



**Release Notes for EV-2156x EZ-KIT[®] Rel.
1.0.0**

Contents

1	Release Notes	4
2	Release Dependencies	5
3	Examples:	6
3.1	Power_On_Self_Test:	6
3.2	Device_Programmer:	6
3.3	Device Drivers examples:	6
3.4	System Services examples:	7
4	Known Issues:	8
4.1	POST	8
4.2	Device Programmer	8
4.3	Device Drivers and Services Examples	8
4.3.1	GPIO:	8
4.3.2	ADC:	8

EV-2156x EZ-KIT® Board Support Package v1.0.0

1 Release Notes

Thank you for installing the EV-2156x EZ-KIT® Board Support Package (BSP). The BSP provides software and documentation in support of the EV-21569 EZ-KIT®.

The EV-21569 EZ-Kit is designed for use with CrossCore® Embedded Studio (CCES) for Analog Devices Processors software development tools. 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.

The EV-2156x EZ-KIT® BSP provides comprehensive software support for the EV-21569 EZ-KIT. In this release, various examples are provided to demonstrate the on-chip drivers and services.

2 Release Dependencies

- Requires CrossCore® Embedded Studio version 2.9.3
- EV-21569-SOM REV E
- EV-SOM-CRR EZ-KIT® REV A

3 Examples:

3.1 Power_On_Self_Test:

This example allows the user to test many peripherals of the EV-21569 EZ-KIT®. `Readme_Power_On_Self_Test_21569.html` is provided in the POST example to understand how these tests are run.

3.2 Device_Programmer:

This example allows the user to program the flash device on the EV-21569 EZ-KIT® in conjunction with the "Command-Line Device Programmer (cldp)".

A pre-built binary exists so that users can just program the flash device without having to build the example.

3.3 Device Drivers examples:

Examples are provided for following peripherals-

- ADC
- Asynchronous Sampling Rate Converter (ASRC)
- Cyclic Redundancy Check (CRC)
- Security Packet Engine (PKTE)
- FIR Accelerator
- IIR Accelerator
- LinkPort (LP)
- Octal SPI (OSPI)
- Sony/Philips Digital Interface (S/PDIF)
- Serial Peripheral Interface (SPI)
- Serial Port (SPORT)
- Thermal Monitoring Unit (TMU)
- Two-Wired Interface (TWI)
- Universal Asynchronous Receiver Transmitter (UART)

3.4 System Services examples:

Examples are provided for following peripherals-

- Enhanced Memory DMA (EMDMA)
- General Purpose Ports (GPIO)
- Memory DMA (MDMA)
- Clock Generation Unit (CGU/PWR)
- Reset Control Unit (RCU)
- System Memory Protection Unit (SMPU)
- Standard I/O (STDIO)
- General Purpose Timer (TMR)
- Watchdog Timer (WDOG)

4 Known Issues:

4.1 POST

1. `adi_post_flash2_test` does not work in the standard loop but works fine in standalone mode. So to test the Macronix flash relying on this test in standalone mode is recommended
2. `adi_post_rotary_test` has been excluded from standard loop to conflict on pin-mux with Link Port. To test Rotary switch, select the test in stand alone mode and ensure that Link Port cable is not connected.

4.2 Device Programmer

1. Device Programmer example does not work in release mode.
2. SPI Fast Mode is required to be enabled if SPI clock is configured to run at more than 45.875 MHz.

4.3 Device Drivers and Services Examples

4.3.1 GPIO:

- GPIO examples do not run as expected in release mode.

4.3.2 ADC:

- Sometimes noise is observed when `SPDIF_ASRC_DAC_AudioPassthrough` example is ran across EZ-KITs of different revisions. Provide a hardware reset/power-up and reload the application to avoid the noise.