ADSP-21469 EZ-Board™Board Support Package (BSP) v1.0.0 Release Notes

Thank you for installing the ADSP-21469 EZ-Board Board Support Package (BSP). The BSP is designed for use with CrossCore® Embedded Studio (CCES) for Analog Devices Processors software development tools release 1.0.1 or later. The CCES development environment aids advanced application code development and debug, such as:

- Create, compile, assemble, and link application programs written in C++, C, and assembly
- Load, run, step, halt, and set breakpoints in application programs
- Read and write data and program memory
- Read and write core and peripheral registers
- Plot memory

For more details on CCES, please visit www.analog.com/cces. For more on the ADSP-21469 EZ-Board, please visit www.analog.com/21469EZBoard.
The CCES Help environment provides complete hardware and software documentation.

License Checking

There are no license requirements for the BSP.

Installation Logging

The installer does not create a log file by default. If you encounter installation issues, you can generate an installation log file by running the installer from the command prompt.
Change to the directory containing downloaded installer executable and run the following from the command prompt:

```
ADI_ADSP-21469_EZKIT-Rel1.0.0.exe /v"/l*v c:\temp\installer.log"
```

Support and Assistance

There are several options for contacting support:

- Submit your questions online at
  - http://www.analog.com/support.
- E-mail your Processor and DSP software and development tools questions from within CCES located under the Help menu:
  - processor.tools.support@analog.com.
- E-mail your Processors and DSP applications and processor questions to:
  - processor.support@analog.com OR

    - processor.china@analog.com (Greater China support)
- Post your questions in the Processors and DSP online technical support community in Engineer Zone at:
    - http://ez.analog.com/community/dsp.

Software Requirements

To build the projects included in the BSP, CCES version 1.0.1 or later is required.

Getting Started

Creating a project

In order to create a project you should follow the instructions provided in the CCES help. As part of the project creation, the page "Add-in selection" contains a list of all the available add-ins for the project that you are creating based on the installed products and the project's chosen processor and type.
The ADSP-21469 EZ-Board add-in generates a call to adi_initComponents(). For more information on adi_initComponents(), please refer to the CCES help section: CrossCore® Embedded Studio 1.0.1 > Integrated Development Environment > System Configuration
Notes:

- If the IDE detects that adi_initComponents() is not yet present in main(), it prompts you to add it and offers to insert it for you.

Examples

- Power_On_Self_Test example includes a test for each peripheral on the EZ-Board in order to make sure that the board is functioning correctly.
- Parallel flash programmer interface application used with the device programmer to access the M29W320 parallel flash device on the EZ-Board.
- Serial flash programmer interface application used with the device programmer to access the M25P16 SPI flash device on the EZ-Board.

The flash programmer interface applications are used in conjunction with the Command Line Device Programmer application (cldp.exe) in the system sub-directory of the CCES installation. Cldp can be executed standalone from a Windows command box or as a post build step from within the CCES project. For additional information, in CCES go to Help > Search and enter "Device Programmer".

Location

In order to locate the BSP examples, you can use the following:

- Open CCES Example Browser which can be found in CCES under Help. Select in the Product section ADSP-21469 EZ-Board v1.0.0 for a full list of examples.
- Import projects located in your BSP installation folder under the example directory in product installation "ADSP-21469_EZKIT-Rel1.0.0\21469_EZ-Board\Examples".

Documentation

CCES provides a coherent interrupt management mechanism which allows for the same interface to be used in RTOS and non-RTOS applications. This means that interrupt service routines in all applications must be written in C and use the adi_int interface. Any thread-safety requirements or interactions with tasks are handled by the adi_int interface. For more information on the adi_int API, in CCES go to Help > Search and enter adi_int. General information on the interrupt model can be found in CCES help under:

CrossCore® Embedded Studio 1.0.1 > System Run-Time Documentation > Interrupt Support

Known issues with this BSP

None