

Debug Application on AM243x/AM64x Device With SEGGER J-Link



Tushar Thakur, Anil Swargam, Michael Gross, Matthias Konegen

Abstract

This application note describes how to set up and use the SEGGER J-Link debug probe with Texas Instruments AM243x/AM64x devices for loading and debugging embedded applications. The document covers the complete workflow, beginning with required software installations — including the AM243x/AM64x MCU+ SDK (v12.00+), Code Composer Studio, SysConfig, TI-ARM-CLANG compiler, SEGGER J-Link software package, and the Ozone debugger. It then details the physical hardware requirements and step-by-step connection procedure between the J-Link probe and the AM243x/AM64x EVM JTAG connector, including the correct power sequencing to avoid enumeration issues. Connection verification using J-Link Commander is demonstrated, followed by procedures for flashing firmware and debugging applications on the AM243x/AM64x device using J-Link. This guide is intended for embedded software engineers who want to use SEGGER J-Link as a JTAG debug probe with the AM243x/AM64x MCU+ SDK development environment.

Table of Contents

1 Introduction	2
2 Software Setup	3
3 Hardware Setup	3
3.1 J-Link Connection	4
3.2 AM243x/AM64x Connection	4
4 Verifying Connection	5
4.1 Power Sequence	5
4.2 Verify Connection	5
5 Flashing with J-Link on AM243x/AM64x	6
6 Debugging with J-Link on AM243x/AM64x	8
7 Troubleshooting	11
8 Summary	11
9 References	11

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The Texas Instruments AM243x/AM64x is an industrial-grade, multi-core microcontroller featuring dual Arm® Cortex®-R5F cores, a Cortex®-M4F core, and dedicated PRU-ICSS subsystems. It is designed for demanding real-time applications including motor drives, robotics, grid infrastructure, and industrial communications protocols.

During firmware development, a capable and reliable JTAG debug probe is critical for efficiently loading and debugging applications across multiple cores. While the AM243x/AM64x EVM includes an onboard XDS110 debug probe for standard CCS-based workflows, the SEGGER J-Link probe offers an industry-proven alternative with advanced capabilities such as high-speed flash programming, real-time memory access without target halting, and compatibility with standalone debug environments such as SEGGER Ozone.

This application note provides a step-by-step guide to configuring and using the SEGGER J-Link debug probe with the AM243x/AM64x MCU+ SDK environment. It covers the required software and hardware setup, physical connection procedure, connection verification using J-Link Commander, and the workflow for flashing and debugging applications on the AM243x/AM64x device.

2 Software Setup

Install the following software and tools required to run MCU_PLUS_SDK applications on AM243x devices.

1. MCU_PLUS_SDK (Version 12.00 and above): [Download link for the AM243x MCU_PLUS_SDK](#)
2. Code Composer Studio™: [Download link for Code Composer Studio \(CCS\)](#)
3. Sysconfig: [Download link for Sysconfig tool](#)
4. TI-ARM-CLANG compiler: [Download link for the TI-ARM-CLANG Compiler](#)
5. Python3: [Download link for Python3](#)
6. OpenSSL: [Download link for OpenSSL](#)

If users need further help, then refer to the Download, Install and Setup SDK and Tools page from the official documentation.

SEGGER J-Link - [Download link for SEGGER J-Link Software](#)

Note

Debug and Flash programming support for TI AM243x and AM64xx devices has been added with following J-Link versions:

- AM243x: J-Link V9.44
 - AM64xx: J-Link V9.50
-

Ozone - [Download Link for Ozone](#)

3 Hardware Setup

The hardware needed to enable SEGGER J-Link is listed below:

1. AM243x/AM64x SoC.
2. SEGGER J-Link debug probe. ([Product Link](#))
3. 20-pin, 0.1" target ribbon cable(included with J-Link).
4. USB Type A to USB B cable for connecting J-Link with host PC. (included with J-Link)
5. J-Link TI CTI-20 connector.([Product Link](#))
6. Power supply to AM243x/AM64x EVM.

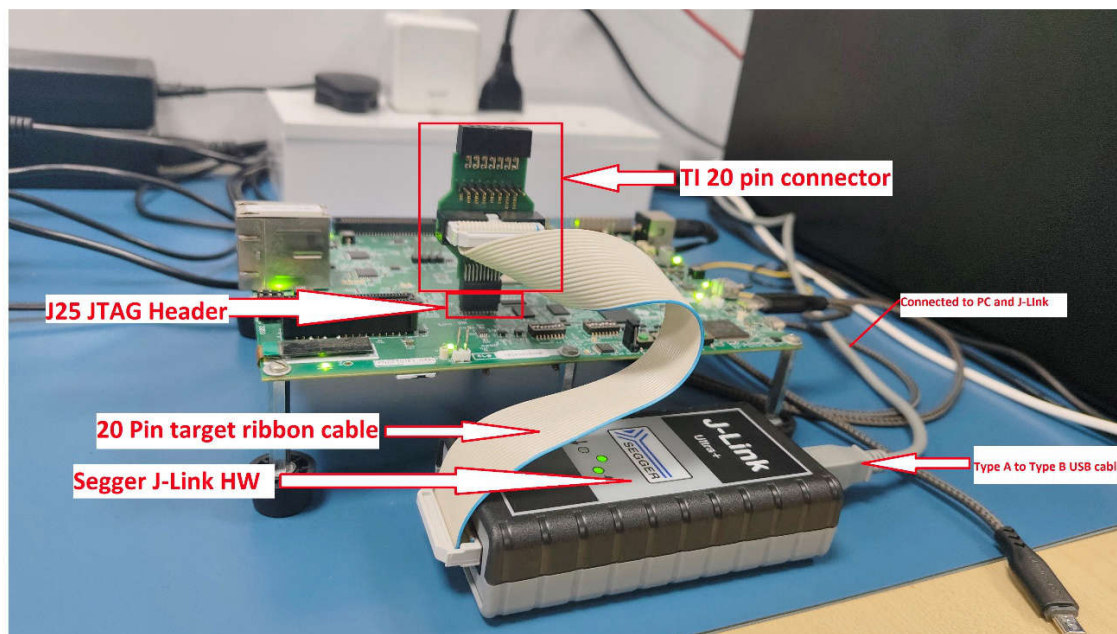


Figure 3-1. Hardware Connection

3.1 J-Link Connection

1. Connect the J-Link with PC using USB type A to USB type B connector.
2. Connect the 20-pin target ribbon cable with J-Link.
3. Attach the J-Link TI-CTI-20 pin connector with the 20-pin ribbon cable.

3.2 AM243x/AM64x Connection

1. Power on the AM243x/AM64x EVM.
2. Connect the J-Link to AM243x/AM64x EVM JTAG connector.(for example, J25 on TI AM243x/AM64x-EVM). Make sure the J-Link is powered up before connecting with the target EVM.

4 Verifying Connection

4.1 Power Sequence

In general, the J-Link / J-Trace must be powered on before connecting it with the target device. The user should first connect the J-Link / J-Trace with the host system through a USB and then connect the J-Link / J-Trace with the target device. Power-on the device after connecting the J-Link / J-Trace to it.

4.2 Verify Connection

If the USB driver is working properly and the J-Link / J-Trace is connected with the host system, the user can connect J-Link / J-Trace to the target hardware. Start J-Link Commander (JLink.exe) which must now display the normal J-Link / J-Trace related information, such as the hardware version and firmware version. After issuing the connect command and providing the additional information required to connect to the device, the J-Link connects to the device. Additional information about the target is shown (for example, ROM-Table).

Follow the steps:

1. Run the J-Link Commander(J-Link.exe) and run connect command.

```
Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: CORTEX-R5
Type '?' for selection dialog
```

Figure 4-1. Connect Command

2. Type ? for device and core selection.

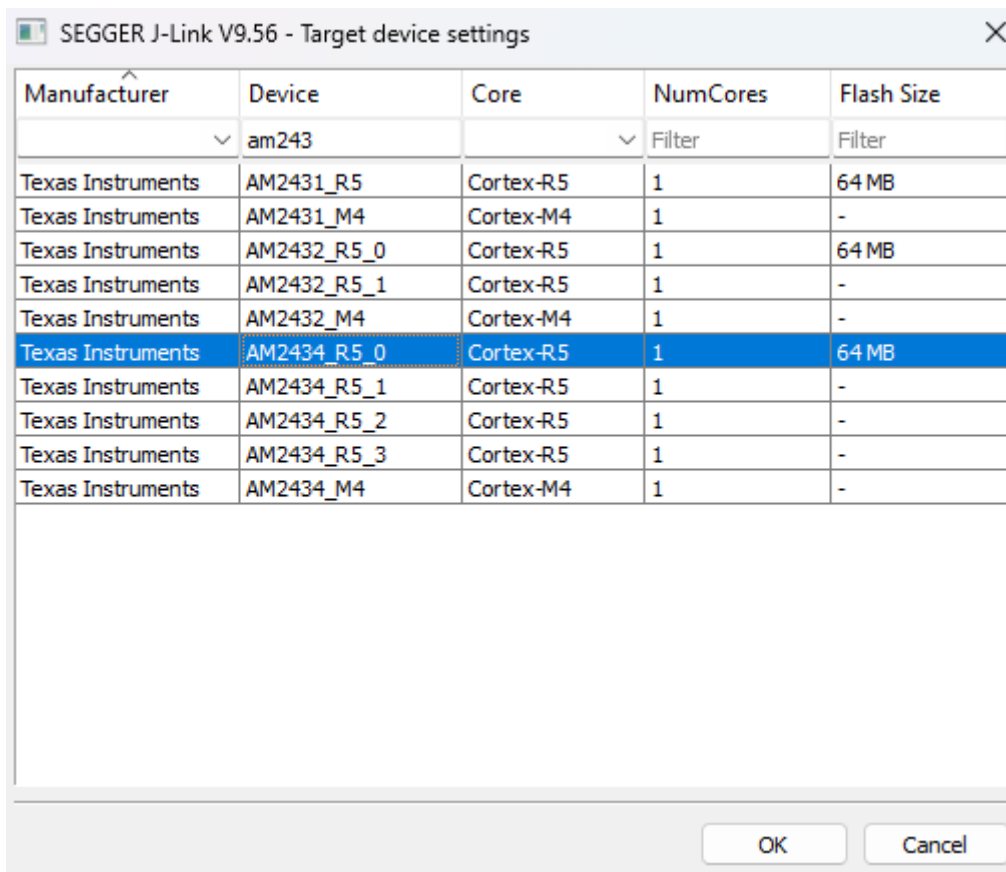


Figure 4-2. Device and Core Selection

3. Select the JTAG option from the list. The default value is JTAG. Press enter to select default.
4. Specify the device position in the JTAG chain. Pressing enter automatically detects the correct device position and should work in most cases.

5. Specify the target interface speed. The default value is 4000KHz. Press enter to select default.
6. Once the connection is successful, run different commands to load and debug the application using J-Link Commander. Some simple commands to try are:
 - *halt*: Halts the core.
 - *go*: Starts the core
 - *ishalted*: Returns whether the core is currently halted or not.
 - *regs*: Returns the current CPU registers

More details on SEGGER J-Link Commander can be found at https://kb.segger.com/J-Link_Commander

5 Flashing with J-Link on AM243x/AM64x

To Flash the AM243x/AM64x EVM, use J-Flash. J-Flash is a stand-alone flash programming software for PCs running Windows, Linux or macOS.

J-Flash has an intuitive user interface and makes programming flash devices convenient. J-Flash requires a J-Link / Flasher as an interface to the target hardware.

Below are the steps to flash the hello world example using J-Flash.

Steps to Flash using J-Flash:

1. Keep the AM243x/AM64x EVM in DEV BOOT mode.

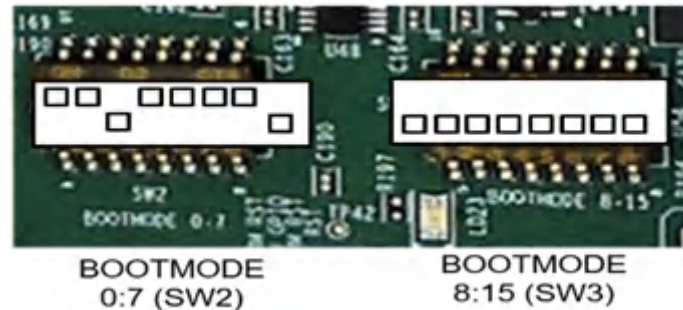


Figure 5-1. DEV Boot

2. Once the boot mode is set and EVM is turned on. Follow the steps mentioned at [Verify Connection](#).
3. After successful connection, run the following commands from J-Link Commander(J-Link.exe).

```
halt // Halt R5FSS0_CORE0
//
// !!! TODO: Update TI AM243x/AM64x SDK path !!!
//
loadfile "C:\TI\mcu_plus_sdk_{device}_{version}\tools\ccs_load\{device}
\sciclient_ccs_init.release.out"
setpc 0x0 // Set program counter to start of RAM
go // Start R5FSS0_CORE0
exit // Close J-Link Commander
```

4. After successful execution of above commands, run J-Flash.exe
5. Click on *Create New Project -> Start J-Flash -> Choose AM243x_R5_0*.
6. Select the *Target Interface as JTAG* and *enable Flash banks*.

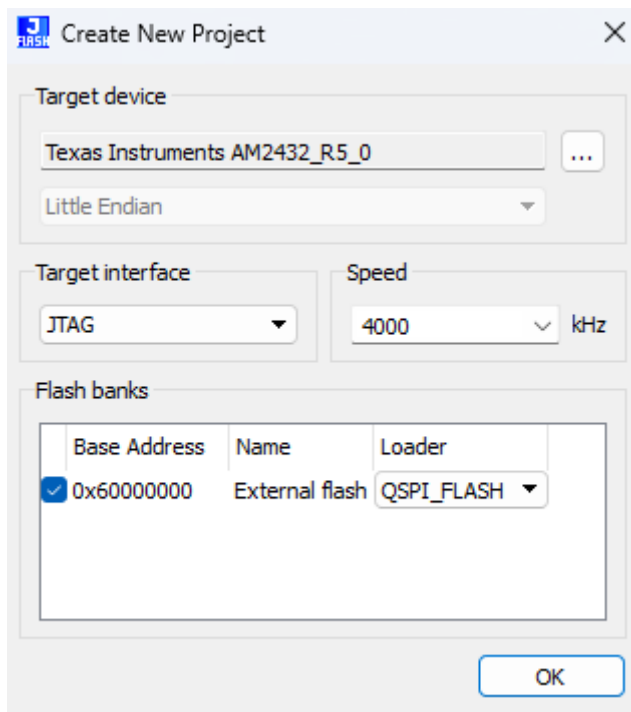


Figure 5-2. J-Flash Config

7. Click OK and navigate to *File -> Open Data File -> Select the SBL tiboot3.bin*. The file is available at `{MCU_SDK}\examples\drivers\boot\sbl\{evm}\{core}\{compiler}\tiboot3.bin`.
8. Enter the start address as `0x60000000`. (starting address of flash)
9. Navigate to *Target -> Production Programming*.
10. Copy the *appimage/mcelf* file of *hello_world* program which needs to be flashed. The hello world mcelf file will be generated at path `{MCU_SDK}\examples\hello_world\{evm}\{core}\{compiler}`.
11. Change the extension of copied file to *.bin*.
12. Navigate to *File -> Open Data File -> Select the hello_world.{debug|release}.mcelf.hs_fs.bin* file.
13. Enter the start address as `0x60200000`. (Flash offset from where SBL expects appimage to be present).

After successful execution of above steps, the binaries are now flashed. Change the boot mode to OSPI and the user can see the OSPI SBL and Hello world logs onto the UART console after the EVM is turned on.

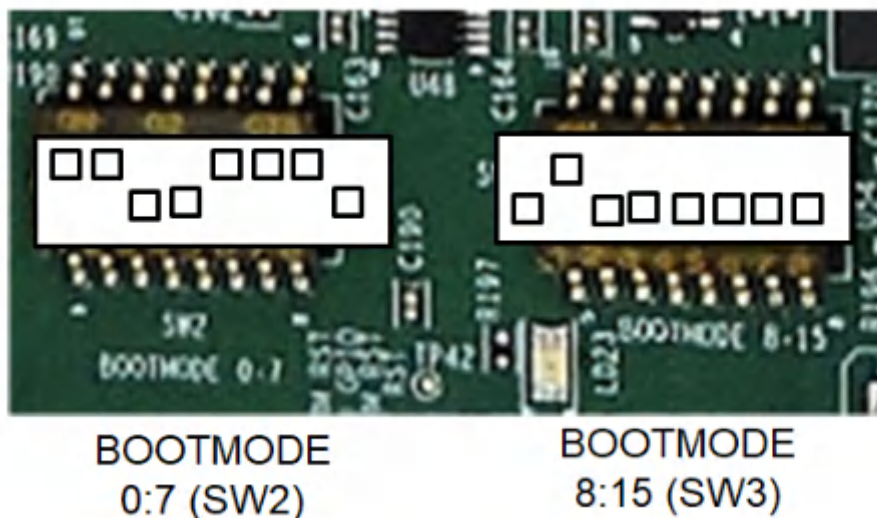


Figure 5-3. OSPI Boot

After successfully flashing an image which contains an SBL, successive flash programming operations can be executed in SPI Boot Mode without requiring the initialization steps one through three and the steps in the codeblock. See the following documentation from SEGGER J-Link for Flashing purpose.

1. For AM243x SoC Init - [Initializing_the_DMSC](#)
2. For AM64xx SoC Init- [Initializing_the_DMSC](#)
3. Getting started with J-Flash - [JFlash_Getting_Started](#)

6 Debugging with J-Link on AM243x/AM64x

Debugging involves tracing the source of problems that cause software to crash or behave unexpectedly and correcting them to verify the code functions exactly as intended. To debug applications using J-Link, the example uses Ozone software provided by SEGGER.

To download Ozone, visit <https://www.segger.com/downloads/jlink/#Ozone>

Steps to debug using Ozone:

1. Open the Ozone software and navigate to *File -> New -> New Project Wizard*
2. Select the device and core to debug.

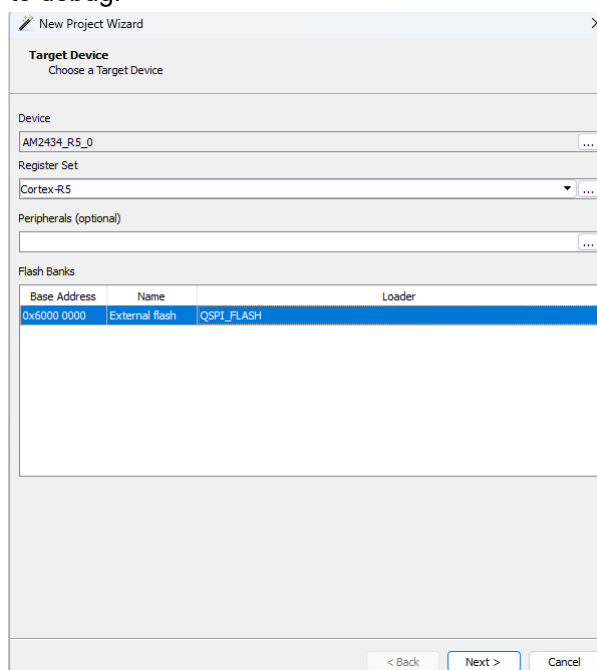


Figure 6-1. Device and Core Selection

3. Select the Target as JTAG and Host Interface as USB. Keep all settings as default.

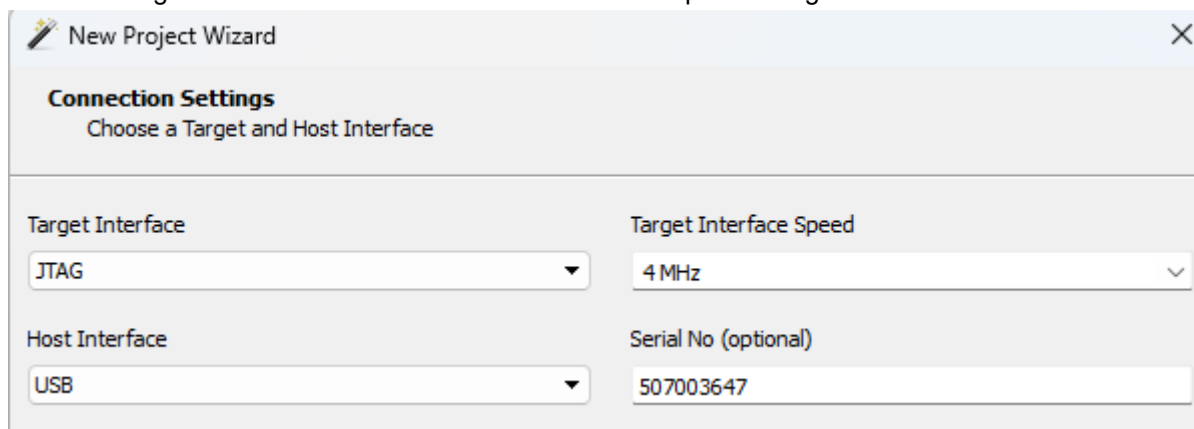


Figure 6-2. Connection Setting

4. Select the Program file(*.out) which needs to be debugged.

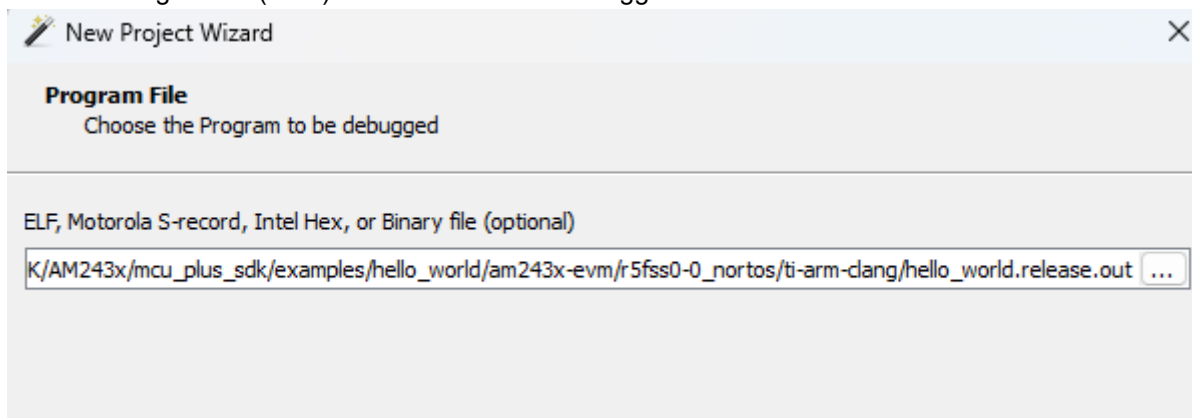


Figure 6-3. Program File

5. Keep the optional settings to be default.
6. Now the source code for the binary is opened in Ozone.
7. In the toolbar menu, different options are available to connect with the core. Click on the dropdown of power icon shown in [Figure 6-4](#).

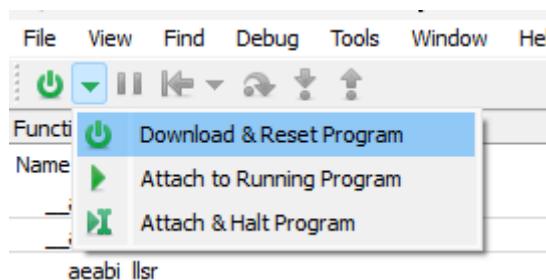


Figure 6-4. Power Icon Dropdown

- a. Download and Reset Program – Resets the CPU and load the Program to CPU
 - b. Attach to Running Program – Loads the debug symbols of the target application, which is already running on the core, and does not halt the CPU.
 - c. Attach and Halt Program – Loads the debug symbols of the target application, which is already running on the core and also halts the CPU.
8. Once connected to the core, different debugging options appears.

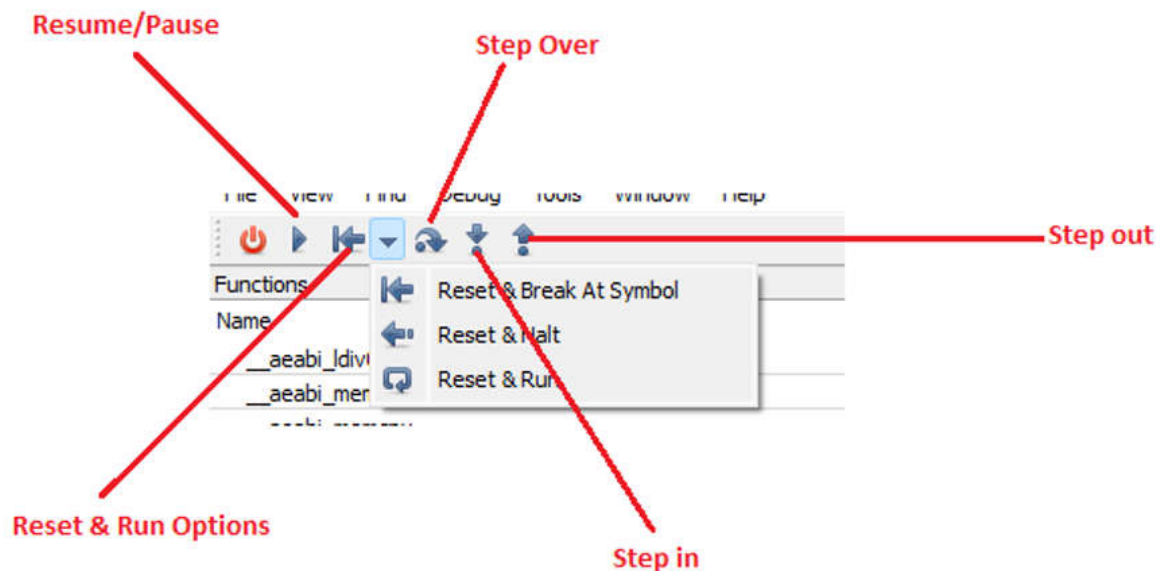


Figure 6-5. Debugging Menu

More detailed document on debugging using Ozone can be found at <https://www.segger.com/products/development-tools/ozone-j-link-debugger/technology/ozone-trace-tutorial/>

Note

The same debugging steps are applicable for other devices. See <https://www.segger.com/supported-devices/>

7 Troubleshooting

In case any of the above steps fail, following aspects must be considered:

- Is the evaluation board correctly powered?
- Is the J-Link correctly connected to the evaluation board?
- Are the boot mode switches correctly set for the desired operation?
- Do the J-Link logs show useful information on why debugging or flash programming failed? File logging can be enabled using following guide: https://kb.segger.com/Enable_J-Link_log_file

8 Summary

This application note presented a complete setup guide for using the SEGGER J-Link debug probe with the AM243x and AM64x device. Software setup required the following tools to be installed: AM243x and AM64x MCU+ SDK (version 12.00 or above), Code Composer Studio, SysConfig, TI-ARM-CLANG compiler, Python3, OpenSSL, the SEGGER J-Link software package, and the Ozone debugger. Hardware setup required an AM243x/AM64x SoC/EVM, a SEGGER J-Link debug probe, a 20-pin 0.1" target ribbon cable with a J-Link TI CTI-20 connector, a USB Type-A to Type-B cable, and a J-Link connector adapter. Connection and power sequencing followed this order: connect J-Link to the host PC through USB first, then connect J-Link to the AM243x and AM64x EVM JTAG connector (J25 on the TI AM243x and AM64x-EVM), and finally power on the EVM. This sequence verifies the J-Link enumerates correctly before the target is powered. Connection verification was performed using J-Link Commander (JLink.exe) by issuing the connect command, selecting the appropriate device and JTAG interface, and confirming that the ROM table is read back successfully. With the connection established, developers can use J-Link to flash firmware onto the AM243x and AM64x device and perform full debug sessions, providing an alternative debug path to the default XDS110 JTAG probe included with the EVM.

9 References

1. SEGGER J-Link, [UM08001_J-Link_/J-Trace_User_Guide](#), user's guide.
2. Texas Instruments, [AM243x/AM64x EVM User Guide](#), EVM user's guide.
3. SEGGER Knowledge Base, [TI AM243x Device Support](#), webpage.
4. SEGGER Knowledge Base, [TI AM64xx Device Support](#), webpage.
5. SEGGER Knowledge Base, [TI AM62xx Device Support](#), webpage.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#), [TI's General Quality Guidelines](#), or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025