参考資料

Technical White Paper

CCStudio™ IDE のエージェント型 AI で組込み開発を加速

概要

ソフトウェアエンジニアリングがエージェント AI へと移行する中で、組込みシステムには、低水準のハードウェアの制約、複雑なペリフェラル構成、厳格なタイミング要件など、独自の障壁が存在します。このような環境では、ハルシネーションによって生成されたコードが、フィールドでの障害を引き起こす可能性があります。この記事では、テキサスインスツルメンツの CCStudio™ IDE がハードウェアに基づくインテリジェンスを開発プロセスにもたらし、AI 機能をハードウェア構成、ビルド管理、オンチップ デバッグのワークフローにまで拡張する方法を紹介します。CCStudio IDE は多層アーキテクチャを採用しており、AI エージェントが初期設定から最終的なトラブルシューティングまで、実際のシリコン上で作業を進めることができます。

目次

1 概要.....2
2 コア プラットフォームの機能.....3
3 CCStudio IDE 移行スキル.....4
4 移行スキルによる結果の検証方法.....5
5 まとめ.....5

商標

の CCStudio™ is a trademark of Texas Instruments.
すべての商標は、それぞれの所有者に帰属します。

Copyright © 2026 Texas Instruments Incorporated

1 概要

AI ツールは、かつて開発者が直接手をかける必要があったタスクを一層多く担うようになり、最初のアイデアから動作するコードができるまでの時間を短縮しています。Gartner は、2028 年までにエンタープライズ ソフトウェア エンジニアの 90% が AI コード アシスタントを使うようになると予測しています。図 1-1 に、開発ライフサイクル全体で AI ツールをどのように活用できるかを示します。

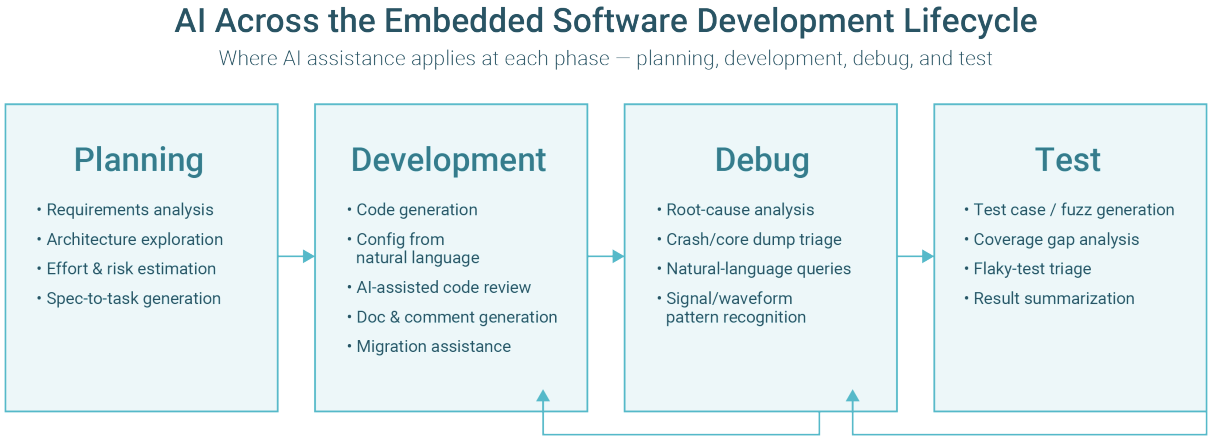


図 1-1. 組込みソフトウェア開発における AI ツールの活用を示すワークフロー図

しかし、標準的な AI モデルは、組込みシステムの複雑さに対応しきれていません。たとえば、ピン割り当て、ペリフェラルの設定、低レベルのタイミングに誤りがある場合、ハードウェア固有のデータにアクセスできないモデルでは、それによって生じたエラーを特定するのが難しくなります。

このような状況で AI 対応ワークフローを成功させるには、組み込み開発者が頼りにしている特定のハードウェアやツールチェーンの信頼できるデータに、エージェントがアクセスできる必要があります。そうしないと、AI は、正しくコンパイルできても、ハードウェアについての誤った前提が原因で実行時に失敗するコードを生成してしまう可能性があります。

CCStudio 開発エコシステムの一部である CCStudio IDE の専用環境では、AI エージェントが TI の編集、ビルド、デバッグ ツールに直接アクセスできます。CCStudio IDE は、組み込みシステム開発向けの完全で自律的なワークフローをサポートしています (図 1-2)。

Copyright © 2026 Texas Instruments Incorporated

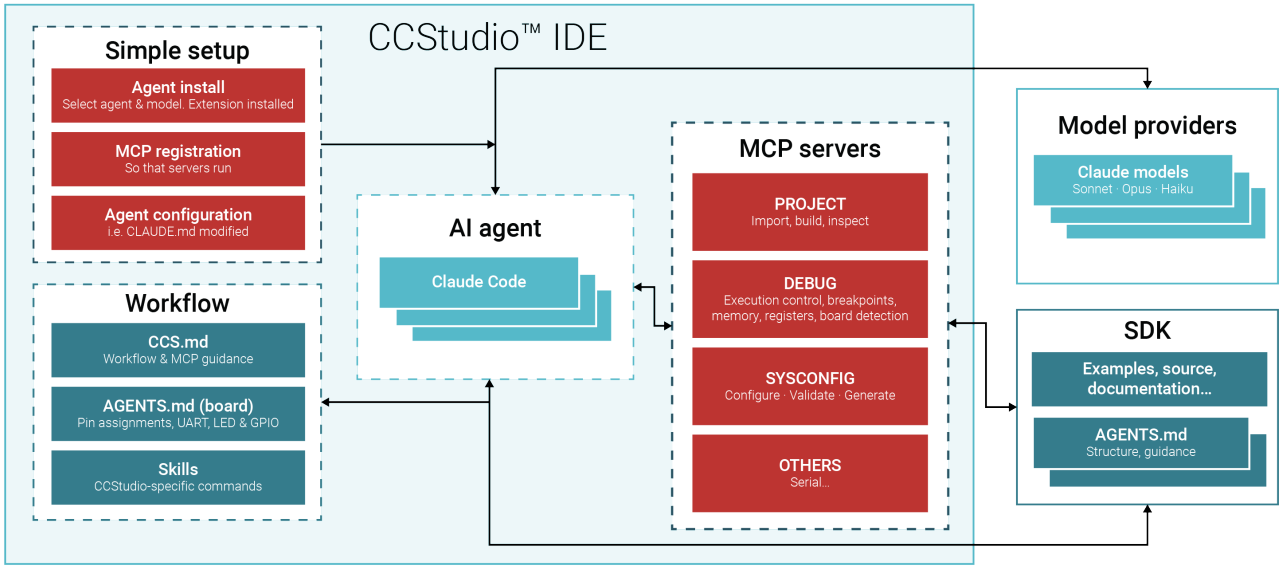


図 1-2. CCStudio IDE へのエージェント型 AI 機能の統合を示す簡略図

2 コア プラットフォームの機能

CCStudio IDE は、以下のようないくつかの統合機能を通じて AI エージェントの効率的な運用に必要な技術インフラを実現します:

1. **マルチモデルの相互運用性:**CCStudio IDE は業界標準のエージェントとモデルをサポートしているため、チームは Claude、GPT、オンプレミス モデルなど、好みのモデルを選択できます。このアプローチは、エンタープライズ セキュリティにも対応しています。オンプレミスやお客様管理のモデル ホスティングを最優先のシナリオとして扱っているため、独自のファームウェアや IP を保護されたネットワーク内にとどめておけます。
2. **自動 AI 構成ウィザード:**AI 構成ウィザードは、手動での設定作業を減らすために、MCP サーバーの登録、認証情報の設定、拡張機能のインストールを自動化します。これにより、開発者が手動でエージェントを構成しなくても、エージェントがプロジェクトを認識し、すぐに作業できる状態になります。

Copyright © 2026 Texas Instruments Incorporated

3. **構造化されたドキュメントと例:** 信頼できるデータがないと、エージェントは誤った構成を生成 (ハルシネーション) することがあります。たとえば、エージェントが正しいクロック ソースを把握していないために、誤ったボー レート除数を計算してしまう場合です。CCStudio IDE では、エージェントが、構造化された API リファレンス、レジスタ マップ、デバイス固有のサンプルなど、検証済みの TI ドキュメントに直接アクセスできます。これらのドキュメントは、構成作業の基準として役立ちます。
4. **SysConfig を使用した決定論的な構成:** SysConfig は、AI エージェントとハードウェアの間で、決定論的なインターフェイスとして機能します。SysConfig は、エージェントにペリフェラルの設定を推測させるのではなく、リアルタイムの検証レイヤを提供します。エージェントは変更を提案し、ピンやクロック ツリーの競合の可能性について即座にフィードバックを受け取り、コードが生成される前に依存関係を解決します。
5. **フル ループのハードウェア デバッグとテレメトリ:** CCStudio IDE では、エージェントがオンチップ デバッグに直接アクセスできるため、自律的なトラブルシューティングが可能になります。エージェントは、ファームウェアの書き込み、ブレークポイントの設定、コール スタックとコア レジスタの確認、シリアル ターミナルのテレメトリのチェックという一連のデバッグ サイクルを実行し、システムの状態を継続的に把握できます。

3 CCStudio IDE 移行スキル

これらの機能を組み合わせると、ベンダー間のデバイス移行など、複雑で価値の高いタスクに取り組む際に最も大きな価値を発揮します。アプリケーションを移行するには、外部コードベースの理解、新しいピン配置へのペリフェラルのマッピング、新しいシリコン上での移植されたアプリケーションの検証が必要です。

CCStudio IDE 移行スキルは、プラットフォームの基盤となるアーキテクチャを使い、専用のオーケストレーション パイプラインによってこのプロセスを自動化します:

移行機能 (図 3-1) は、オーケストレータが一連の専用サブエージェントを統括する構造になっています。各サブエージェントは、移植の一つのフェーズを担当します。

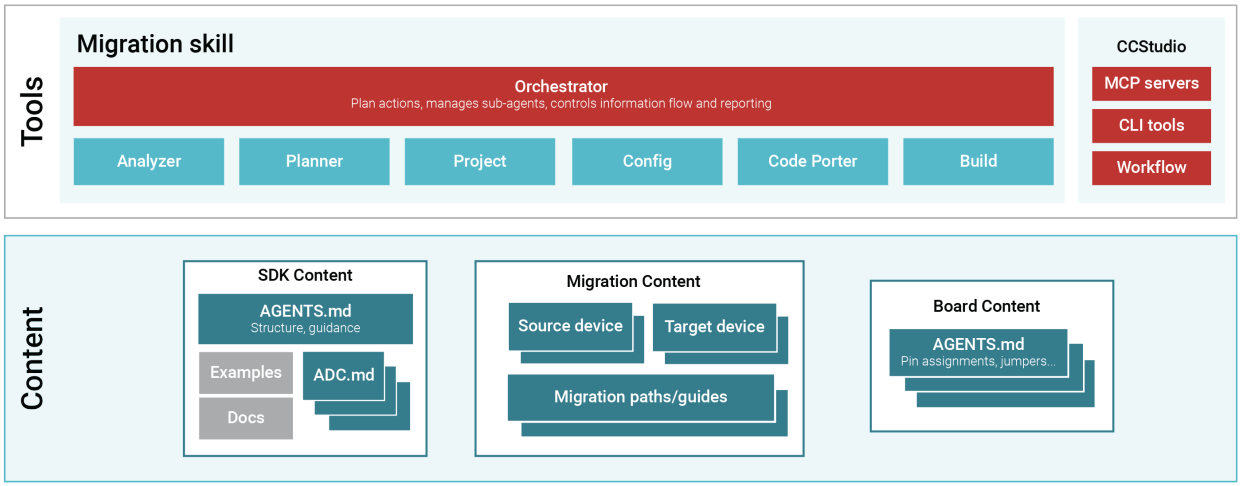


図 3-1. CCStudio IDE 移行ツールのアーキテクチャ

Copyright © 2026 Texas Instruments Incorporated

CCStudio IDE サブエージェントには、以下のものがあります。

- **Analyzer**: 移行元のデバイス、プロジェクトの目的、使用しているペリフェラルを特定します。
- **Planner**: 領域ごとにまとめた移植計画を順序立てて作成します。計画は、ユーザーが確認して調整できるようになっています。
- **Project**: 新しい CCStudio プロジェクトを作成し、デバイスに依存しないアプリケーション ファイルをコピーします。
- **config**: 複数のサブエージェントを連携させて、ペリフェラルとソフトウェアの設定を行います。
- **Code Porter**: こちらも複数のサブエージェントを使って、ソース コードと API の使用箇所を移行先デバイスの SDK に移行します。
- **Build**: プロジェクトをビルドし、ビルド エラーが解決するまで繰り返し処理します。

このオーケストレータは、前述の MCP サーバー、CLI ツール、ワークフロー ガイダンスに加えて、別途管理されている構造化コンテンツも利用します。この構造化コンテンツには、SDK のドキュメントとサンプル、移行元デバイスと移行先デバイスの移行ガイド、ボード固有のピンとジャンパの情報が含まれています。このコンテンツをオーケストレーション ロジックから切り離しておくことで、ドキュメントやサンプルが改善されるにつれて、移行の品質も向上します。

4 移行スキルによる結果の検証方法

移行したプロジェクトを作成するだけでは、作業の半分にすぎません。コードの移行自体には長い時間 (場合によっては数時間) がかかることがあります。そのため、AI 環境で問題が発生した場合や、プロセスを途中で停止する必要がある場合に備えて、プロセスを再開できるように設計されています。移行が完了した後も、作成したアプリケーションが実際のハードウェアで動作することを実証する必要があります。これは、コードの移行後に呼び出される別のテスト移行スキルが、次の三つの部分からなるプロセスで処理します。

- **Orchestrator:** 使用可能なテスト機器を判断し、必要なセットアップ (配線、ジャンパ) をユーザーに案内します。
- **Planner:** 移行レポートを読み取り、優先順位を付けたテスト計画を作成します。
- **Executor:** 各テストを実行し、失敗した場合は修正を適用して再ビルドし、再試行します (試行回数には上限があります)。それでも解決しない場合は、問題を人間に報告します。

結果は受信するたびにログに記録されるため、テストの進捗状況を把握できます。このアプローチは、ベンチにある機器に合わせて調整することもできます。既存のテスト フレームワークや、ロジック アナライザやシグナル ジェネレータなどのラボ機器を駆動できるほか、ターゲット上でのデバッグやシリアル通信も行えます。移行スキルは、AI 環境とは切り離して再利用できるテスト スイートを作成します。

5 まとめ

エージェント型の組込み開発を効果的かつ効率的に行うには、エージェントが制御する物理的なハードウェアに AI を確実に接続できるツールを利用できる必要があります。TI は、構造化されたドキュメントとハードウェアへの直接のデバッグ アクセスを CCStudio IDE に組み込むことで、日常的な組み込みエンジニアリング タスクの処理に必要なものを AI エージェントに提供します。これにより、開発者はシステム レベルの設計判断により多くの時間を割けるようになります。

Copyright © 2026 Texas Instruments Incorporated

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むいかなる保証も、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