

Application Note

SEGGER J-Link を使用した AM243x/AM64x デバイスでのアプリケーションのデバッグ



Tushar Thakur, Anil Swargam, Michael Gross, Matthias Konegen

概要

このアプリケーション ノートでは、テキサス・インスツルメンツの AM243x/AM64x デバイスで SEGGER J-Link デバッグプロローブを設定して使用し、組み込みアプリケーションのロードとデバッグを行う方法について説明します。このドキュメントでは、AM243x/AM64x MCU+ SDK (v12.00 以降)、Code Composer Studio、SysConfig、TI-ARM-CLANG コンパイラ、SEGGER J-Link ソフトウェア パッケージ、Ozone デバッガなど、必要なソフトウェアのインストールから始まる包括的なワークフローについて説明しています。次に、列挙の問題を回避するための正しい電源シーケンスなど、J-Link プロローブと AM243x/AM64x 評価基板の JTAG コネクタの間の物理的なハードウェア要件と、ステップバイステップの接続手順について詳しく説明します。J-Link コマンドを使用した接続検証を示し、続いて、J-Link を使用した AM243x/AM64x デバイスでファームウェアのフラッシュ書き込みおよびデバッグアプリケーションの手順を示します。このガイドは、AM243x/AM64x MCU+ SDK 開発環境で、SEGGER J-Link を JTAG デバッグ プロローブとして使用する、組み込みソフトウェアのエンジニアを対象としています。

目次

1 概要.....	2
2 ソフトウェアの設定.....	3
3 ハードウェア設定.....	3
3.1 J-Link 接続.....	4
3.2 AM243x/AM64x 接続.....	4
4 接続を確認する.....	5
4.1 電源シーケンス.....	5
4.2 接続を確認する.....	5
5 J-Link を使用した AM243x/AM64x でのフラッシュ書き込み.....	6
6 AM243x/AM64x における J-Link を使用したデバッグ.....	8
7 トラブルシューティング.....	11
8 まとめ.....	11
9 参考資料.....	11

商標

すべての商標は、それぞれの所有者に帰属します。

1 概要

テキサス・インスツルメンツの AM243x/AM64x は、デュアル Arm® Cortex®-R5F コア、Cortex®-M4F コア、専用の PRU-ICSS サブシステムを搭載した産業グレードのマルチコア マイコンです。モータードライブ、ロボティクス、グリッド インフラ、産業用通信プロトコルなど、厳しい条件が要求されるリアルタイム アプリケーション向けの設計を採用しています。

ファームウェア開発時に、複数のコアにまたがるアプリケーションのロードとデバッグを効率的に行うには、対応能力と信頼性の高い JTAG デバッグ プローブが不可欠です。AM243x/AM64x 評価基板は、標準の CCS ベース ワークフロー用の オンボード XDS110 デバッグ プローブを搭載していますが、SEGGER J-Link プローブは、高速フラッシュ プログラミング、ターゲット停止なしのリアルタイム メモリ アクセス、SEGGER Ozone などのスタンドアロン デバッグ環境との互換性など、高度な機能を備えた、業界で実証済みの代替選択肢を提供します。

このアプリケーション ノートでは、AM243x/AM64x MCU+ SDK 環境で SEGGER J-Link デバッグ プローブを構成および使用するためのステップバイステップのガイドを掲載しています。必要なソフトウェアとハードウェアのセットアップ、物理的な接続手順、J-Link コマンドを使用した接続検証、AM243x/AM64x デバイス上のアプリケーションのフラッシュ書き込みおよびデバッグのワークフローについて説明します。

2 ソフトウェアの設定

AM243x デバイスで MCU_PLUS_SDK アプリケーションを実行するために必要な、以下のソフトウェアとツールをインストールします。

1. MCU_PLUS_SDK (バージョン 12.00 以降): [AM243x MCU_PLUS_SDK のダウンロードリンク](#)
2. Code Composer Studio™: [Code Composer Studio \(CCS\) のダウンロードリンク](#)
3. Sysconfig: [Sysconfig ツールのダウンロードリンク](#)
4. TI-ARM-CLANG コンパイラ: [TI-ARM-CLANG コンパイラのダウンロードリンク](#)
5. Python3: [Python3 のダウンロードリンク](#)
6. OpenSSL: [OpenSSL のダウンロードリンク](#)

ユーザーがさらにサポートを必要とする場合は、次の公式資料の SDK およびツールのダウンロード、インストール、セットアップページを参照してください

。

SEGGER J-Link - [SEGGER J-Link ソフトウェアのダウンロードリンク](#)

注

TI AM243x および AM64xx デバイスのデバッグおよびフラッシュ プログラミングのサポートが、以下の J-Link バージョンで追加されています。

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

Ozone - [Ozone のダウンロードリンク](#)

3 ハードウェア設定

SEGGER J-Link を有効にするために必要なハードウェアを以下に示します。

1. AM243x/AM64x SoC。
2. SEGGER J-Link デバッグ プロンプト。(製品リンク)
3. 20 ピン、0.1 インチ ターゲットリボン ケーブル (J-Link に付属)。
4. ホスト PC に J-Link を接続するための USB Type A - USB Type B ケーブル。(J-Link に付属)
5. J-Link TI CTI-20 コネクタ。(製品リンク)
6. AM243x/AM64x 評価基板への電源。

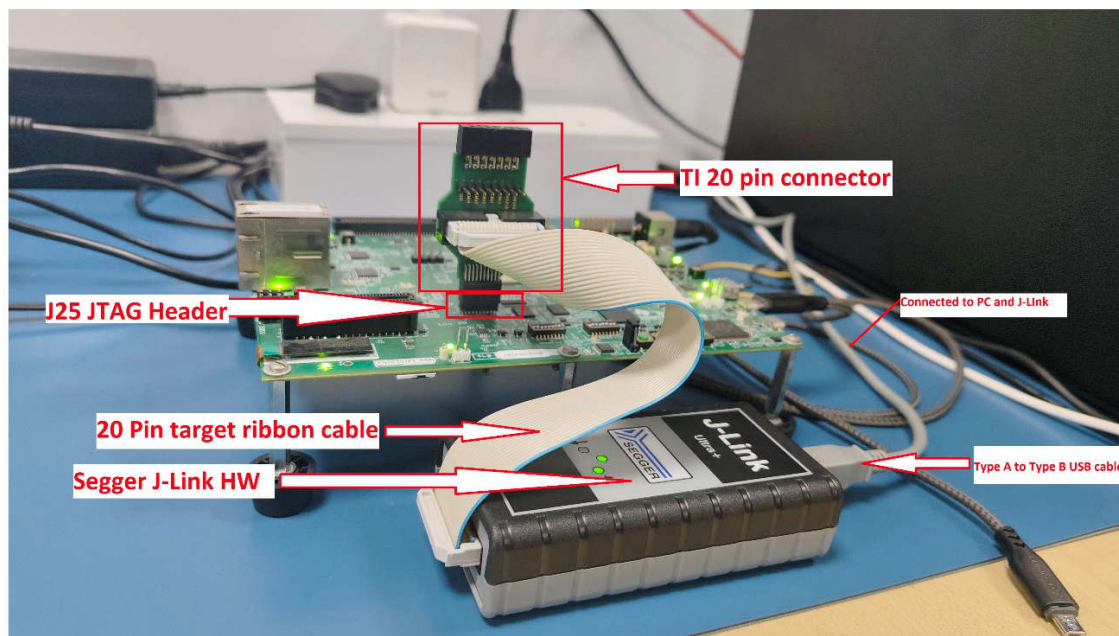


図 3-1. ハードウェア接続

3.1 J-Link 接続

1. USB type A - USB type B のコネクタを使って、J-Link を PC に接続します。
2. 20 ピン ターゲット リボン ケーブル を J-Link に接続します。
3. J-Link TI-CTI-20 ピン コネクタ を 20 ピン リボン ケーブル に接続します。

3.2 AM243x/AM64x 接続

1. AM243x/AM64x 評価基板をパワーオンします。
2. J-Link を AM243x/AM64x 評価基板 JTAG コネクタ (例えば TI AM243x/AM64x-EVM の J25) に接続します。ターゲットの評価基板に接続する前に、J-Link の電源がオンになっていることを確認してください。

4 接続を確認する

4.1 電源シーケンス

一般に、J-Link / J-Trace は、ターゲット デバイスに接続する前に電源をオンにする必要があります。まず USB を介して J-Link / J-Trace をホスト システムに接続し、次に J-Link / J-Trace をターゲット デバイスに接続します。J-Link / J-Trace を接続した後で、デバイスの電源をオンにします。

4.2 接続を確認する

USB ドライバが正常に動作しており、J-Link / J-Trace がホスト システムに接続されている場合は、J-Link / J-Trace をターゲット ハードウェアに接続できます。J-Link Commander (JLink.exe) を起動します。これで、ハードウェアのバージョンやファームウェアのバージョンなど、通常の J-Link / J-Trace 関連情報が表示されます。connect コマンドを発行し、デバイスへの接続に必要な追加情報を提供した後、J-Link はデバイスに接続します。ターゲットに関する追加情報が表示されます (例:ROM テーブル)。

以下の手順に従います:

1. J-Link コマンド (J-Link.exe) を実行し、connect コマンドを実行します。

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

図 4-1. connect コマンド

2. デバイスとコアの選択で「?」と入力します。

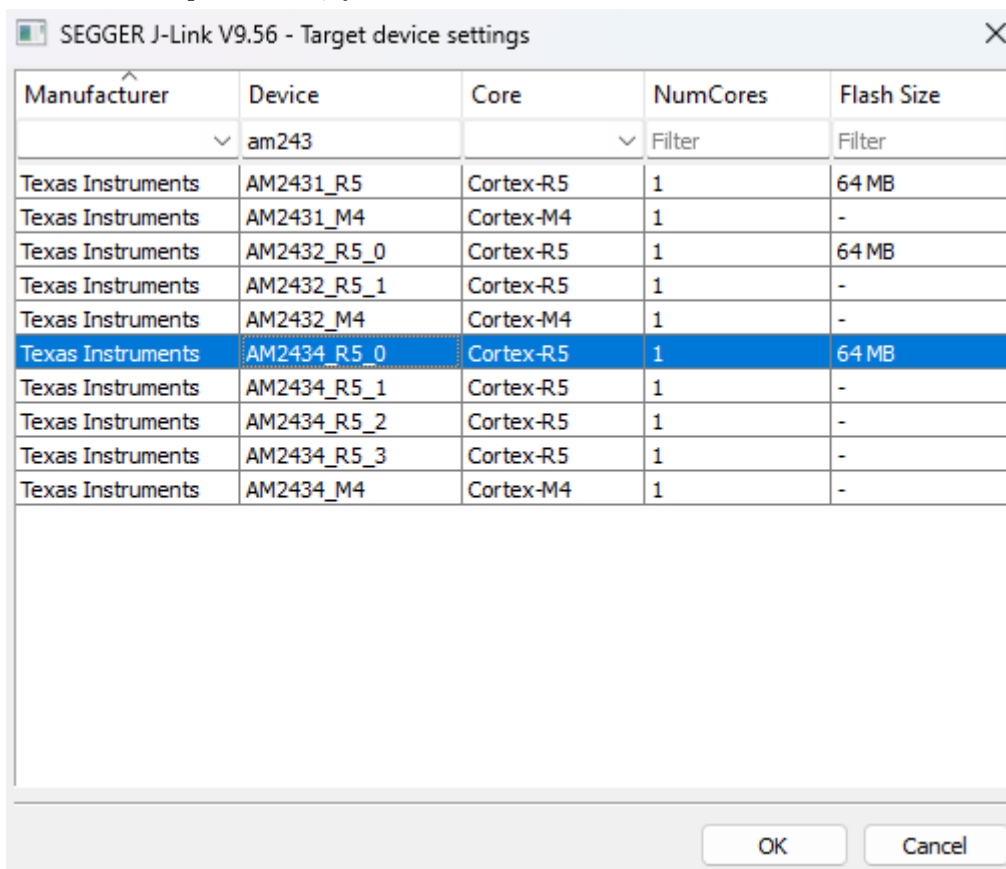


図 4-2. デバイスとコアの選択

3. リストから JTAG オプションを選択します。デフォルト値は JTAG です。Enter キーを押してデフォルトを選択します。

4. JTAG チェーン内のデバイス位置を指定します。ENTER キーを押すと、デバイスの正しい位置が自動的に検出され、ほとんどの場合で動作します。
5. ターゲット インターフェイス速度を指定します。デフォルト値は 4000KHz です。Enter キーを押してデフォルトを選択します。
6. 接続に成功した後、さまざまなコマンドを実行し、J-Link コマンドを使用してアプリケーションをロードして、デバッグします。次のようなシンプルなコマンドを試してみてください。
 - **halt**: コアを停止します。
 - **go**: コアを起動します
 - **ishalted**: コアが現在停止しているかどうかを返します。
 - **regs**: 現在の CPU レジスタを返します

SEGGER J-Link Commander の詳細については、https://kb.segger.com/J-Link_Commander を参照してください

5 J-Link を使用した AM243x/AM64x でのフラッシュ書き込み

AM243x/AM64x 評価基板へのフラッシュ書き込みには、J-Flash を使用します。J-Flash は、Windows、Linux、macOS を実行する PC 向けのスタンドアロンのフラッシュ プログラミング ソフトウェアです。

J-Flash は直感的なユーザー インターフェイスを備えており、フラッシュ デバイスのプログラミングに便利です。J-Flash には、ターゲット ハードウェアへのインターフェイスとして J-Link / Flasher が必要です。

例として、J-Flash を使用して hello world をフラッシュする手順を以下に示します。

J-Flash を使用したフラッシュの手順:

1. AM243x/AM64x 評価基板を、DEV ブート モードに維持します。

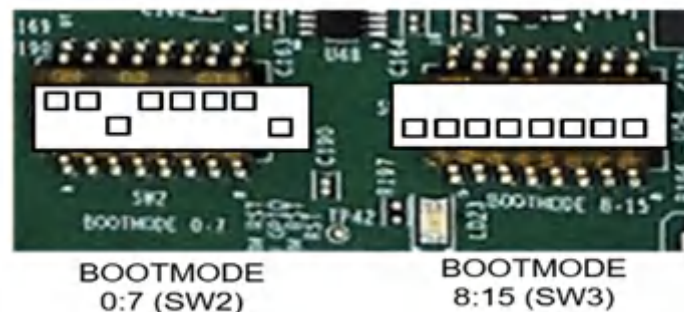


図 5-1. DEV ブート

2. ブート モードが設定され、評価基板がオンになります。「[接続を確認する](#)」に記載されている手順を実行します。
3. 接続が成功したら、J-Link コマンド (J-Link.exe) から次のコマンドを実行します。

```
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. 上述のコマンドが正常に実行されたら、J-Flash.exe を実行します
5. 「Create New Project」(新規プロジェクトの作成) -> 「Start J-Flash」(J-Flash を開始) -> 「Choose AM243x_R5_0」(AM243x_R5_0 を選択) をクリックします。
6. 「Target Interface as JTAG」(ターゲット インターフェイスに JTAG)、「Enable Flash banks」(フラッシュ バンクをイネーブル)を選択します。

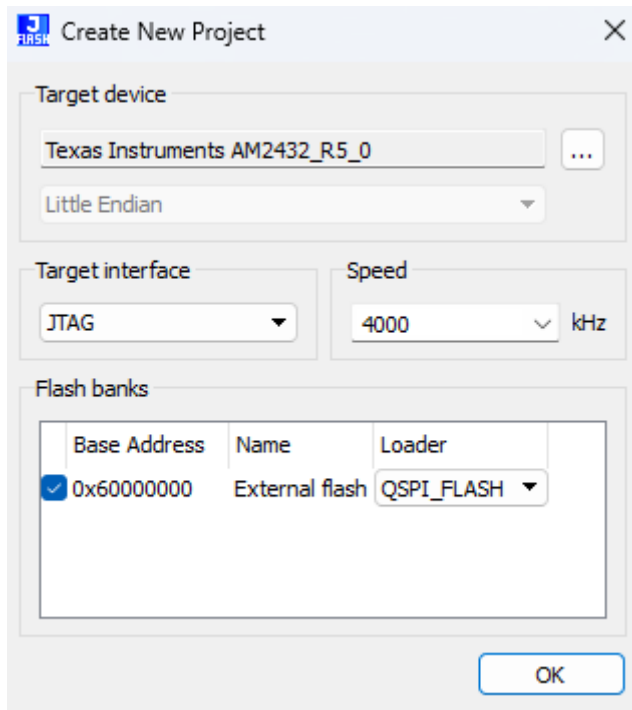


図 5-2. J-Flash の構成

7. [OK] をクリックし、「File」(ファイル) -> 「Open Data File」(データ ファイルを開く) -> 「Select the SBL tiboot3.bin」(SBL tiboot3.bin を選択) と進みます。このファイルは、`{MCU_SDK}\examples\drivers\boot\{sbl}\{evm}\{core}\{compiler}\tiboot3.bin` で入手できます。
8. 開始アドレスに `0x60000000` を入力します。(フラッシュの開始アドレス)
9. 「Target」(ターゲット) -> 「Production Programming」(プロダクション プログラミング) -> に移動します。
10. フラッシュする必要がある hello_world プログラムの `appimage/mcelf` ファイルをコピーします。hello world mcelf ファイルはパス `{MCU_SDK}\examples\hello_world\{evm}\{core}\{compiler}` に生成されます。
11. コピーしたファイルの拡張子を `*.bin` に変更します。
12. 「File」(ファイル) -> 「Open Data File」(データ ファイルを開く) -> `Select the hello_world.{debug|release}.mcelf.hs_fs.bin` ファイルに移動します。
13. 開始アドレスに `0x60200000` を入力します。(SBL が appimage が存在することを想定している Flash オフセット)。

上の手順が正常に実行されると、バイナリがフラッシュされます。ブート モードを OSPI に変更すると、評価基板の電源がオンになった後で、OSPI SBL および Hello World のログを UART コンソールに表示できます。

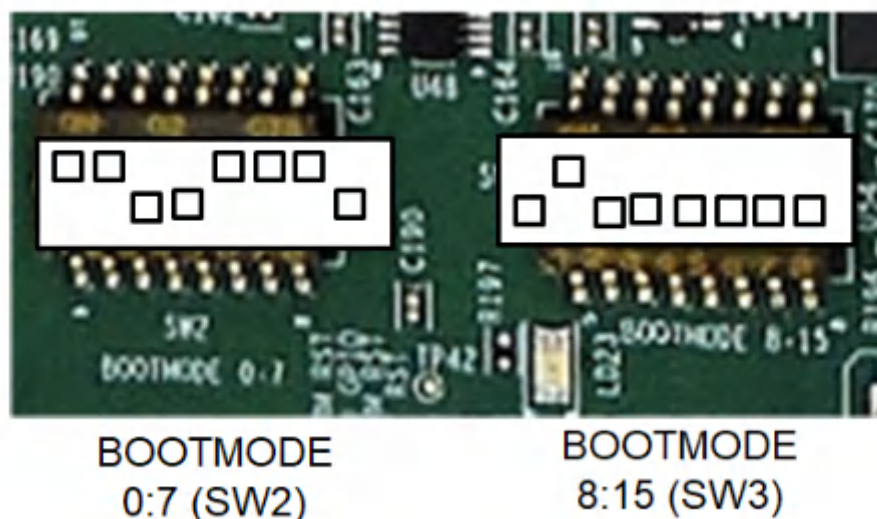


図 5-3. OSPI ブート

SBL を含むイメージを正常にフラッシュ書き込みした後、初期化手順 1 ～ 3 やコードブロックのステップを必要とせずに、連続するフラッシュ プログラミング動作を SPI ブート モードで実行できます。フラッシュの目的については、SEGGER J-Link の次のドキュメントを参照してください。

1. AM243x SoC の初期化については、[Initializing_the_DMSC](#) を参照してください。
2. AM64xx SoC の初期化については、[Initializing_the_DMSC](#) を参照してください。
3. J-Flash の導入方法については、[JFlash_Getting_Started](#) を参照してください。

6 AM243x/AM64x における J-Link を使用したデバッグ

デバッグには、ソフトウェアがクラッシュしたり、予期せず動作したりする問題の根本原因を追跡し、コードが確実に、意図したとおりに機能するように、それらを修正することが含まれます。J-Link を使用してアプリケーションをデバッグするのに、この例では SEGGER が提供する Ozone ソフトウェアを使用しています。

Ozone をダウンロードするには、<https://www.segger.com/downloads/jlink/#Ozone> にアクセスしてください

Ozone を使用してデバッグする手順：

1. Ozone ソフトウェアを開き、「File」(ファイル) -> 「New」(新規作成) -> 「New Project Wizard」(新規プロジェクトウィザード) の順に移動します
2. デバッグするデバイスとコアを選択します。

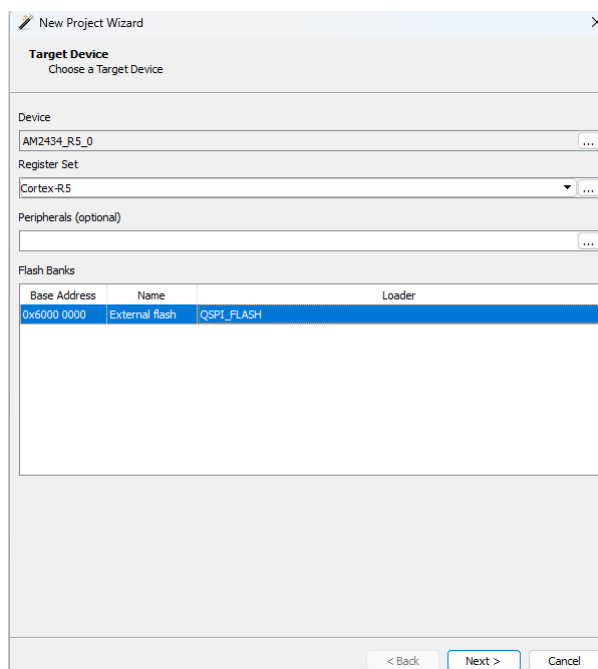


図 6-1. デバイスとコアの選択

3. JTAG としてターゲットを、USB としてホスト インターフェイスを選択します。すべての設定をデフォルトのままにします。

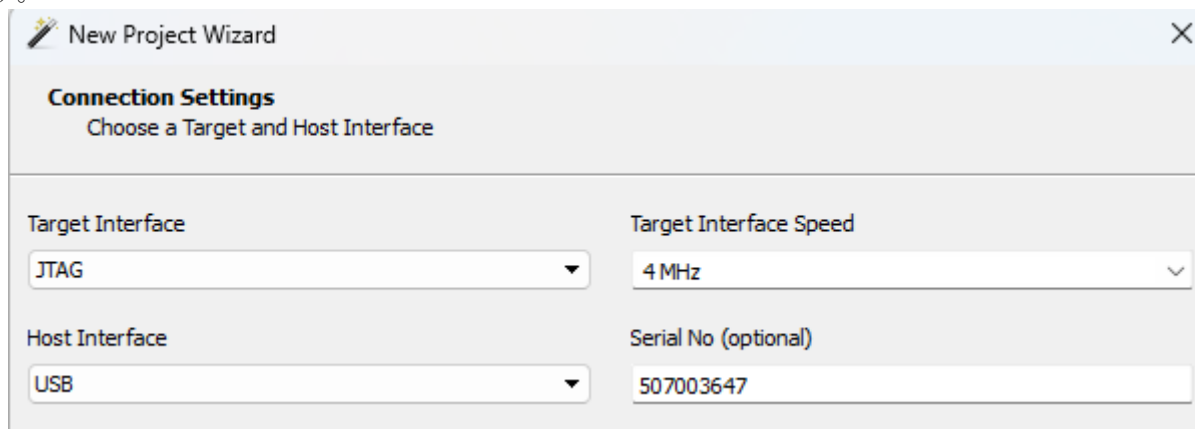


図 6-2. 接続設定

4. デバッグする必要があるプログラム ファイル (*.out) を選択します。

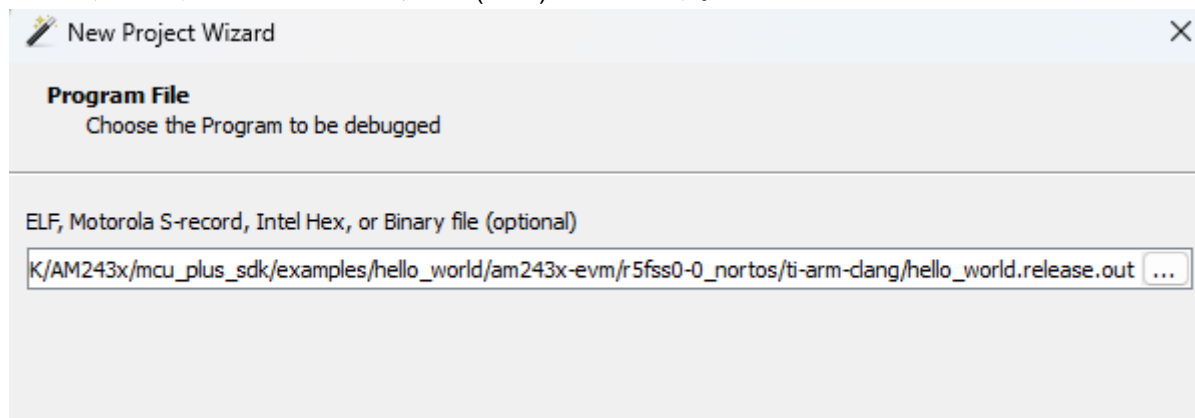


図 6-3. プログラム ファイル

5. オプションの設定はデフォルトのままにします。

6. これで、バイナリのソース コードが Ozone で開かれます。
7. ツールバーのメニューでは、コアと接続するためのさまざまなオプションを使用できます。図 6-4 に示すように、電源アイコンのドロップダウンをクリックします。

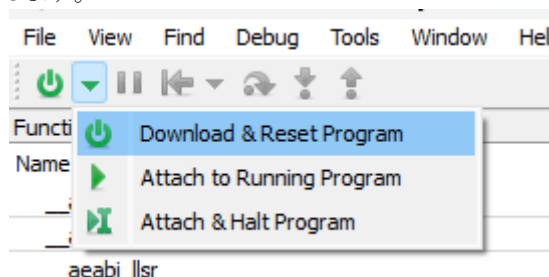


図 6-4. 電源アイコンのドロップダウン

- a. Download and Reset Program (プログラムのダウンロードとリセット) - CPU をリセットし、プログラムを CPU にロードします
 - b. Attach to Running Program (プログラムにアタッチして実行) - コア上ですでに動作しているターゲット アプリケーションのデバッグ シンボルを読み込み、CPU を停止させません。
 - c. Attach and Halt Program (プログラムにアタッチして停止) - コア上ですでに動作しているターゲット アプリケーションのデバッグ シンボルを読み込み、CPU を停止させます。
8. コアに接続すると、さまざまなデバッグ オプションが表示されます。

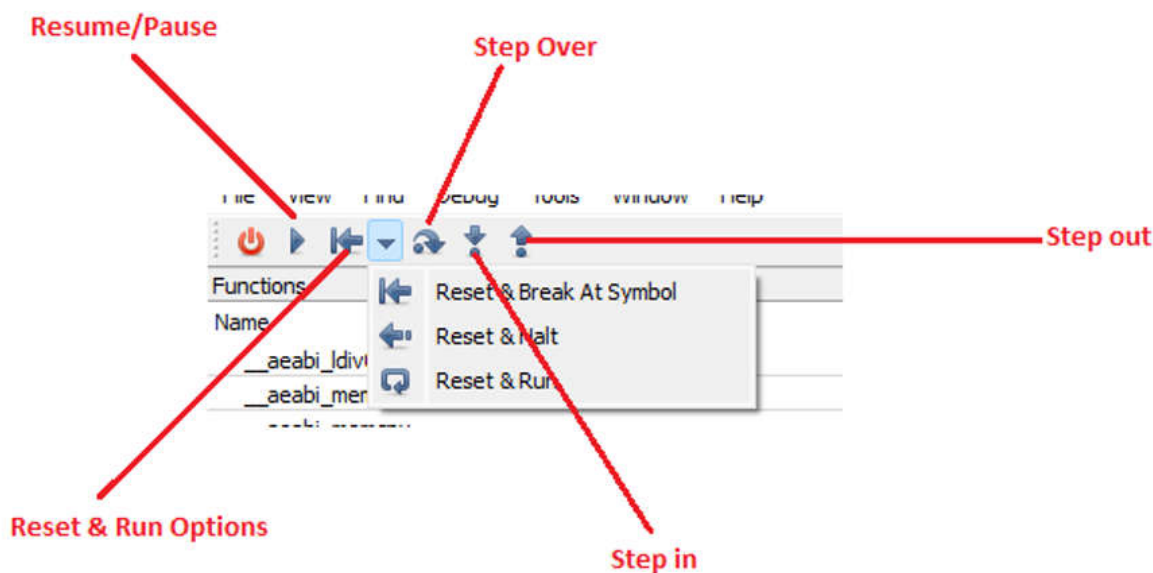


図 6-5. デバッグ メニュー

Ozone を使用したデバッグに関する詳細なドキュメントは、<https://www.segger.com/products/development-tools/ozone-j-link-debugger/technology/ozone-trace-tutorial/> にあります

注

他のデバイスにも、同じデバッグ手順を適用できます。<https://www.segger.com/supported-devices/> を参照してください

7 トラブルシューティング

上の手順のいずれかに失敗した場合は、次の点を考慮する必要があります。

- 評価ボードに正しく電源が供給されているか
- J-Link は評価ボードに正しく接続されているか
- ブート モード スイッチは目的の操作に適した設定になっているか
- デバッグまたはフラッシュ プログラミングが失敗した理由に関する、有用な情報が J-Link ログに表示されているか
ファイアログは、https://kb.segger.com/Enable_J-Link_log_file のガイドを使用して有効にできます

8 まとめ

このアプリケーション ノートは、AM243x および AM64x デバイスで SEGGER J-Link デバッグ プローブを使用するための包括的なセットアップ ガイドを提供しました。ソフトウェアのセットアップでは、AM243x および AM64x MCU+ SDK (バージョン 12.00 以上)、Code Composer Studio、SysConfig、TI-ARM-CLANG コンパイラ、Python3、OpenSSL、SEGGER J-Link ソフトウェア パッケージ、および Ozone デバッグをインストールする必要があります。ハードウェアのセットアップでは、AM243x/AM64x SoC / 評価基板、SEGGER J-Link デバッグ プローブ、J-Link TI CTI USB-20 コネクタ付き 20 ピン 0.1 インチ ターゲットリボン ケーブル、USB Type-A から Type-B へのケーブル、J-Link コネクタアダプタが必要になります。次の順序で、接続と電源シーケンスを実行します。まず USB 経由で J-Link をホスト PC に接続し、次に J-Link を AM243x および AM64x 評価基板 JTAG コネクタ (TI AM243x および AM64x-EVM の J25) に接続し、最後に評価基板の電源をオンにします。このシーケンスでは、ターゲットに電力が供給される前に、J-Link が正しく列挙されることを検証します。接続検証は、connect コマンドを発行し、適切なデバイスと JTAG インターフェイスを選択し、ROM テーブルが正常に読み取られることを確認することで、J-Link Commander (JLink.exe) を使用して実行されました。接続が確立されると、開発者は J-Link を使用して AM243x と AM64x の各デバイスにファームウェアをフラッシュ書き込み、包括的なデバッグセッションを実行できるので、評価基板に付属しているデフォルトの XDS110 JTAG プローブへの代替のデバッグ パスを確保できます。

9 参考資料

1. SEGGER J-Link、『[UM08001_J-Link/_J-Trace_User_Guide](#)』、ユーザー ガイド。
2. テキサス・インスツルメンツ、『[AM243x/AM64x 評価基板ユーザー ガイド](#)』、評価基板ユーザー ガイド。
3. SEGGER ナレッジ ベース、[TI AM243x Device Support](#)、Web ページ。
4. SEGGER ナレッジ ベース、[TI AM64xx Device Support](#)、Web ページ。
5. SEGGER ナレッジ ベース、[TI AM62xx Device Support](#)、Web ページ。

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含みいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとし、TI は一切の責任を拒否します。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、[TI の販売条件](#)、[TI の総合的な品質ガイドライン](#)、[ti.com](#) または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

Copyright © 2026, Texas Instruments Incorporated

最終更新日：2025 年 10 月