver Installation....................- 8 -
BDTI Dice Dot Counting Demo and Reference Design....................- 8 -
Overview....................- 8 -
Who Created the Application?....................- 9 -
Setting up the Demo....................- 10 -
Hardware Setup....................- 10 -
Software Setup – CCES Workspace Creation and Project Import....................- 13 -
Loading the Demo....................- 22 -
Option (A) – Booting from SD Flash Card....................- 23 -
Option (B) – Loading the Pre-built Application....................- 23 -
Option (C) – Building and Loading the Application....................- 27 -
Running the Demo....................- 30 -
Operating Modes....................- 30 -
Focusing the Lens – IMPORTANT!....................- 31 -

Canny Edge Detection Example - 32 -
References - 34 -
FinBoard - 34 -
Analog Devices - 34 -
Aptina - 34 -
Known Issues and Limitations - 35 -
Application Shows "0" For Large Number of Dots - 35 -
Troubleshooting - 36 -
Dice Dot Counting Demo not loading at power up - 36 -
Dice count result is incorrect or unstable - 36 -
Could not open source file "adi_graphics.h" - 36 -

# Table of Figures

Figure 1 - FinBoard Feature Map - 3 -
Figure 2 – FinBoard Block Diagram - 4 -
Figure 3 - BDTI Dice Dot Counting Demonstration - 9 -
Figure 4 - Dice Dot Counting Demo Hardware Setup - 10 -
Figure 5 - Attach tripod mount - 11 -
Figure 6 - Tripod and ICE-100B attached - 11 -
Figure 7 - Attaching FinBoard to box - 12 -
Figure 8 - FinBoard Demo Hardware Setup (ICE-100B not shown) - 13 -
Figure 9 - Focus with lens tool or adhesive tape - 31 -
Figure 10 - Assembled FinBoard Kit Running the BDTI Dice Dot Counting Demo - 32 -

# FinBoard BF609 Embedded Vision Starter Kit

## LICENSE AGREEMENT

THE AVNET DESIGN KIT (“DESIGN KIT” OR “PRODUCT”) AND ANY SUPPORTING DOCUMENTATION (“DOCUMENTATION” OR “PRODUCT DOCUMENTATION”) IS SUBJECT TO THIS LICENSE AGREEMENT (“LICENSE”). USE OF THE PRODUCT OR DOCUMENTATION SIGNIFIES ACCEPTANCE OF THE TERMS AND CONDITIONS OF THIS LICENSE. THE TERMS OF THIS LICENSE AGREEMENT ARE IN ADDITION TO THE AVNET CUSTOMER TERMS AND CONDITIONS, WHICH CAN BE VIEWED AT www.em.avnet.com. THE TERMS OF THIS LICENSE AGREEMENT WILL CONTROL IN THE EVENT OF A CONFLICT.

1. Limited License. Avnet grants You, the Customer, (“You” “Your” or “Customer”) a limited, non-exclusive, non-transferable, license to: (a) use the Product for Your own internal testing, evaluation and design efforts at a single Customer site; (b) create a single derivative work based on the Product using the same semiconductor supplier product or product family as used in the Product; and (c) make, use and sell the Product in a single production unit. No other rights are granted and Avnet and any other Product licensor reserves all rights not specifically granted in this License Agreement. Except as expressly permitted in this License, neither the Design Kit, Documentation, nor any portion may be reverse engineered, disassembled, decompiled, sold, donated, shared, leased, assigned, sublicensed or otherwise transferred by Customer. The term of this License is in effect until terminated. Customer may terminate this license at any time by destroying the Product and all copies of the Product Documentation.
2. Changes. Avnet may make changes to the Product or Product Documentation at any time without notice. Avnet makes no commitment to update or upgrade the Product or Product Documentation and Avnet reserves the right to discontinue the Product or Product Documentation at any time without notice.
3. Limited Warranty. ALL PRODUCTS AND DOCUMENTATION ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND. AVNET MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THE PRODUCTS AND DOCUMENTATION PROVIDED HEREUNDER. AVNET SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY AGAINST INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF ANY THIRD PARTY WITH REGARD TO THE PRODUCTS AND DOCUMENTATION.
4. **LIMITATIONS OF LIABILITY. CUSTOMER SHALL NOT BE ENTITLED TO AND AVNET WILL NOT LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY KIND OR NATURE, INCLUDING, WITHOUT LIMITATION, BUSINESS INTERRUPTION COSTS, LOSS OF PROFIT OR REVENUE, LOSS OF DATA, PROMOTIONAL OR MANUFACTURING EXPENSES, OVERHEAD, COSTS OR EXPENSES ASSOCIATED WITH WARRANTY OR INTELLECTUAL PROPERTY INFRINGEMENT CLAIMS, INJURY TO REPUTATION OR LOSS OF CUSTOMERS, EVEN IF AVNET HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE PRODUCTS AND DOCUMENTATION ARE NOT DESIGNED, AUTHORIZED OR WARRANTED TO BE SUITABLE FOR USE IN MEDICAL, MILITARY, AIR CRAFT, SPACE OR LIFE SUPPORT EQUIPMENT NOR IN APPLICATIONS WHERE FAILURE OR MALFUNCTION OF THE PRODUCTS CAN REASONABLY BE EXPECTED TO RESULT IN A PERSONAL INJURY, DEATH OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. INCLUSION OR USE OF PRODUCTS IN SUCH EQUIPMENT OR APPLICATIONS, WITHOUT PRIOR AUTHORIZATION IN WRITING OF AVNET, IS NOT PERMITTED AND IS AT CUSTOMER’S OWN RISK. CUSTOMER AGREES TO FULLY INDEMNIFY AVNET FOR ANY DAMAGES RESULTING FROM SUCH INCLUSION OR USE.**
5. **LIMITATION OF DAMAGES. CUSTOMER’S RECOVERY FROM AVNET FOR ANY CLAIM SHALL NOT EXCEED CUSTOMER’S PURCHASE PRICE FOR THE PRODUCT GIVING RISE TO SUCH CLAIM IRRESPECTIVE OF THE NATURE OF THE CLAIM, WHETHER IN CONTRACT, TORT, WARRANTY, OR OTHERWISE.**
6. **INDEMNIFICATION. AVNET SHALL NOT BE LIABLE FOR AND CUSTOMER SHALL INDEMNIFY, DEFEND AND HOLD AVNET HARMLESS FROM ANY CLAIMS BASED ON AVNET’S COMPLIANCE WITH CUSTOMER’S DESIGNS, SPECIFICATIONS OR IN¬STRUCTIONS, OR MODIFICATION OF ANY PRODUCT BY PARTIES OTHER THAN AVNET, OR USE IN COMBINATION WITH OTHER PRODUCTS.**
7. U.S. Government Restricted Rights. The Product and Product Documentation are provided with “RESTRICTED RIGHTS.” If the Product and Product Documentation and related technology or documentation are provided to or made available to the United States Government, any use, duplication, or disclosure by the United States Government is subject to restrictions applicable to proprietary commercial computer software as set forth in FAR 52.227-14 and DFAR 252.227-7013, et seq., its successor and other applicable laws and regulations. Use of the Product by the United States Government constitutes acknowledgment of the proprietary rights of Avnet and any third parties. No other governments are authorized to use the Product without written agreement of Avnet and applicable third parties.
8. Ownership. Licensee acknowledges and agrees that Avnet or Avnet’s licensors are the sole and exclusive owner of all Intellectual Property Rights in the Licensed Materials, and Licensee shall acquire no right, title, or interest in the Licensed Materials, other than any rights expressly granted in this Agreement.
9. Intellectual Property. All trademarks, service marks, logos, slogans, domain names and trade names (collectively “Marks”) are the properties of their respective owners. Avnet disclaims any proprietary interest in Marks other than its

own. Avnet and AV design logos are registered trademarks and service marks of Avnet, Inc. Avnet's Marks may be used only with the prior written permission of Avnet, Inc.

**10.** General. The terms and conditions set forth in the License Agreement or at www.em.avnet.com will apply notwithstanding any conflicting, contrary or additional terms and conditions in any purchase order, sales acknowledgement confirmation or other document. If there is any conflict, the terms of this License Agreement will control. This License may not be assigned by Customer, by operation of law, merger or otherwise, without the prior written consent of Avnet and any attempted or purported assignment shall be void. Licensee understands that portions of the Licensed Materials may have been licensed to Avnet from third parties and that such third parties are intended beneficiaries of the provisions of this Agreement. In the event any of the provisions of this Agreement are for any reason determined to be void or unenforceable, the remaining provisions will remain in full effect.This constitutes the entire agreement between the parties with respect to the use of this Product, and supersedes all prior or contemporaneous understandings or agreements, written or oral, regarding such subject matter. No waiver or modification is effective unless agreed to in writing and signed by authorized representatives of both parties. The obligations, rights, terms and conditions shall be binding on the parties and their respective successors and assigns. The License Agreement is governed by and construed in accordance with the laws of the State of Arizona excluding any law or principle, which would apply the law of any other jurisdiction. The United Nations Convention for the International Sale of Goods shall not apply.

# About this Guide

This guide provides an introduction to FinBoard from Avnet Electronics Marketing. It contains an overview of the Embedded Vision Starter Kit, software installation, and step-by-step instructions for running the BDTI Dice Dot Counting Demo and Reference Design, and the Canny Edge Detection example.

# Additional Resources

## Support for FinBoard

All documentation and support for FinBoard are located at the product website: www.FinBoard.org. There you will find the following items:

- Design tutorials
- BDTI Dice Dot Counting Demo and Reference Design Software User Guide
- Hardware User's Guide
- Hardware schematics
- Technical support forum

## Support for Blackfin BF609 Processor

To search a database of answers or request technical support for the BF609 Blackfin Dual-Core Processor or CrossCore Embedded Studio, please visit the Analog Devices online technical support community at: http://ez.analog.com

## Design Tutorials

Several design tutorials are available to help users learn how to start developing custom embedded vision applications using FinBoard and the CrossCore Embedded Studio tools. They are available for download at www.FinBoard.org.

# Introduction

The Analog Devices ADSP-BF609 Blackfin processor is optimized for embedded vision and video analytics applications using a dual-core fixed-point DSP processor with a unique pipelined vision processor (PVP). The PVP is a set of functional blocks next to the Blackfin cores designed to accelerate image processing algorithms and reduce overall bandwidth requirements.

FinBoard is an evaluation platform for the Blackfin BF609 dual-core processor, targeting embedded vision applications. It is based on the modular Analog Devices BF609 EZ-KIT. Avnet's FinBoard is a smaller footprint, self-contained version of the EZ-KIT with on-board HD CMOS image sensor for video capture, high brightness LEDs for target illumination, and HDMI output for display. To understand the differences from the EZ-KIT, please consult the FinBoard Hardware User's Guide.

## Hardware Overview

FinBoard has the following features and interfaces:

- Blackfin ADSP-BF609 500 MHz Dual Core Processor
- 128MB DDR2 SDRAM
- 32Mb Quad SPI Flash
- Aptina MT9M114 HD 720p CMOS Color Image Sensor
- AD7511 HDMI video encoder
- OSRAM High Flux LEDs for target illumination
- 10/100 Ethernet
- USB-OTG
- ADM1032 Temperature Sensor
- MicroSD Flash Memory Card interface
- IDT5V9885T programmable multi-clock generator
- Tripod mount
- User pushbuttons and LEDs

**Figure 1** points out the main features and interfaces on FinBoard while **Figure 2** shows a simplified view of the block diagram.

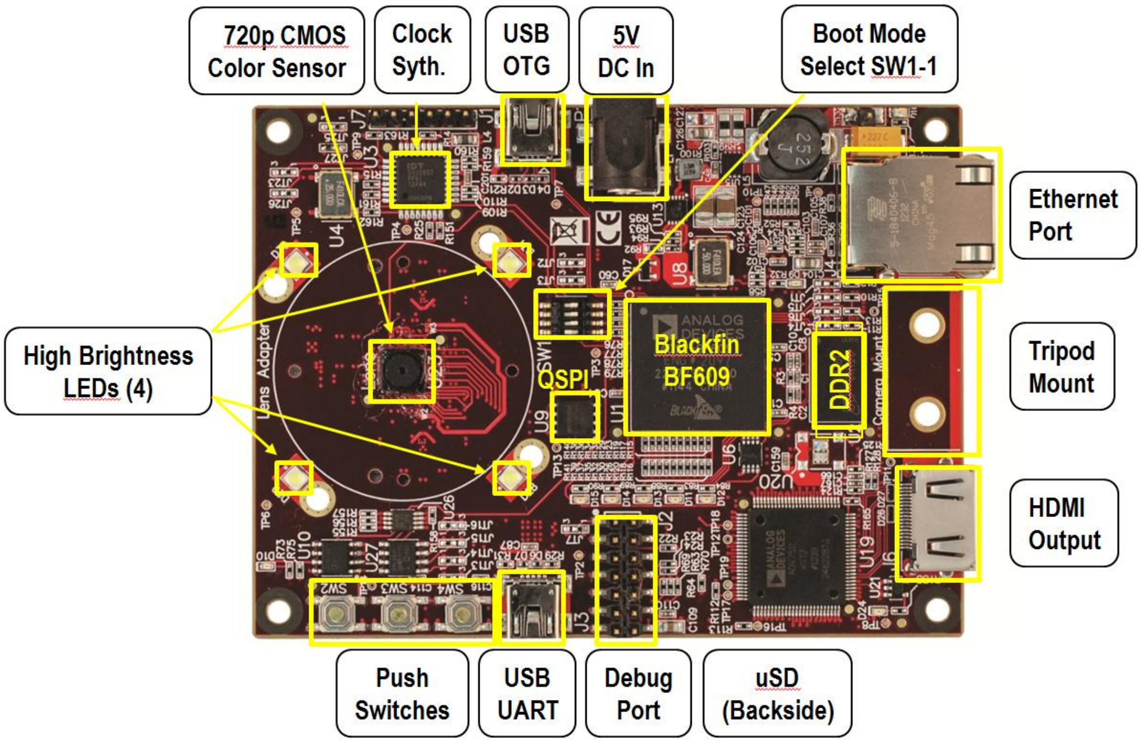


**Figure 1 - FinBoard Feature Map**

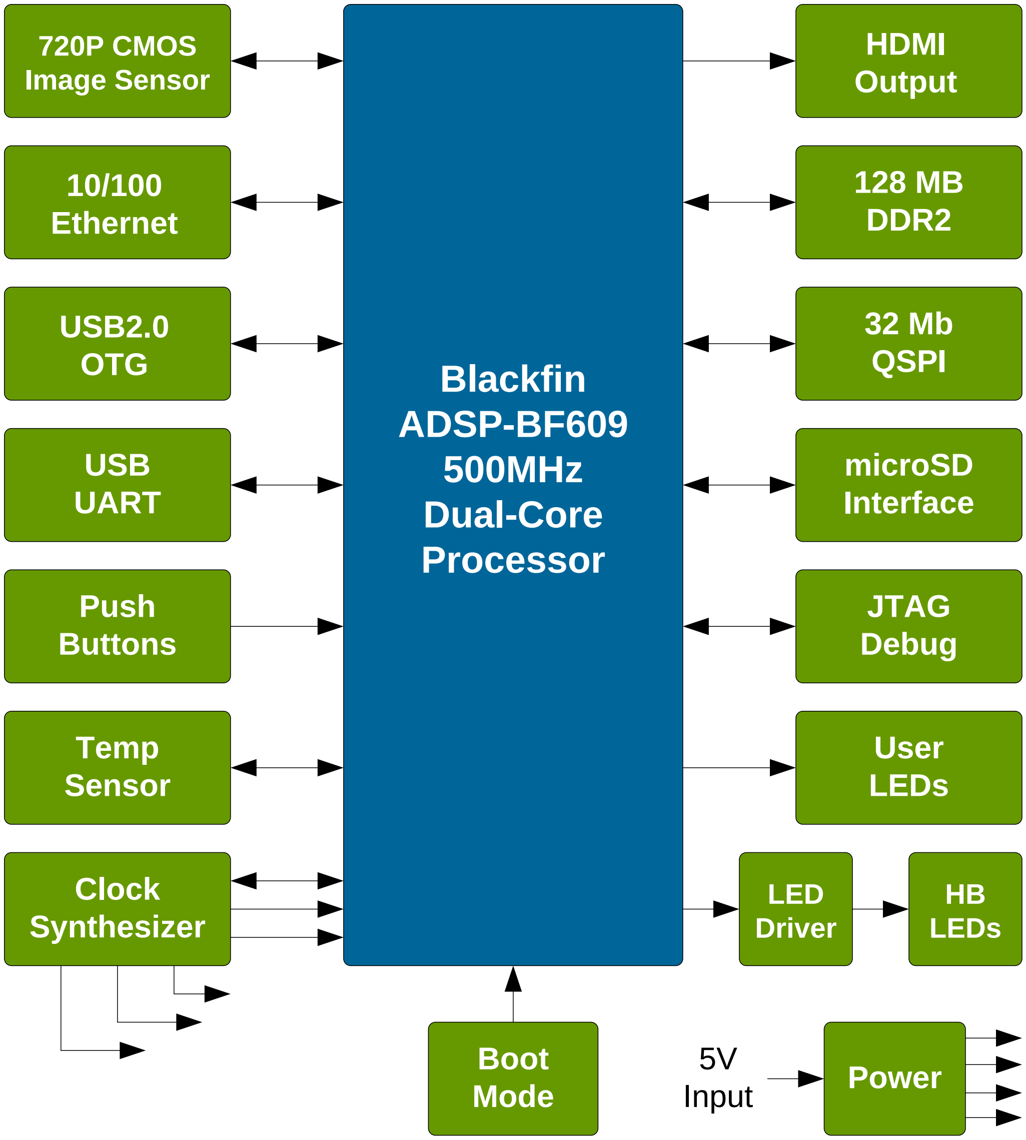


**Figure 2 – FinBoard Block Diagram**

# What's Inside the Box

The FinBoard Embedded Vision Starter kit contains everything required to run the BDTI Dice Dot Counting Demo and Reference Design, the design tutorials, and start development right out of the box. The kit includes:

- Avnet FinBoard
- Analog Devices ICE-100B JTAG debugger
- 2 GB microSD card
- 5V Power supply
- HDMI cable
- USB cable (Type A to Mini-B)
- Tripod
- 5 White dice
- 12-month CrossCore™ Embedded Studio license (two, 6-month licenses)
- Quick Start card

# Requirements

The software and hardware requirements are outlined in the following sections.

### Software

Building and loading the demo application requires download and installation of the following software packages. The URLs and files are listed below. Choose the latest production release unless otherwise stated.

- **ADI CrossCore® Embedded Studio (CCES)**
  - URL: www.analog.com/cces
  - CrossCore Embedded Studio software
  - CrossCore Embedded Studio Patch software

- **ADI Camera EI3 Extender Board Support Package (BSP) Software**
  - URL: www.analog.com/EX3-Camera

- **ADI Video Encoder EI3 Extender Board Support Package (BSP) Software**
  - URL: www.analog.com/EX3-VidEncoder

- **ADI Blackfin 2D Graphics Library (BF2DGL-OGL / OpenGL-based)**
  - URL: www.analog.com/BF-GFX2D-00

- **ADI Blackfin Image Processing Toolbox**
  - URL: www.analog.com/BF-IPTBX-00

- **ADI Blackfin Vision Vision Analytics Toolbox (VAT)**
  - **VAT (OBJDT-01) Rel 2.3.0 or later**
  - URL: www.analog.com/BF-VAT

- **ADI Board Support Package for BF609 EZ-Kit**
  - URL: www.analog.com/BF609EZBoard

- **BDTI Dice Dot Counting Demo and Reference Design**
  - URL: www.finboard.org/design

### Hardware

The minimum hardware required to run the reference design is:

- Host PC with Windows XP or Windows 7
- FinBoard hardware
- ICE-100B JTAG emulator (optional)
- HDMI (or DVI-D) monitor capable of 720p60 video resolution

# Installing Software

In order to build application source files and load them onto FinBoard you must first install Analog Devices CrossCore Embedded Studio (CCES) tools and associated support packages for FinBoard. Note that the BDTI Dice Dot Counting Demo is pre-loaded in the microSD flash card included with FinBoard. If you do not wish to re-build or load the application with CCES and the ICE-100B emulator, you may skip the software installation steps and go to the **BDTI Dice Dot**

**Counting Demo and Reference Design** section.

## CrossCore Embedded Studio Installation

1. Download the files listed in the previous section of software requirements.
2. First double-click **ADI_CrossCoreEmbeddedStudio-Rel1.0.1.exe** to install CCES.

   **NOTE**:  accept the default installation for all tools
3. Once the CCES installation completes, install the CCES patch software
4. Next, install the remaining packages in any order by double-clicking on each .exe file.

## BDTI Dice Dot Counting Demo Installation

1. Unzip the contents of the **BDTI_DiceDotCountingDemo.zip** file to your root C:\ drive so that your file path is → **C:\BDTI_DiceDotCountingDemo**

## FinBoard Board Support Package (BSP)

FinBoard is very similar in function to the Analog Devices BF609 EZ-KIT.  Therefore only a small number of modifications are required to the EZ-KIT BSP that you installed in the previous section. These modifications are delivered in the following files:

- Avnet FinBoard Board Support Package (BSP)
  - **finboard_bsp.c**
  - **finboard_bsp.h**

Please note that the reference designs and tutorials posted on www.FinBoard.org already have these files installed in the project.  They can be found in the “common” folder in the project directory.

## Licensing

Your FinBoard purchase includes two 6 month CCES licenses that may be used sequentially by one user to achieve one year of full featured development, or in parallel by two users for 6 months.  In order to activate CCES with your license, invoke the CCES tool and enter the license number printed on your voucher when prompted. To update your license information at any time, simply click on **Help → Manage Licenses…**

By default your CCES license information is stored in the following location:

C:\ProgramData\Analog Devices\CrossCore Embedded Studio\license.dat

### ADZS-ICE-100B Emulator Driver Installation

The ICE-100B emulator is used to load binary files and debug code on the BF609 processor. A driver must be installed on the host PC before you can use the ICE-100B with FinBoard. Be sure that you have installed the CCES software before attaching your ICE-100B to the host PC.

1. Plug the ICE-100B module into the host PC using the provided USB cable.
2. The Windows hardware setup wizard will pop up.
3. Click ‘No, not this time’ if prompted to search Windows Update for the driver.
4. Select the option to install automatically.
5. If the driver cannot be found automatically, select the option to “Install from a list or specific location” and press “Next” button in the window.
6. Select the “Search for the best driver in these locations” option.
7. Enable option “Include this location in search” by clicking on the check list and add the location **<CCES Installation Directory>\Setup\Emu_Drivers** to the search path. Press the “Next” button.
8. The wizard will open a window showing “Completing the Found New Hardware Wizard”. Click the “Finish” button and the ICE will be ready to use.

## BDTI Dice Dot Counting Demo and Reference Design

### Overview

Many vision applications must detect objects in live video. To demonstrate the capabilities of the Blackfin BF609 processor, FinBoard includes a unique application that is able to detect and count the number of dots on dice. The color image sensor delivers raw video to the Blackfin BF609, where the dedicated Pipelined Vision Processor (PVP) performs accelerated edge detection. A software application running on the dual Blackfin cores then finds contours in the image and determines which are valid dots to be counted. Finally, a graphic overlay is applied, tracing the images edges, highlighting the dots, and displaying the count on the HDMI monitor attached to FinBoard.

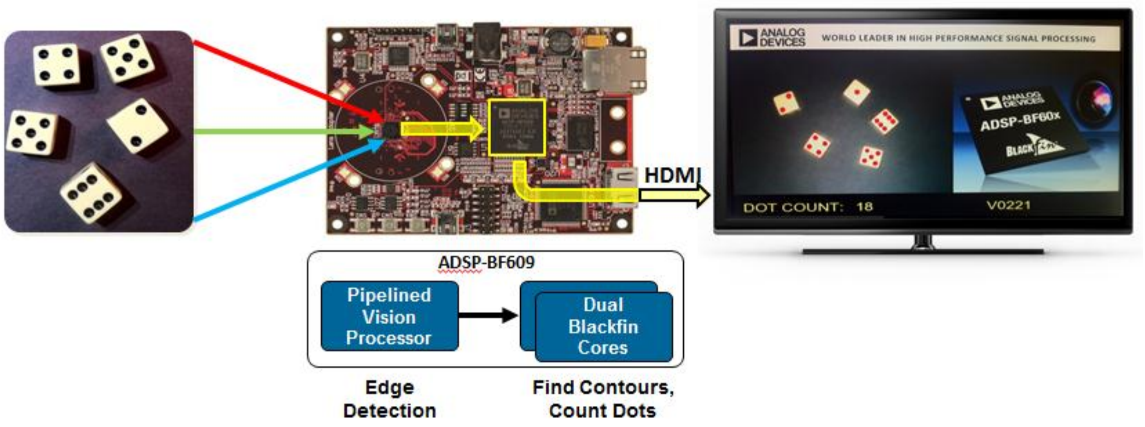

HDMI
ANALOG DEVICES
WORLD LEADER IN HIGH PERFORMANCE SIGNAL PROCESSING
ADSP-BF60x
DOT COUNT: 18
V0221
ADSP-BF609
Pipelined Vision Processor
Dual Blackfin Cores
Edge Detection
Find Contours, Count Dots

**Figure 3 - BDTI Dice Dot Counting Demonstration**

➔ The "**BDTI Dice Dot Counting Demo and Reference Design Software User's Guide**" is posted at www.FinBoard.org and provides a detailed explanation of the software algorithms used in the demo application.

## Who Created the Application?

The dice-counting application software provided in the FinBoard Blackfin BF609 Embedded Vision Starter Kit was created by Berkeley Design Technology, Inc. (BDTI). BDTI is the industry's most trusted source of analysis, advice, and engineering for embedded processing technology and applications. For more than 20 years, BDTI has been the go-to resource for highly optimized, highly efficient implementations of signal processing applications on embedded processing platforms. Contact BDTI at http://www.BDTI.com/Contact.

Information about BDTI's software engineering services may be found at http://www.BDTI.com/Services/Engineering/Software and information about BDTI's experience and expertise in designing embedded vision applications may be found at http://www.BDTI.com/Services/Engineering/EmbeddedVision.

## Setting up the Demo

### Hardware Setup

The box that FinBoard is packaged in has a clever alternative use. With the addition of the tripod and mount (included with the kit), the box is used to suspend FinBoard above the viewing area for the BDTI Dice Dot Counting Demo, as shown in **Figure 4**. Step-by-step instructions for assembling the pieces are given below.

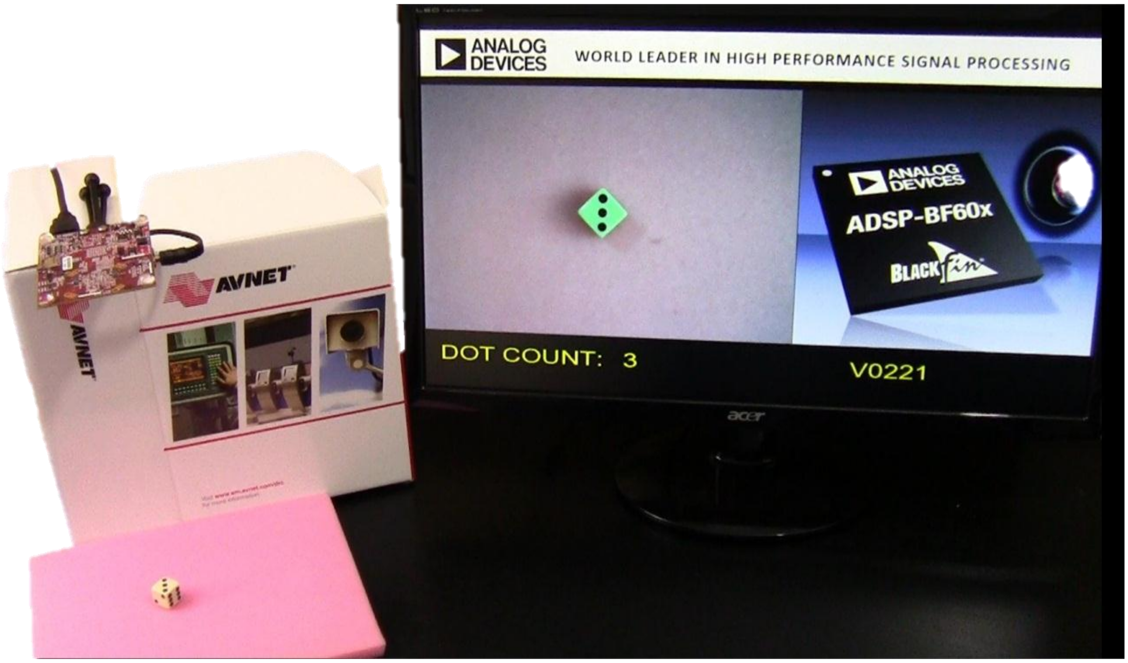


**Figure 4 - Dice Dot Counting Demo Hardware Setup**

**IMPORTANT!** The BDTI Dice Dot Counting Demo software parameters have been tuned specifically for this setup. It is important that the kit is set up as shown in order to achieve robust operation of the application.

1. Attach the tripod mounting adapter to FinBoard with the (2) screws provided.

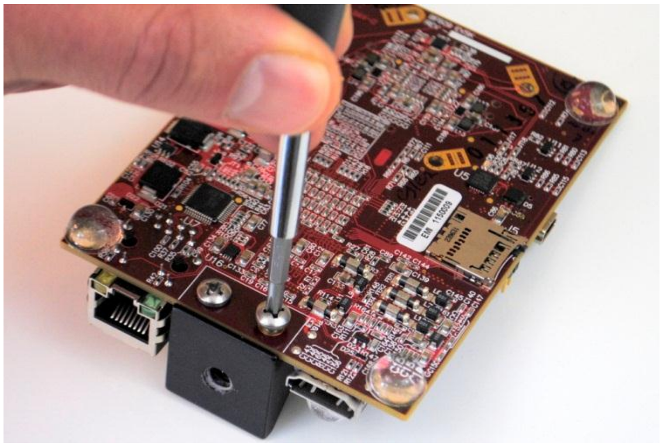


**Figure 5 - Attach tripod mount**

2. Attach the tripod to the mount.

3. Attach the ICE-100B module (only if you plan to load the application from CCES). FinBoard should like the image shown below.

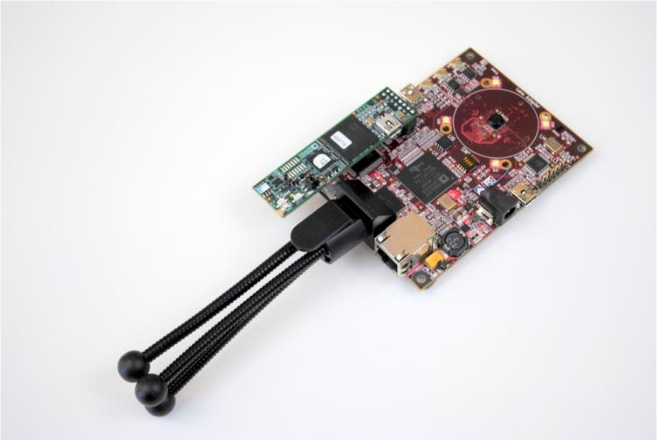


**Figure 6 - Tripod and ICE-100B attached**

4. Attach the provided USB cable to the ICE-100B and your host PC.

Next you will use the tripod's integrated clip to attach FinBoard to the top corner of the box.

5. Position the Avnet box as shown in **Figure 4** with the flap opened 90 degrees to add stability to the setup.

6. Slide the tripod clip between the cardboard at the corner fold of the box as shown below.

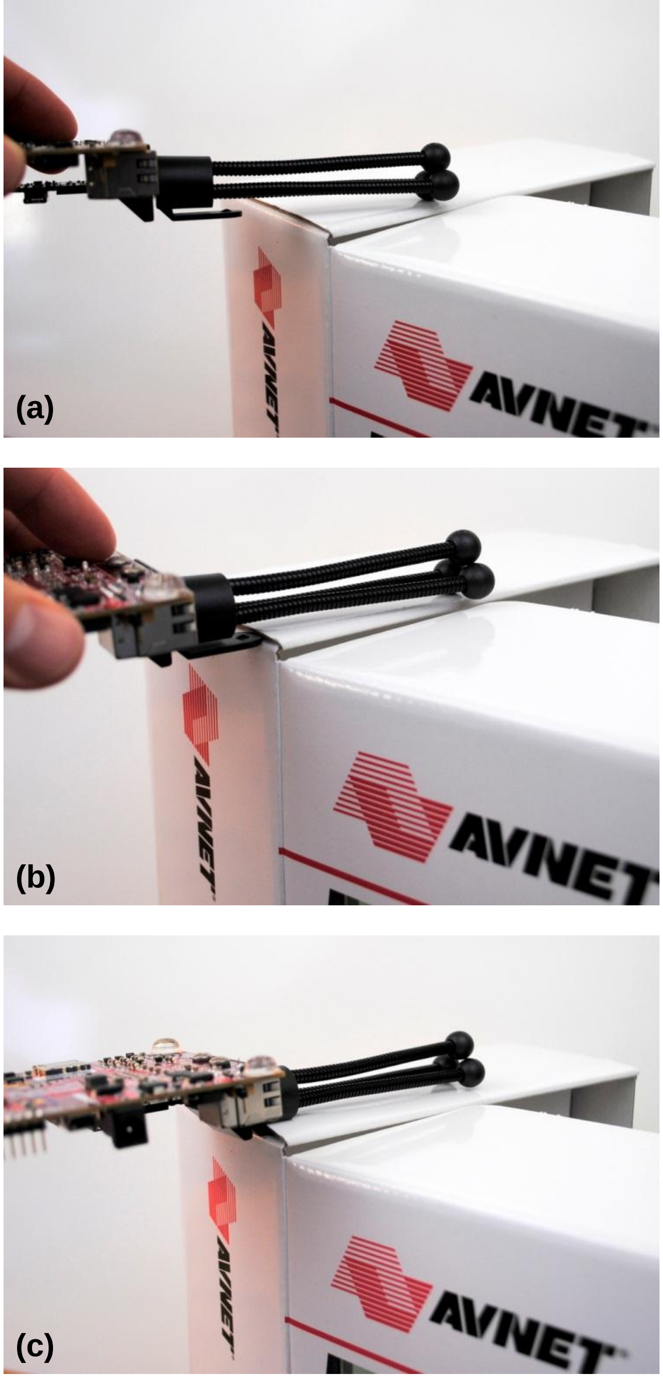


**Figure 7 - Attaching FinBoard to box**

7. Be sure to slide the clip in as far as possible for best stability.

8. Plug the HDMI cable into FinBoard and the display monitor. Your monitor must be capable of 720P60 video resolution or greater.

9. Plug in the 5V power adapter.

10. Route the cables as shown in the picture below for best stability. You may also use a standard office binder clip or adhesive tape to anchor the cables firmly to the box.

11. Place the pink foam on surface directly below FinBoard.

**IMPORTANT!** FinBoard must be positioned parallel to the viewing surface, with the image sensor pointing down. Failure to do so will distort the sampled dice dots and introduce errors in the resulting dot count.

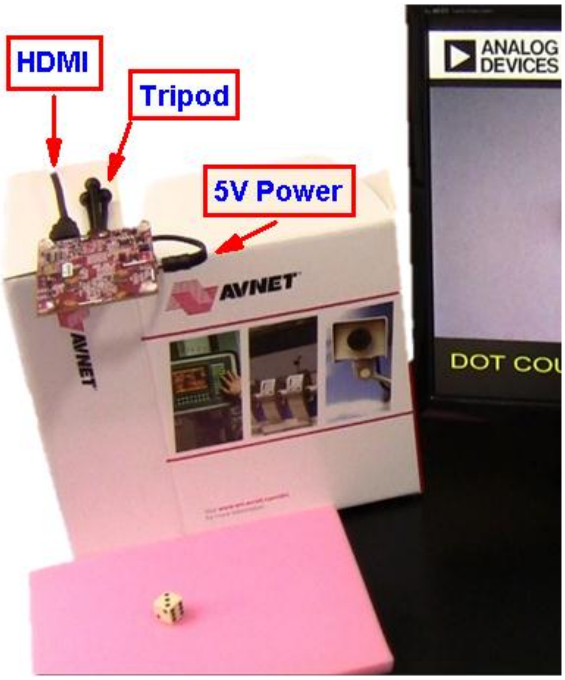


**Figure 8 - FinBoard Demo Hardware Setup (ICE-100B not shown)**

### Software Setup – CCES Workspace Creation and Project Import

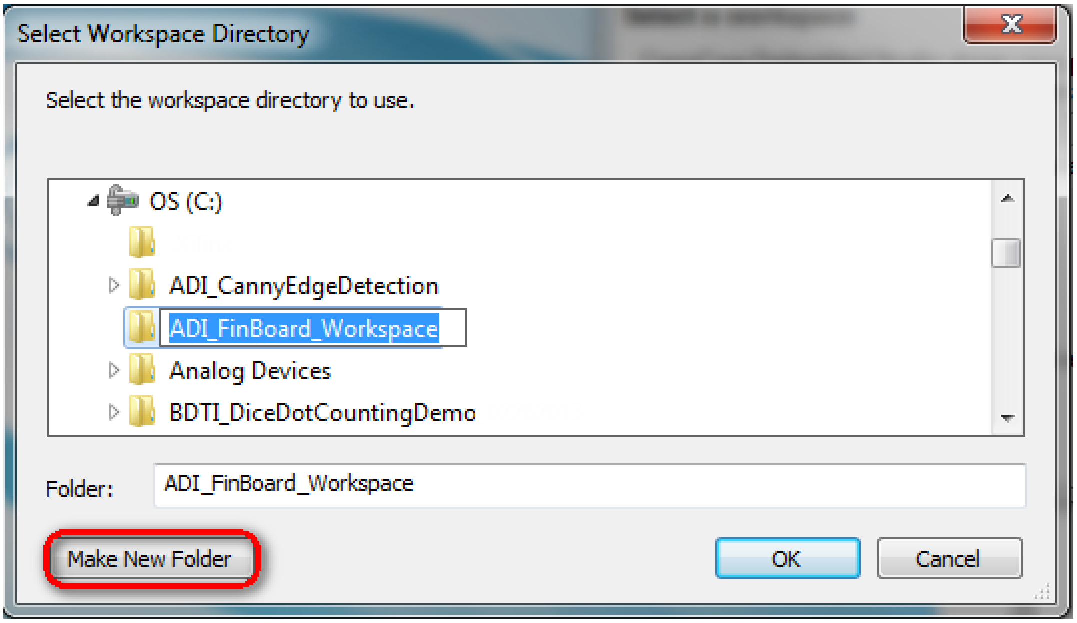
CrossCore Embedded Studio (CCES) uses the concept of a workspace to define user preferences for project development. In this section you will create and customize a workspace. There are several steps, but the process only needs to done once. The next time you launch CCES, you can simply select this workspace and everything will be automatically configured.

Note – if you plan to boot directly from FinBoard's microSD flash card, skip this section and go to the **Loading the Demo** section.

1. Launch CCES.

2. You will be prompted to select a workspace. Click the 'Browse' button.

3. From the "Select Workspace Directory" window, use the **Make New Folder** button to create a folder called "ADI_FinBoard_Workspace" at the root C:\ directory (shown below).



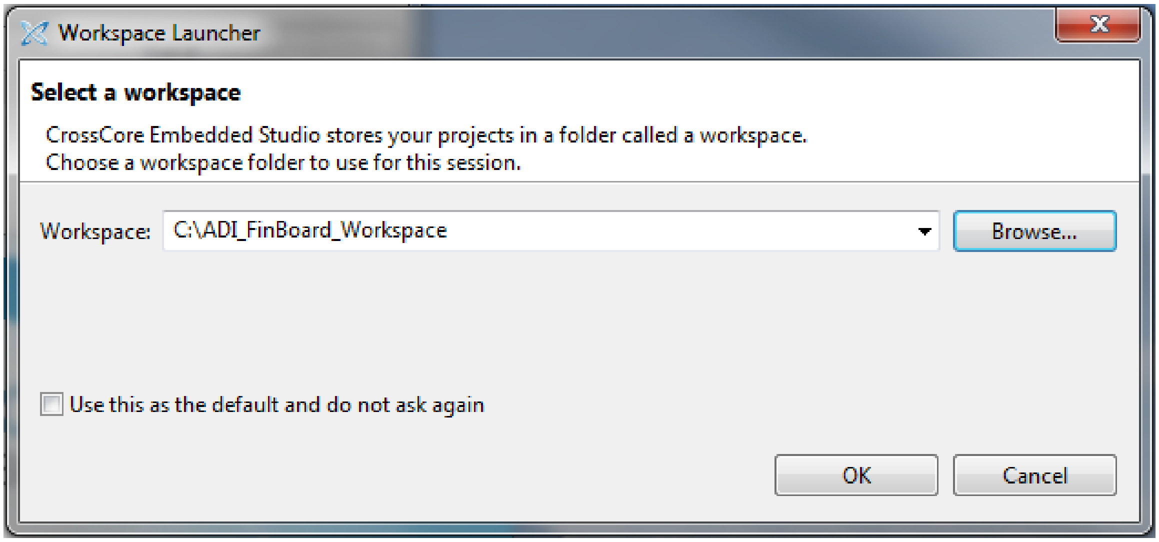
Select Workspace Directory

Select the workspace directory to use.

Folder: ADI_FinBoard_Workspace

4. Click **Ok**.

5. In the Workspace Launcher, ensure that "ADI_FinBoard_Workspace" is selected and is located at the root C:\ directory as shown below. Click **Ok**.



**Select a workspace**

CrossCore Embedded Studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

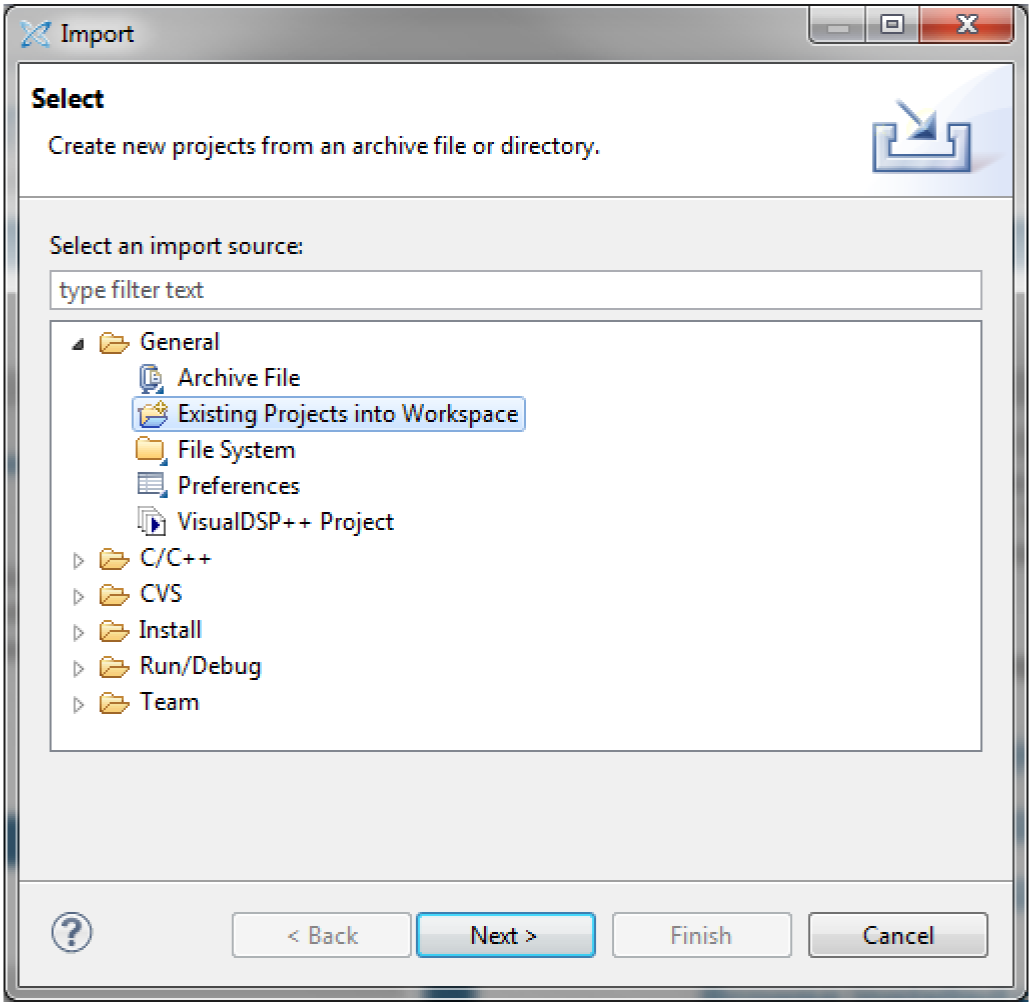
Workspace: C:\ADI_FinBoard_Workspace

□ Use this as the default and do not ask again

6. CCES will now open to the “Welcome” screen. Next you will import the BDTI Dice Dot Counting Demo project files.

7. Select **File → Import …**

8. Select the “Existing Projects into Workspace” option under the “General” folder as shown below, and then click **Next**.



Import

**Select**

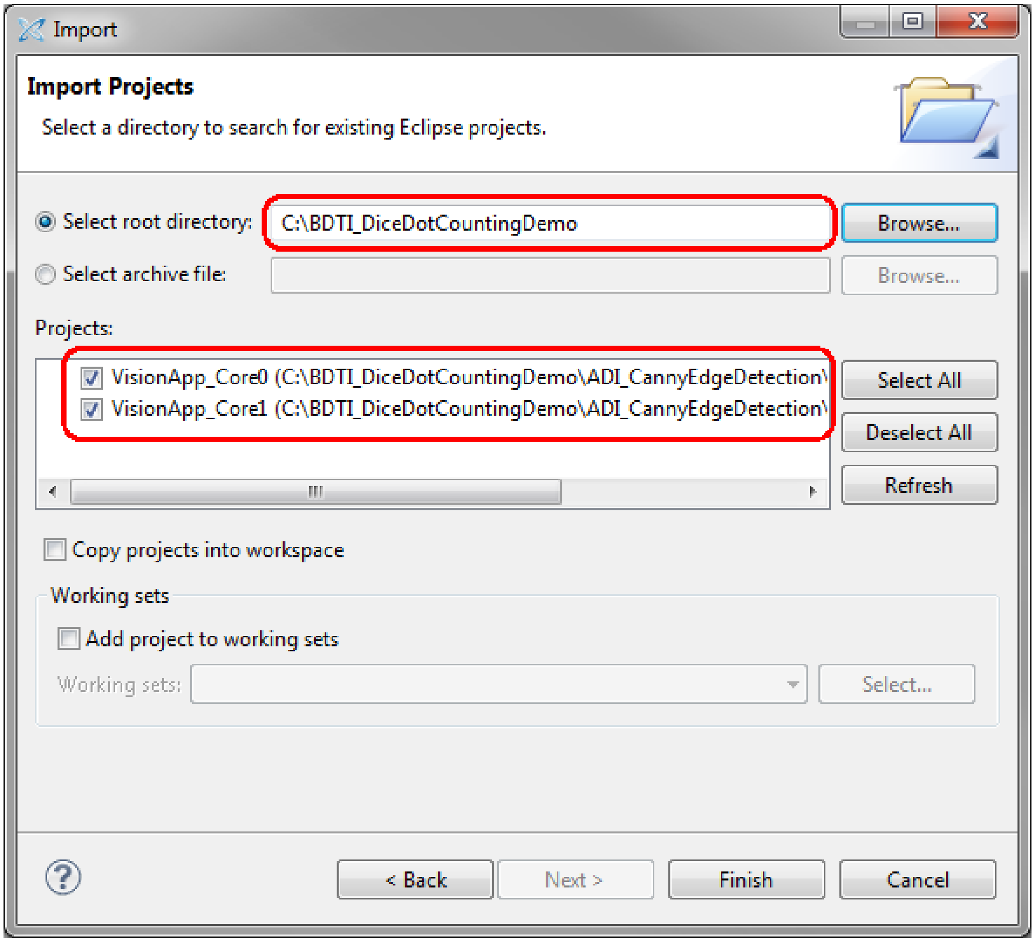
Create new projects from an archive file or directory.

Select an import source:

type filter text

- General
  - Archive File
  - Existing Projects into Workspace
  - File System
  - Preferences
  - VisualDSP++ Project
- C/C++
- CVS
- Install
- Run/Debug
- Team

9. Click the **Browse** button next to “Select root directory:” and navigate to the **BDTI_DiceDotCountingDemo** folder that was unzipped to your root C:\ directory in the **Installing Software** section of this document.



● Select root directory: C:\BDTI_DiceDotCountingDemo

○ Select archive file:

☑ VisionApp_Core0 (C:\BDTI_DiceDotCountingDemo\ADI_CannyEdgeDetection\

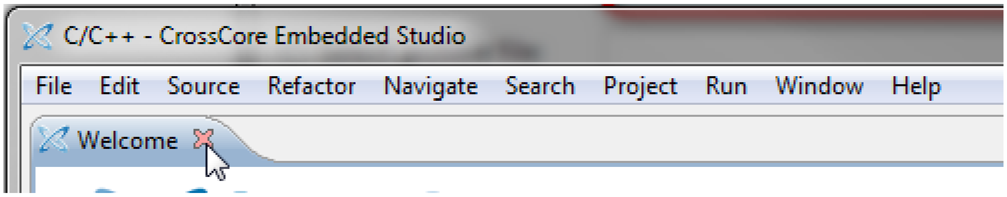
☑ VisionApp_Core1 (C:\BDTI_DiceDotCountingDemo\ADI_CannyEdgeDetection\

□ Copy projects into workspace

□ Add project to working sets

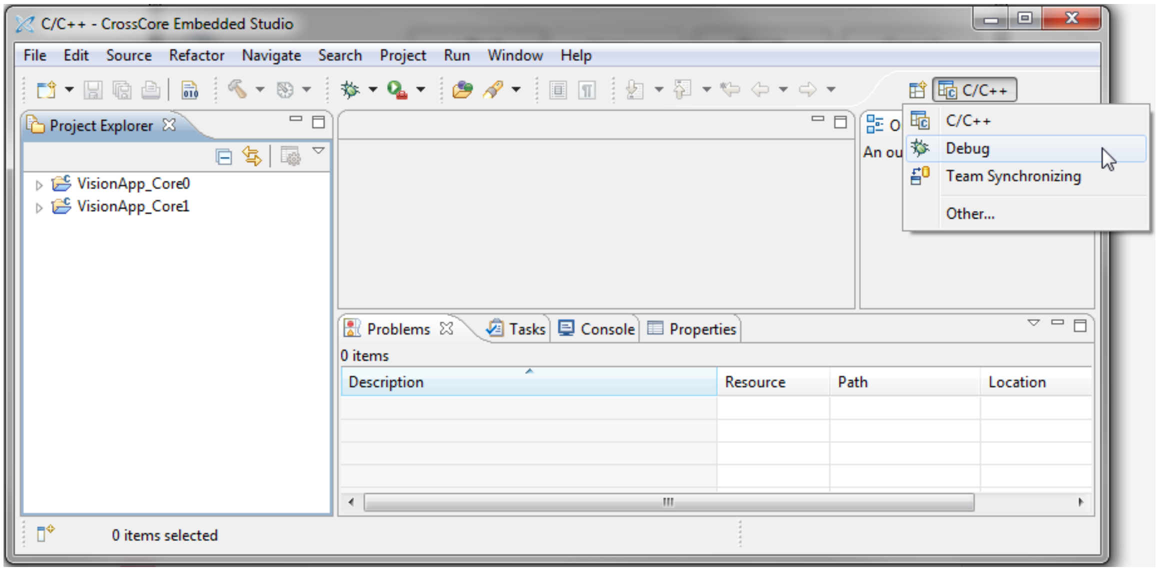
10. This design utilizes both Blackfin cores in the BF609, so ensure that both VisionApp_Core projects are checked. Then click **Finish**.

11. In CCES, close the Welcome tab by clicking on the “x”.



12. Like most Eclipse-based development tools, CCES includes the concept of perspectives to automatically customize the GUI depending on the context of your development stage.

    Add a “debug” perspective by clicking the ⊞ icon next to the C/C++ perspective as shown below.

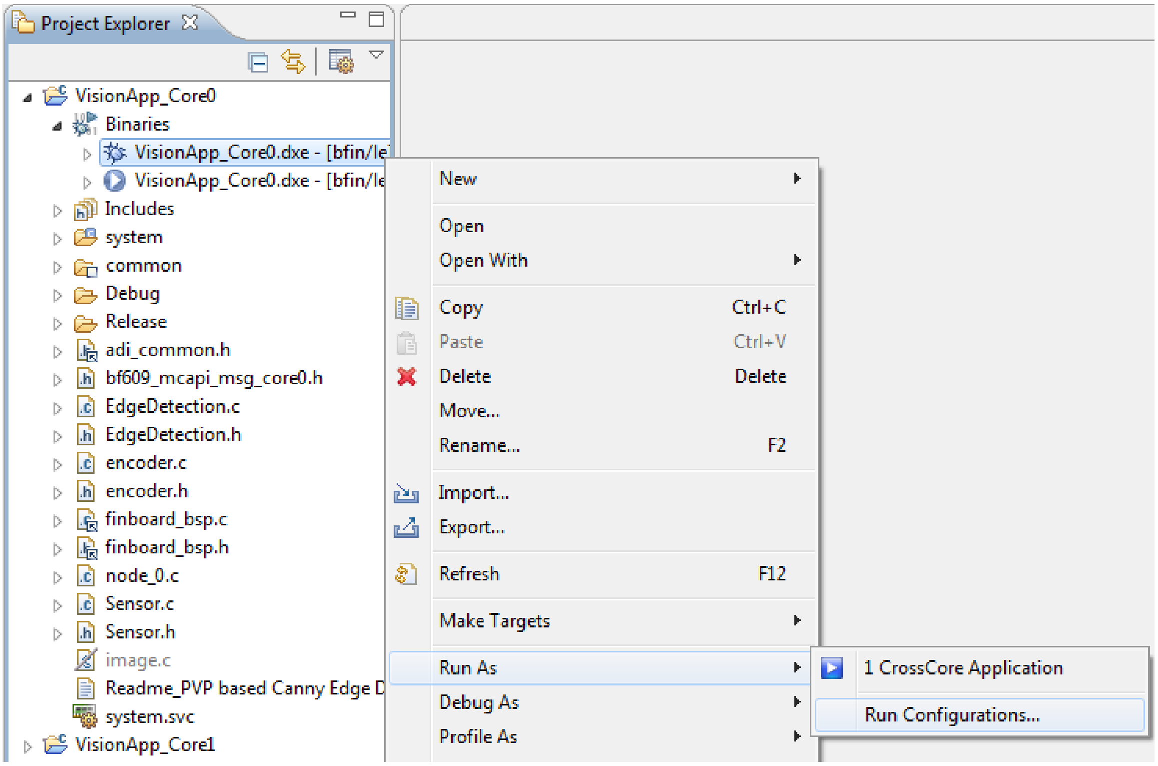


13. Return to the C/C++ perspective by clicking the C/C++ icon.

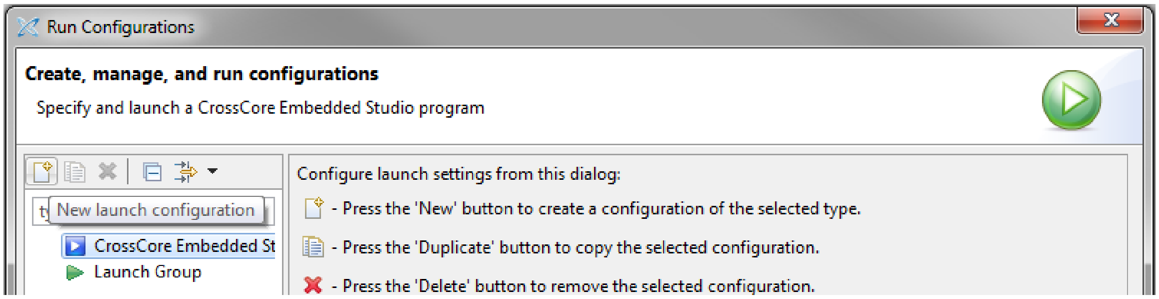
Next, you will configure CCES settings that are specific to building and loading applications to FinBoard.

14. Expand the **VisionApp_Core0** project within the CCES Project Explorer and double-click on "Binaries" to expand the folder.

15. Right-click on the existing **DotCountApp_Core0.dxe** binary file with the debug icon next to it, then select **Run As → Run Configurations …**



16. In the “Run Configurations” window, select “CrossCore Embedded Studio Application”, and then click the New Launch Configuration icon.



**Create, manage, and run configurations**

Specify and launch a CrossCore Embedded Studio program

Configure launch settings from this dialog:

- Press the 'New' button to create a configuration of the selected type.
- Press the 'Duplicate' button to copy the selected configuration.
- Press the 'Delete' button to remove the selected configuration.

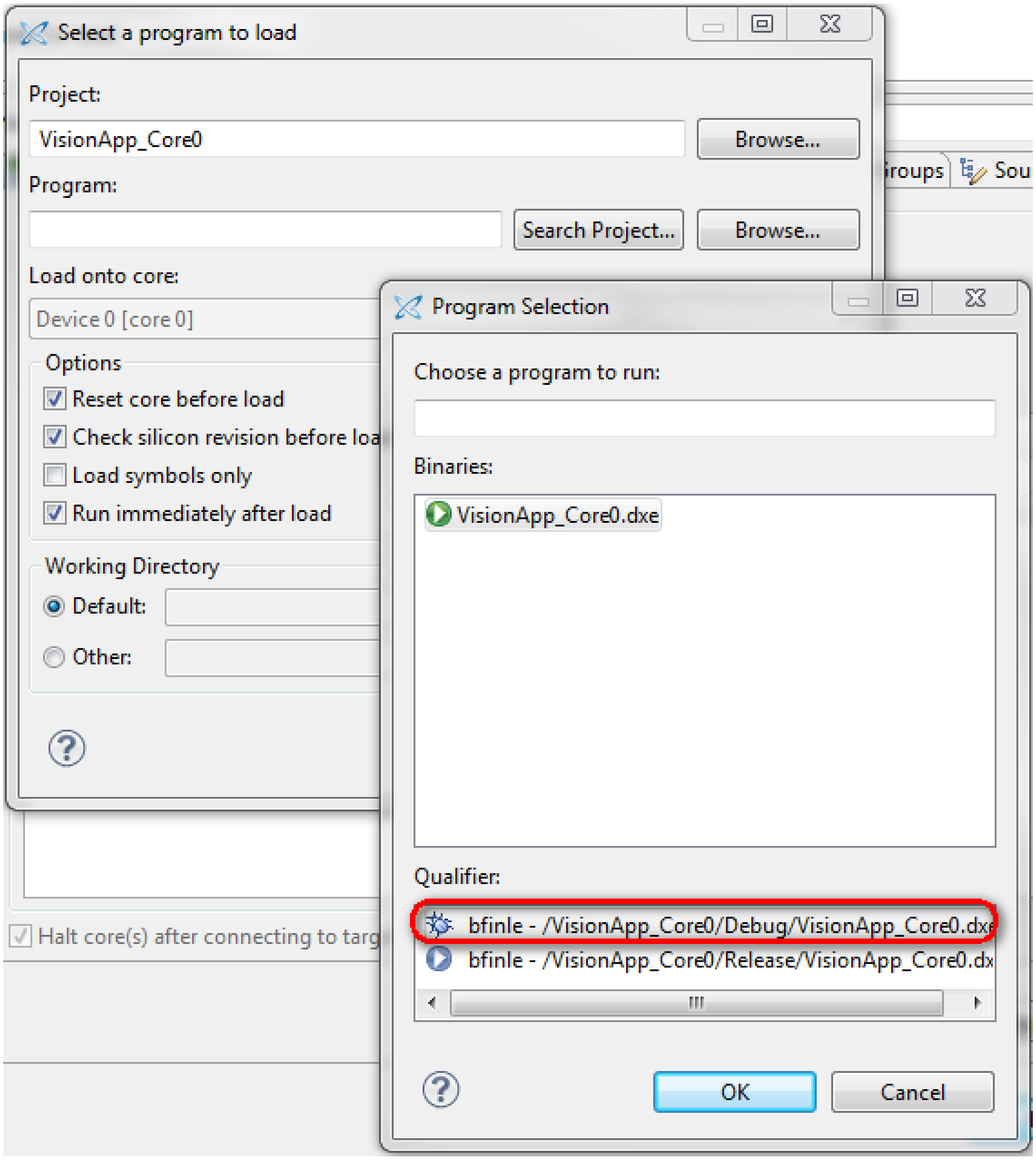
17. In the Session Wizard, make sure “**Blackfin**” is selected as Processor family, then choose **ADSP-BF609** and click **Next**.

18. Select “**Emulator**” as the Connection Type and click **Next**.

19. Select “**ADSP-BF609 via ICE-100B**” as the Platform Type, and then click **Finish.**

20. In the “Programs(s) to load” section of the Session Wizard find **Device 0 [core 0]** and double-click <Click here to select a program to load>.

21. Click the **Search Project** button and select **VisionApp_Core0.dxe** file as shown below.



Select a program to load

Project: VisionApp_Core0

Program:

Load onto core: Device 0 [core 0]

Options

☑ Reset core before load

☑ Check silicon revision before loa

☐ Load symbols only

☑ Run immediately after load

Working Directory

● Default:

○ Other:

☑ Halt core(s) after connecting to targ

Program Selection

Choose a program to run:

Binaries:

VisionApp_Core0.dxe

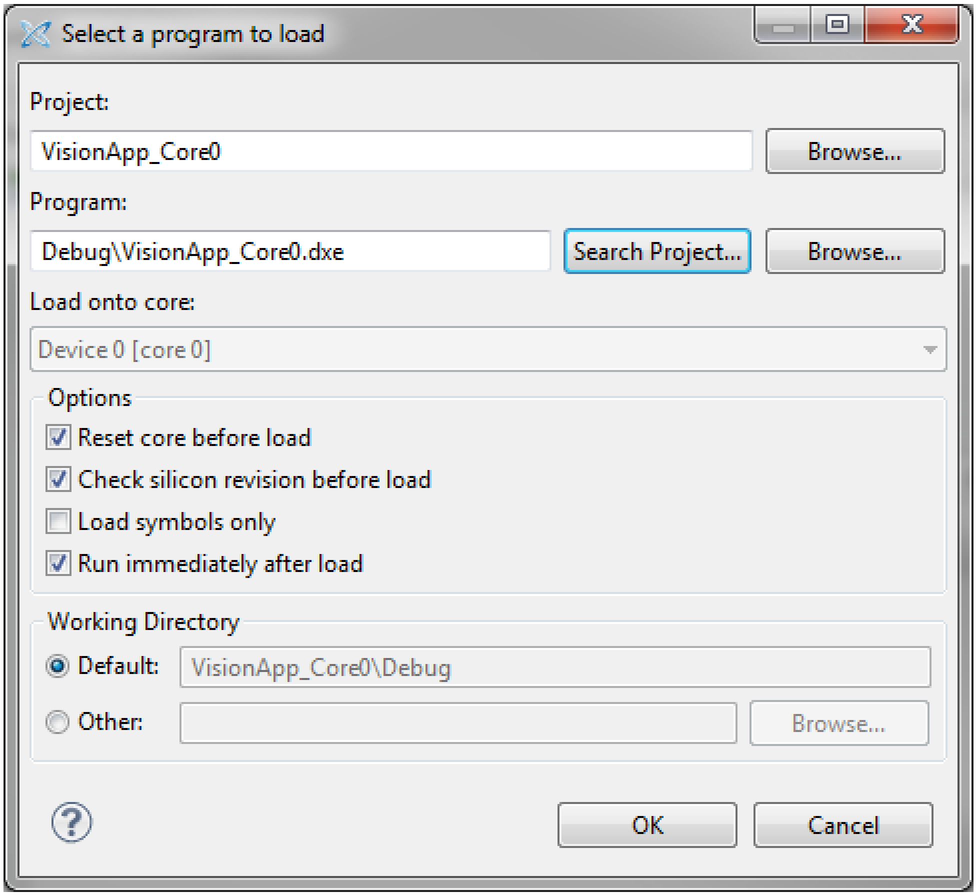
Qualifier:

bfinle - /VisionApp_Core0/Debug/VisionApp_Core0.dxe

bfinle - /VisionApp_Core0/Release/VisionApp_Core0.dx

22. In the “Qualifier:” field, ensure that the “debug” file is selected. This is first file in the list and is identified by the icon next to it. This version of the executable allows code inspection and intuitive features for debugging the application.

23. Click **OK**. Your window should like like the picture below. Click **OK**.



Select a program to load

Project: VisionApp_Core0

Program: Debug\VisionApp_Core0.dxe

Load onto core: Device 0 [core 0]

Options

☑ Reset core before load

☑ Check silicon revision before load

☐ Load symbols only

☑ Run immediately after load

Working Directory

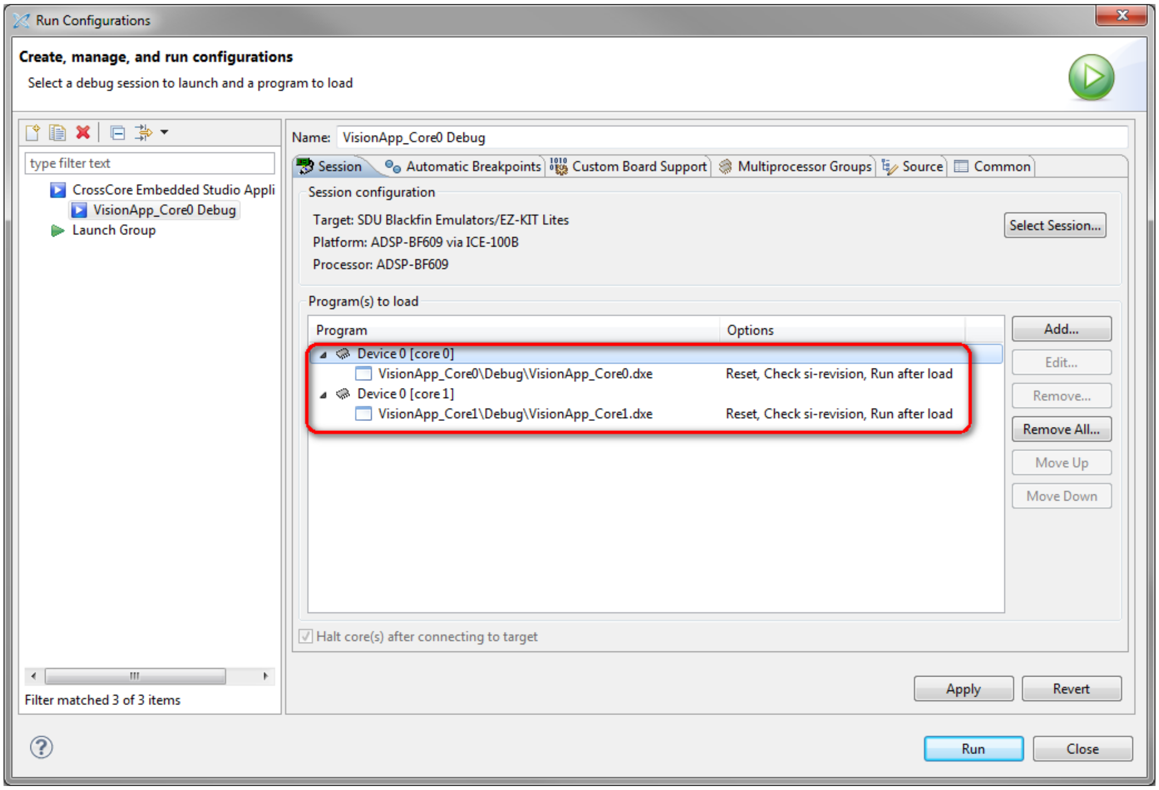
● Default: VisionApp_Core0\Debug

○ Other:

24. Repeat steps 20 – 23 for **Device 0 [core 1]**, this time choosing the **VisionApp_Core1.dxe** executable (debug version) to run on the 2nd Blackfin core.

25. Your Run Configurations should look like the picture below, with .dxe binaries targeted to each Blackfin core (Core0 and Core1).

**Note** – recall that the FinBoard board support package (BSP) is built upon the existing Analog Devices EZ-KIT BSP. That's why EZ-KIT is listed as the Target. This is normal.



**Create, manage, and run configurations**

Select a debug session to launch and a program to load

- CrossCore Embedded Studio Appli
  - VisionApp_Core0 Debug
- Launch Group

Filter matched 3 of 3 items

Name: VisionApp_Core0 Debug

Session configuration

Target: SDU Blackfin Emulators/EZ-KIT Lites
Platform: ADSP-BF609 via ICE-100B
Processor: ADSP-BF609

Program(s) to load

| Program | Options |
|---|---|
| Device 0 [core 0] | |
| VisionApp_Core0\Debug\VisionApp_Core0.dxe | Reset, Check si-revision, Run after load |
| Device 0 [core 1] | |
| VisionApp_Core1\Debug\VisionApp_Core1.dxe | Reset, Check si-revision, Run after load |

☑ Halt core(s) after connecting to target

26. Click **Apply**, and then click **CLOSE**.

27. You have configured your custom workspace for FinBoard and you are now ready to use the Dice Dot Counting Demo project in CCES! The next time you launch CCES, you can simply select this workspace and everything will be automatically configured with these settings.

## Loading the Demo

There are three methods for running the BDTI Dice Dot Counting Demo and Reference Design. They are summarized below, followed by detailed instructions for running each of them.

- Option (A) – automatically boot directly from the pre-loaded microSD flash card.

- Option (B) – download and execute the pre-built program files on FinBoard using CCES and the ICE-100B emulator.

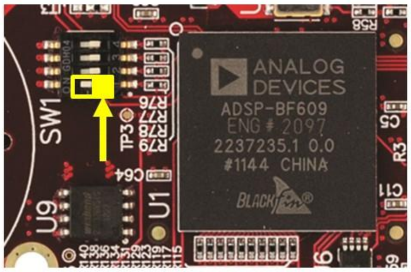
- Option (C) – re-build the application, then download and execute it on FinBoard using CCES and the ICE-100B emulator.

Options (B) and (C) require installation of CCES and all packages, as described in the **Installing Software** section of this document.

## Option (A) – Booting from SD Flash Card

The BDTI Dice Dot Counting Demo comes pre-loaded in the microSD flash card included in the kit. To automatically boot directly from the microSD card when power is applied to FinBoard, follow these simple steps.

1. Insert the microSD card.
2. Set the boot mode select switch SW1-1 to **OFF** (Boot Mode “010” RSIO Master Boot).



3. Attach an HDMI monitor.
4. Attach the 5V power adapter to FinBoard.

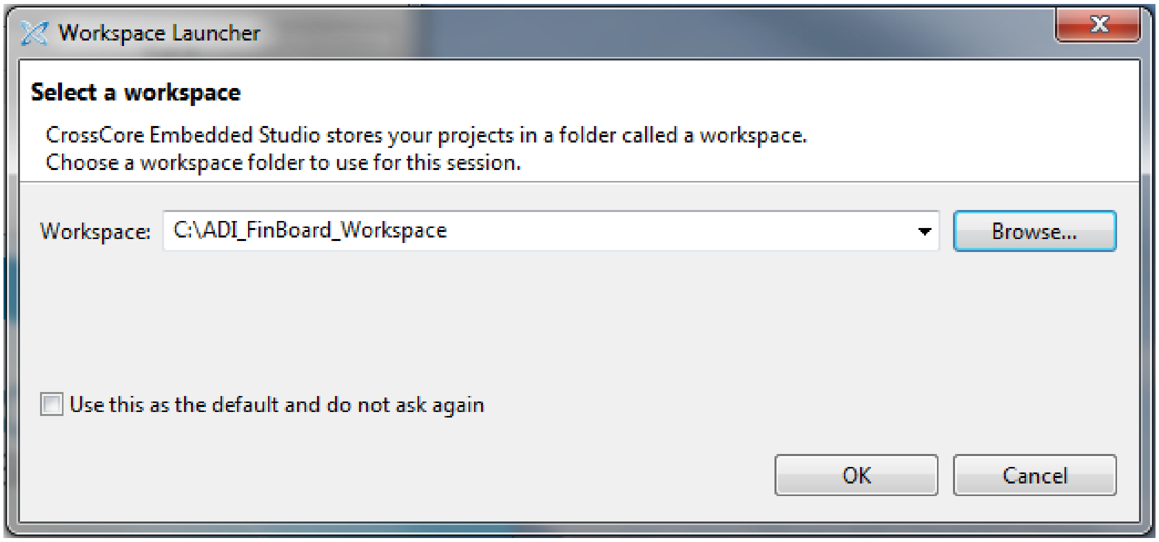
The BDTI Dice Dot Counting Demo loads and executes automatically. Refer to the **Running the Demo** section for further instruction on using this demo.

## Option (B) – Loading the Pre-built Application

Prebuilt executable files (.dxe) are supplied with the BDTI Dice Dot Counting Demo project and can be loaded into the BF609 using CCES and the ICE-100B emulator. The following instructions provide a step-by-step procedure for downloading and executing the pre-built design.

1. Launch CCES.

2. Choose the “ADI_FinBoard_Workspace” workspace that you configured in the **Software Setup – CCES Workspace Creation and Project Import** section of this document.



Workspace Launcher

**Select a workspace**

CrossCore Embedded Studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

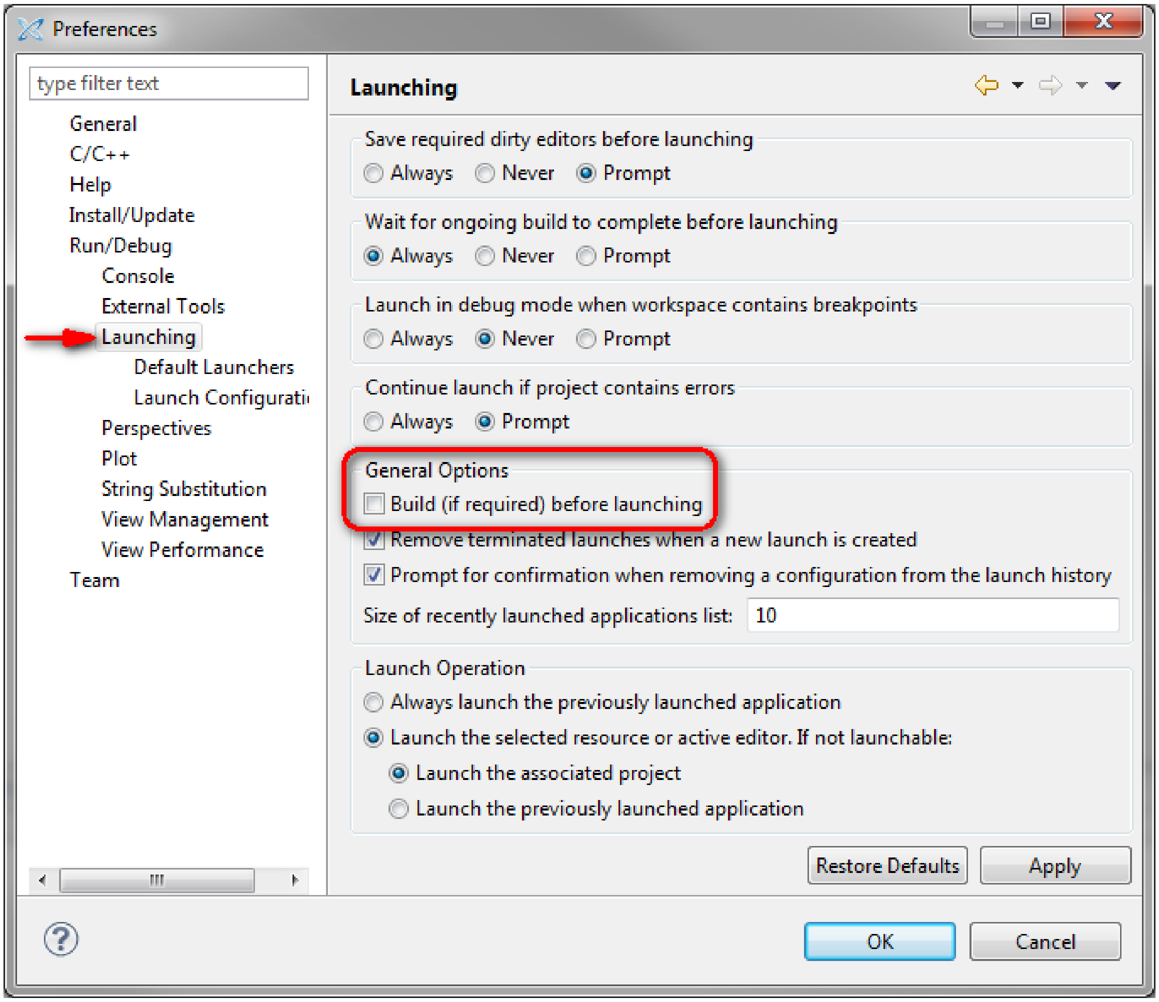
Workspace: C:\ADI_FinBoard_Workspace

□ Use this as the default and do not ask again

3. Click **OK**.

4. In CCES, select **Window → Preferences → Run/Debug → Launching → General Options.**

5. Uncheck “Build (if required) before launching” to avoid over-writing the pre-built .dxe file when launching the debugger.



**Launching**

Save required dirty editors before launching

○ Always ○ Never ● Prompt

Wait for ongoing build to complete before launching

● Always ○ Never ○ Prompt

Launch in debug mode when workspace contains breakpoints

○ Always ● Never ○ Prompt

Continue launch if project contains errors

○ Always ● Prompt

General Options

☐ Build (if required) before launching

☑ Remove terminated launches when a new launch is created

☑ Prompt for confirmation when removing a configuration from the launch history

Size of recently launched applications list: 10

Launch Operation

○ Always launch the previously launched application

● Launch the selected resource or active editor. If not launchable:

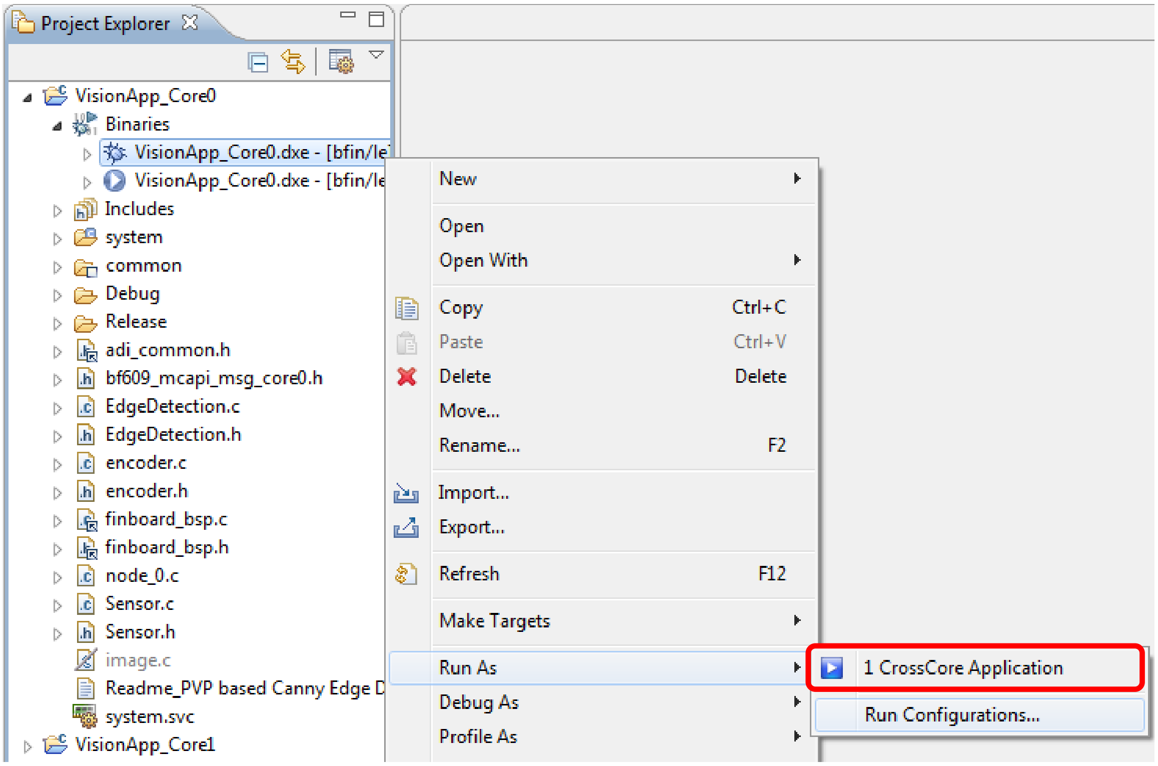
  ● Launch the associated project

  ○ Launch the previously launched application

6. Click **OK**.

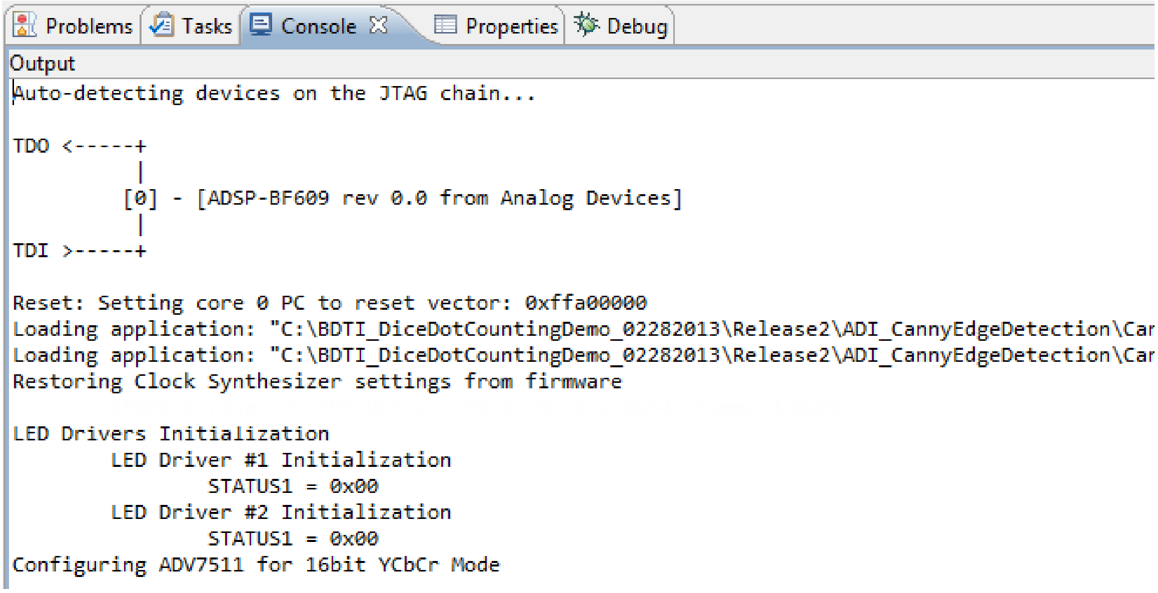
7. Expand the **VisionApp_Core0** project within the CCES Project Explorer and double-click on "Binaries" to expand the folder.

8. Right-click on the existing **DotCountApp_Core0.dxe** binary file with the debug icon next to it, then select **Run As → CrossCore Application**



9. The binary executable files will be loaded to FinBoard and executed automatically.

10. In CCES, select the Console tab near the bottom to view the load status (shown below).



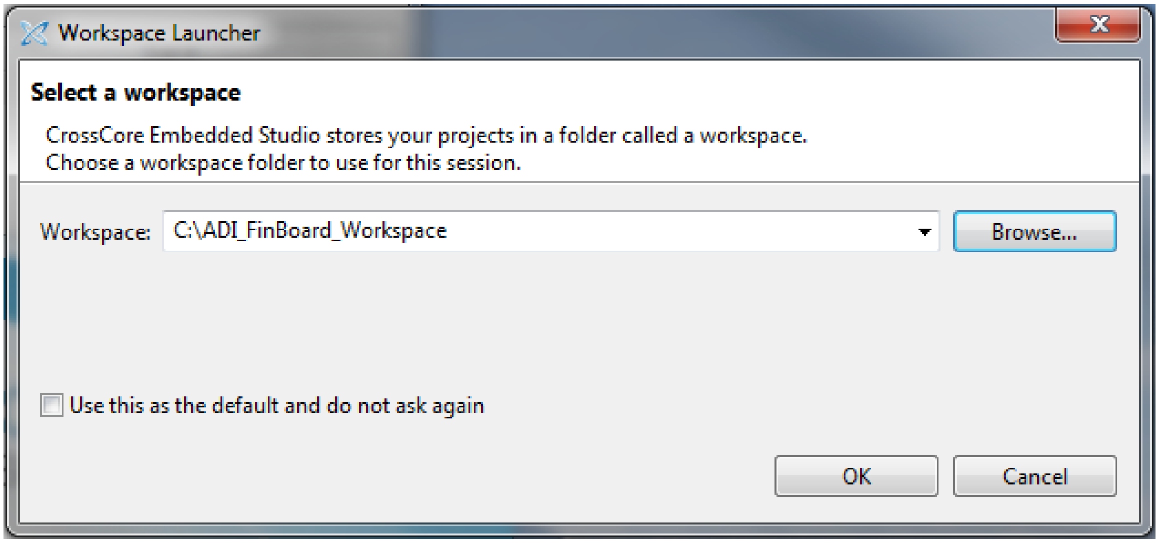
```
Auto-detecting devices on the JTAG chain...

TDO <-----+
          |
         [0] - [ADSP-BF609 rev 0.0 from Analog Devices]
          |
TDI >-----+

Reset: Setting core 0 PC to reset vector: 0xffa00000
Loading application: "C:\BDTI_DiceDotCountingDemo_02282013\Release2\ADI_CannyEdgeDetection\Car
Loading application: "C:\BDTI_DiceDotCountingDemo_02282013\Release2\ADI_CannyEdgeDetection\Car
Restoring Clock Synthesizer settings from firmware

LED Drivers Initialization
        LED Driver #1 Initialization
                STATUS1 = 0x00
        LED Driver #2 Initialization
                STATUS1 = 0x00
Configuring ADV7511 for 16bit YCbCr Mode
```

11. The application should now be displaying video on the HDMI monitor. Refer to the **Running the Demo** section for further instruction on using this demo.

## Option (C) – Building and Loading the Application

The BDTI Dice Dot Counting Demo can be re-built in CCES and downloaded to FinBoard using the ICE-100B emulator by following these steps.

1. Launch CCES.
2. Choose the “ADI_FinBoard_Workspace” workspace that you configured in the **Software Setup – CCES Workspace Creation and Project Import** section of this document.



Workspace Launcher

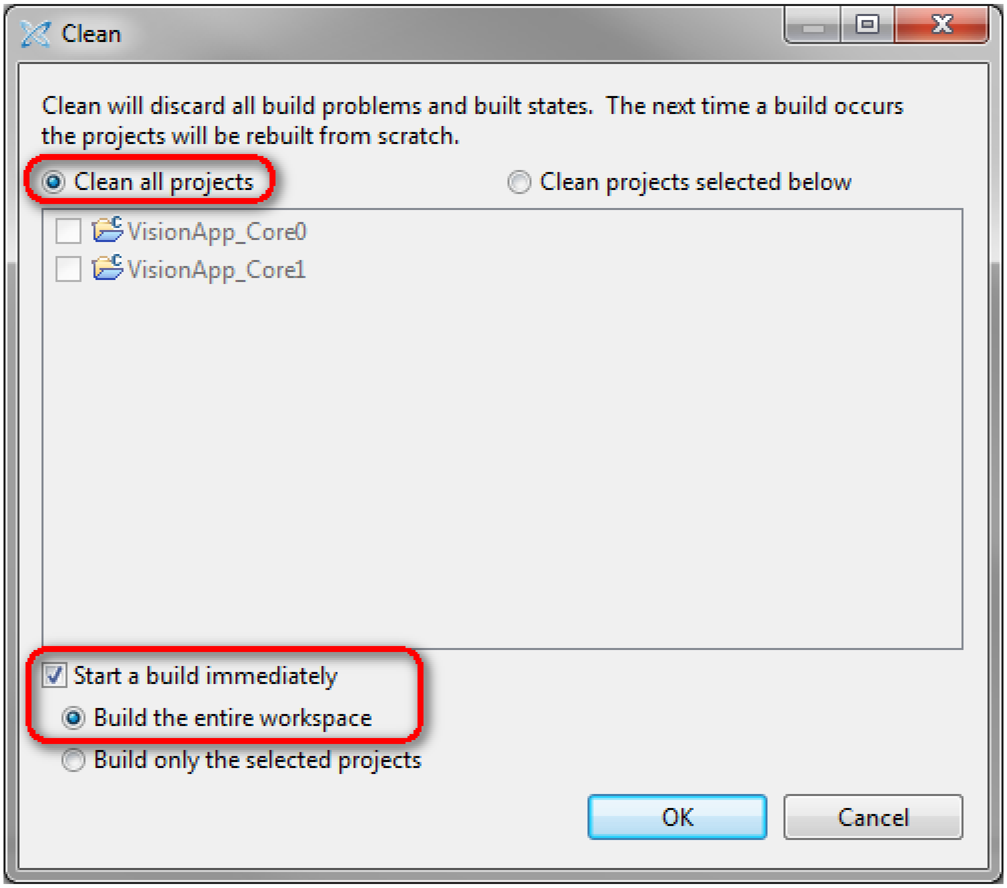
**Select a workspace**

CrossCore Embedded Studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\ADI_FinBoard_Workspace

□ Use this as the default and do not ask again

3. Click **OK**.
4. In CCES, select **Project → Clean …**
5. Select **Clean all projects** and ensure the box is checked for **Start a build immediately** as shown below.



Clean will discard all build problems and built states. The next time a build occurs the projects will be rebuilt from scratch.

● Clean all projects ○ Clean projects selected below

□ VisionApp_Core0
□ VisionApp_Core1

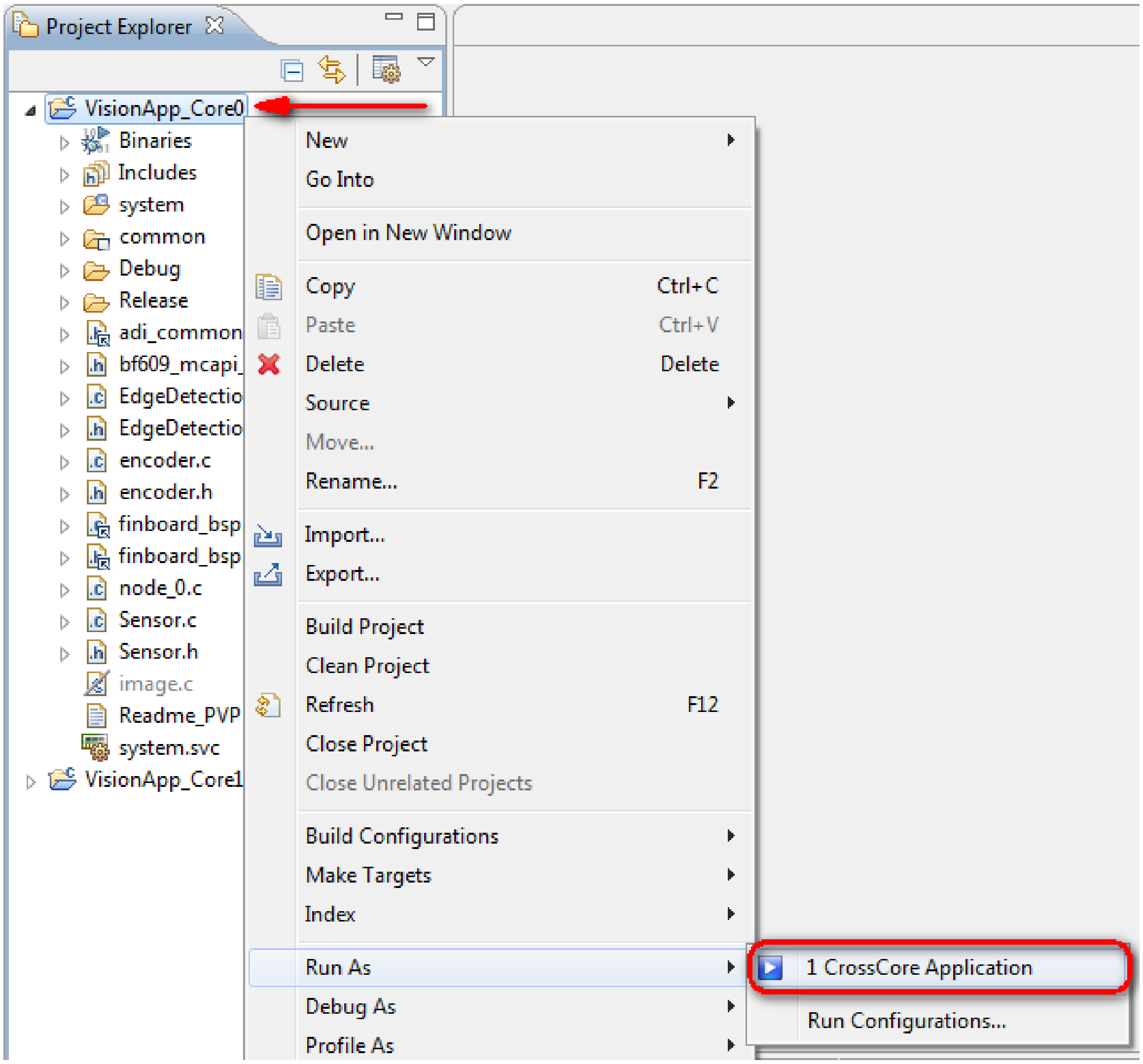
☑ Start a build immediately
● Build the entire workspace
○ Build only the selected projects

**Note** – if you later add other projects to this workspace, you will likely not want to re-build all projects. In that case, select the ”Clean projects selected below” radio button, and select the “Build only the selected projects” radio button.

6. Click **OK**. The project will be cleaned and re-built.

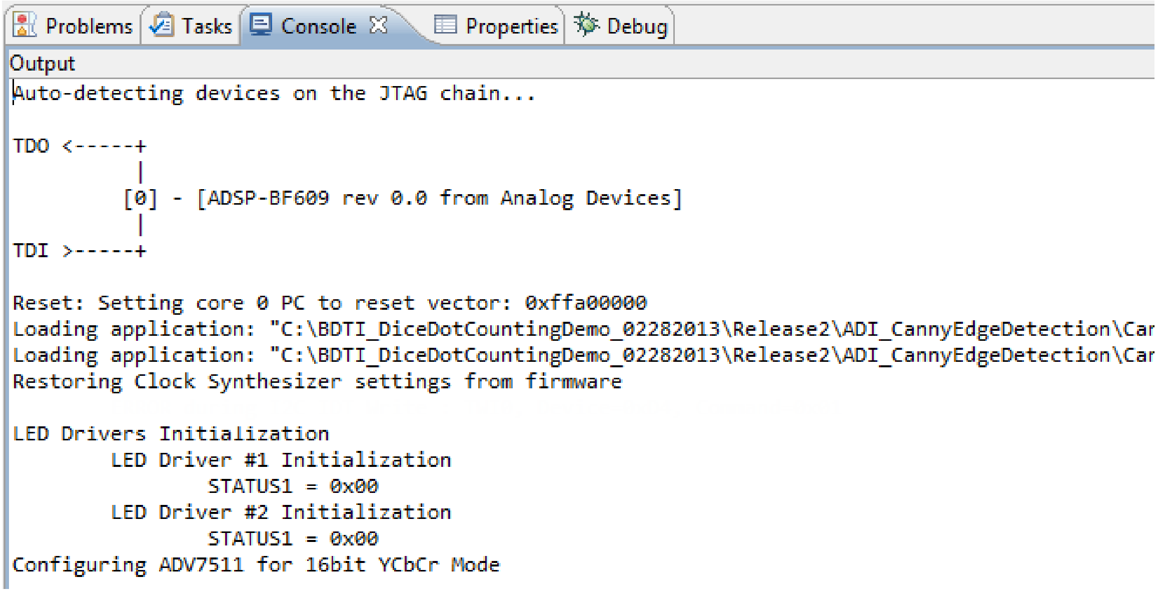
7. When the build process completes, click the “Problems” tab near the bottom of CCES and ensure that no errors were encountered. If errors are listed, see the **Troubleshooting** section of this document for suggestions on debugging the issue.

8. In the Project Explorer, right-click on “VisionApp_Core0” and select **Run As → CrossCore Application**



9. The newly compiled binary executable files will be loaded to FinBoard and executed automatically.

10. In CCES, select the Console tab near the bottom to view the load status (shown below).



```
Auto-detecting devices on the JTAG chain...

TDO <-----+
          |
         [0] - [ADSP-BF609 rev 0.0 from Analog Devices]
          |
TDI >-----+

Reset: Setting core 0 PC to reset vector: 0xffa00000
Loading application: "C:\BDTI_DiceDotCountingDemo_02282013\Release2\ADI_CannyEdgeDetection\Car
Loading application: "C:\BDTI_DiceDotCountingDemo_02282013\Release2\ADI_CannyEdgeDetection\Car
Restoring Clock Synthesizer settings from firmware

LED Drivers Initialization
        LED Driver #1 Initialization
                STATUS1 = 0x00
        LED Driver #2 Initialization
                STATUS1 = 0x00
Configuring ADV7511 for 16bit YCbCr Mode
```

11. The application should now be displaying video on the HDMI monitor. Refer to the **Running the Demo** section for further instruction on using this demo.

## Running the Demo

Once you have the hardware set up and the the BDTI Dice Dot Counting Demo loaded and running on FinBoard, you are ready to use the application. You just need to focus the lens and select the demo mode. Follow these instructions.

### Operating Modes

The application has several operating modes. Set them as follows:

1. Set the High Brightness LEDs to the maximum illumination, by pressing SW4 on FinBoard multiple times. There are eight settings, including OFF.

2. Demo mode #1 – press SW3 once to select the **Find Contours** mode.

3. Demo mode #2 – press SW3 again to select **Edge Trace and Dot Fill** mode.

**Note** – pressing SW2 will reset the BF609 processor and the application will need to be reloaded.

For more information on the modes of operation, please download "**BDTI Dice Dot Counting Demo and Reference Design Software User Guide**" at FinBoard.org.

### Focusing the Lens – IMPORTANT!

The BDTI Dice Dot Counting Demo and Reference Design require the lens to be properly focused on the dice in order to correctly detect and count the dots. Once you have the application loaded and running on FinBoard, focusing the lens is straightforward. Simply rotate the lens until the viewing surface comes into focus on the HDMI monitor.

The lens is very small and can be difficult to rotate with your fingers.

1. Use the lens tool that is provided with the kit, taking care not to scratch the lens with the metal pins (**Figure 9**). Alternatively use a piece of adhesive tape commonly found in an office supply closet. Place the tape on your index finger; push it against the lens, and twist.

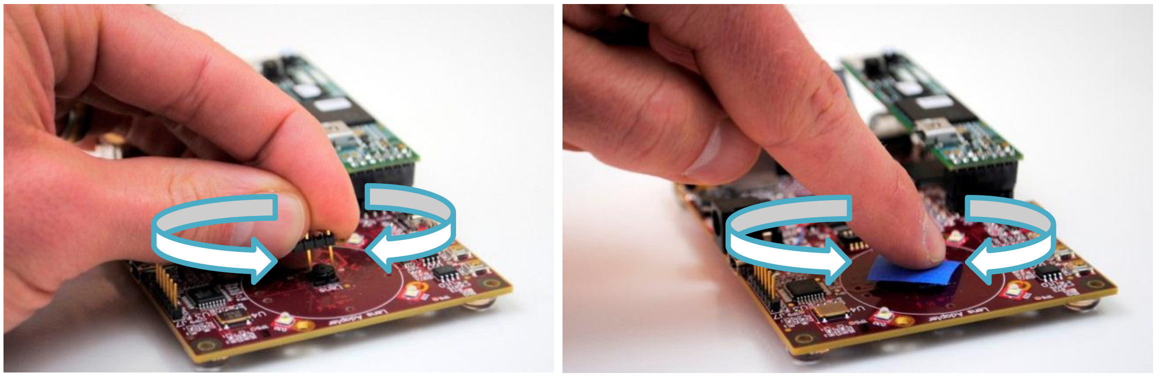


**Figure 9 - Focus with lens tool or adhesive tape**

2. Place the Quick Start Card or Welcome Letter included with this kit within the field of view to provide a detailed target for focusing the lens. Remove the printed sheet once you have completed focusing the lens.

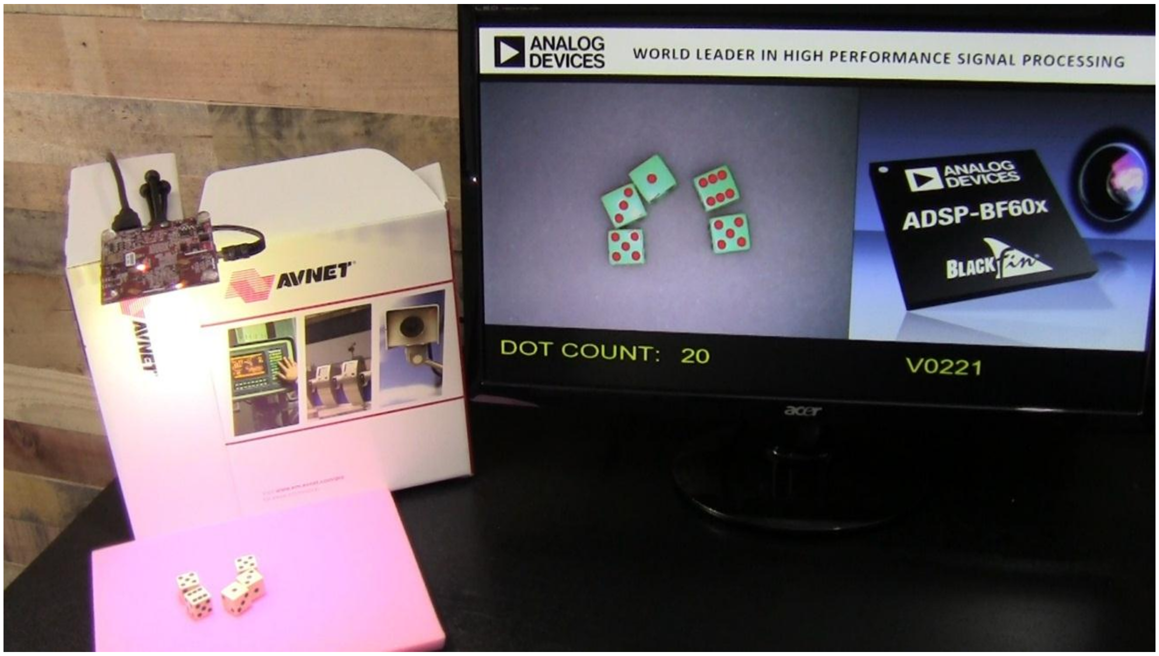


**Figure 10 - Assembled FinBoard Kit Running the BDTI Dice Dot Counting Demo**

3. Place the 5 dice (included with the kit) within the field of view and observe the application automatically display the dice dot count on your monitor.

**Congratulations!** You have successfully run the BDTI Dice Dot Counting Demo and Reference Design. If you encounter any issues please review the **Known Issues and Limitations** and **Troubleshooting** sections of this document. Also be sure to check for answers on the support forums at www.FinBoard.org. In addition, you will find other reference designs, tutorials, and community projects for FinBoard like the Canny Edge Detection Example.

# Canny Edge Detection Example

An example project that demonstrates the popular edge detection algorithm known as Canny Edge Detection may be downloaded from www.FinBoard.org. The BDTI Dice Dot Counting Demo is a software application built upon this Canny Edge Detection design, so the hardware and software setup steps are nearly identical. To learn more about how these two projects are related please download "**BDTI Dice Dot Counting Demo and Reference Design Software User Guide**" from the FinBoard.org website.

Once you have downloaded and unzipped the Canny Edge Detection project archive, follow the instructions in the **BDTI Dice Dot Counting** Demo section of this document in order to build, load and execute the project on FinBoard.  Note that **Option (A) – Booting from SD Flash Card** will not be applicable.

# References

## FinBoard

All documentation and support for FinBoard are located at the product website www.FinBoard.org. There you will find the following items:

- Design tutorials
- BDTI Dice Dot Counting Demo and Reference Design Software User's Guide
- Hardware User's Guide
- Hardware schematics & bill of materials
- Technical support forum

## Analog Devices

For more information on the Analog Devices parts, please visit the following resources:

- BF609 – Blackfin Dual-Core Processor – http://www.analog.com/bf609
- ADV7511 – HDMI Transmitter – http://www.analog.com/adv7511

## Aptina

For more information on the Aptina image sensor, please visit the following resources:

- MT9M114 : 1.3MP/ 720pHD 1/6-Inch SOC Image Sensor
  http://www.aptina.com/products/soc/mt9m114/

# Known Issues and Limitations

The following issues are known to exist. When applicable, the workaround is described.

## Application Shows “0” For Large Number of Dots

In the BDTI Dice Dot Counting Demo and Reference Design, memory has been reserved for a maximum of 6000 RLE nodes. RLE stands for Run Length Encoded. Each RLE node represents a unit of horizontal continuous edges. Imagine you have 5 continuous edge pixels on a horizontal line, starting at an X value of 100. That would be encoded by a single RLE node starting at 100 with a length of 5. Any video frame which has more than 6000 RLE nodes will be rejected and will not be processed further. You may see “DOT COUNT” as “0” when the number of RLE nodes exceeds 6000. This will typically occur when the dot count exceeds ~100. The restriction can be changed by redefining the macro “MAX_NUM_RUN_LEN_NODE” located at “adi_contours.h”.

# Troubleshooting

## Dice Dot Counting Demo not loading at power up

Verify that switch SW1-1 is in the OFF position (Boot Mode "010" RSIO Master Boot), which configures the BF609 to boot from the FinBoard microSD flash card interface.

## Dice count result is incorrect or unstable

The BDTI Dice Dot Counting Demo and Reference Design software parameters have been tuned specifically for the setup described in this guide. It is important that the kit is set up as shown in order to achieve robust operation of the application. If you are observing incorrect or unstable dot counts, please review the hardware and software setup sections. Several possible issues are proposed below with suggested solutions.

| **Possible Issue** | **Suggestion** |
|---|---|
| Out of focus | See **Focusing the Lens** – IMPORTANT! |
| Scratched lens | Submit request on FinBoard.org forum |
| FinBoard not parallel to the viewing surface | Reposition FinBoard |
| Not enough illumination | Set HB LEDs to maxim illumination |
| Pink foam not used for viewing surface | Use pink foam provided with kit |
| Dice positioned near or outside the field-of-view (FOV) | Move dice towards the FOV center |
| Reflective objects in the FOV | Remove any reflective objects |

## Could not open source file "adi_graphics.h"

Though uncommon, when the VisionApp is built the first time, the following error may occur with the following alert in the CCES console:

```
… could not open source file "adi_graphics.h"
#include "adi_graphics.h"
                          ^
```

```
1 catastrophic error detected in the compilation of “…Graphics.c”
```

The solution to this issue is to simply clean the project, then re-build as follows:

1. In the CCES Project Explorer, select the **VisionApp_Core0** project
2. Right-click and select **Clean Project**.
3. Right-click and select **Build Project**