

Application Note

TI SysConfig に統合された AM13x モデル ベースの設計



Arun Munuswamy, Nilabh Anand, Val Singh

概要

組込み制御アプリケーションの複雑化に伴い、アルゴリズム設計の加速、早期段階でのテストの実現、市場投入までの時間の短縮を推進できる開発手法が求められています。この課題に対処する強力なパラダイムとして、MATLAB/Simulink を用いたモデル ベース設計 (MBD) が登場しました。これにより、エンジニアは量産対応の組込み C コードの設計、シミュレーション、自動生成を行い、開発サイクルの初期段階で設計エラーを検出することができます。

同時に、テキサス インストルメンツのシステム構成ツール (SysConfig) は、TI のマイコン (? EP) 製品ラインアップ全体にわたり、ペリフェラルやクロックの構成を行うための統合グラフィカル インターフェイスとしての地位を確立しています。

SysConfig は、ピン多重化からドライバの設定に至るまで、一貫性ある単一の構成環境を提供します。これにより、手書きの初期化ボーイプレート コードを排除できるだけでなく、ハードウェア構成の意図を明確に記録して、プロジェクトやチーム間で共有および再利用することが可能となります。このアプリケーション ノートでは、[テキサス インストルメンツ Am13x 向け Embedded Coder サポート パッケージ](#)におけるこれら 2 つの補完技術の統合について説明します。

目次

1 概要.....	2
2 前提条件と入門.....	2
3 MATLAB および Simulink との SysConfig 統合.....	2
4 モデルの構造と SysConfig エントリ ポイント.....	3
5 ビルドと導の概要.....	5
5.1 Codegen の構造.....	5
5.2 CCS での Simulink 生成コードのデバッグとカスタマイズ.....	6
6 リファレンス モーター制御設定の SysConfig.....	7
6.1 AM13x Simulink モデル.....	7
6.2 SysConfig 内の PWM 構成.....	8
6.3 SysConfig 内の ADC 構成.....	9
7 まとめ.....	10
8 参考資料.....	10

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

この統合の中核となるイネーブラは、MATLAB/Simulink モデルと Code Composer Studio (CCS) IDE プロジェクトの間で共有 **syscfg** 構成ファイルです。この共通の構成により、ペリフェラル設定、ピン構成、クロック定義が両方のワークフロー間で整合および同期していることが検証されます。これにより、冗長設定が排除され、シミュレーションとハードウェア間での構成の不整合が生じるリスクが低減されます。

この相互運用性は、従来型の手書きコードによる開発からモデル ベース設計へと移行するチームにとって自然な橋渡しとなります。あるいは逆に、**CCS** 環境で作業する組込みソフトウェア エンジニアに成果物を引き渡す必要のある、または共同作業を行う必要のあるモデル ベース開発者にとっても有用です。**Simulink** で開発および検証されたアルゴリズム IP は、基盤となるハードウェア構成を再定義することなく、**CCS** ベースの統合環境 (またはその逆) にシームレスに移行できます。

その結果、統合に伴う摩擦が大幅に低減され、分野横断的な並行開発が促進されて、最終的に **AM13x** デバイス上で検証済みかつ展開可能な組込みアプリケーションを構築するまでのプロセスを加速させる一貫性ある柔軟な開発エコシステムが構築されます。

2 前提条件と入門

- インストール手順と必要なツールについては、[GitHub - TexasInstruments/am13x-hsp](#) を参照してください
- ビデオ チュートリアルと FAQ (よくある質問) は、「[\[FAQ\] テキサス インストルメンツ AM13x 向け Embedded Coder サポート パッケージ](#)」に含まれています

3 MATLAB および Simulink との SysConfig 統合

従来、モデルの設定ではペリフェラルや他のリソースの構成も行っていましたが、現在では **SysConfig** がこれらの機能をすべて提供しているため、ユーザーは既存の **SDK** サンプルから取得するか、**Code Composer Studio** 内で対話的に作成した **.syscfg** ファイルを用意するだけで済みます。**Embedded Coder** のビルド プロセスでは、**SysConfig** ツールを起動して、必要なすべてのペリフェラル初期化コードを生成します。この **SysConfig** で生成されたコードは、**Simulink** で生成されたアルゴリズムとスケジューラ コードとリンクされており、アプリケーション バイナリ全体が形成されます。

4 モデルの構造と SysConfig エントリ ポイント

この [led の例](#) は、SysConfig との統合について説明する最も簡単な例の 1 つです

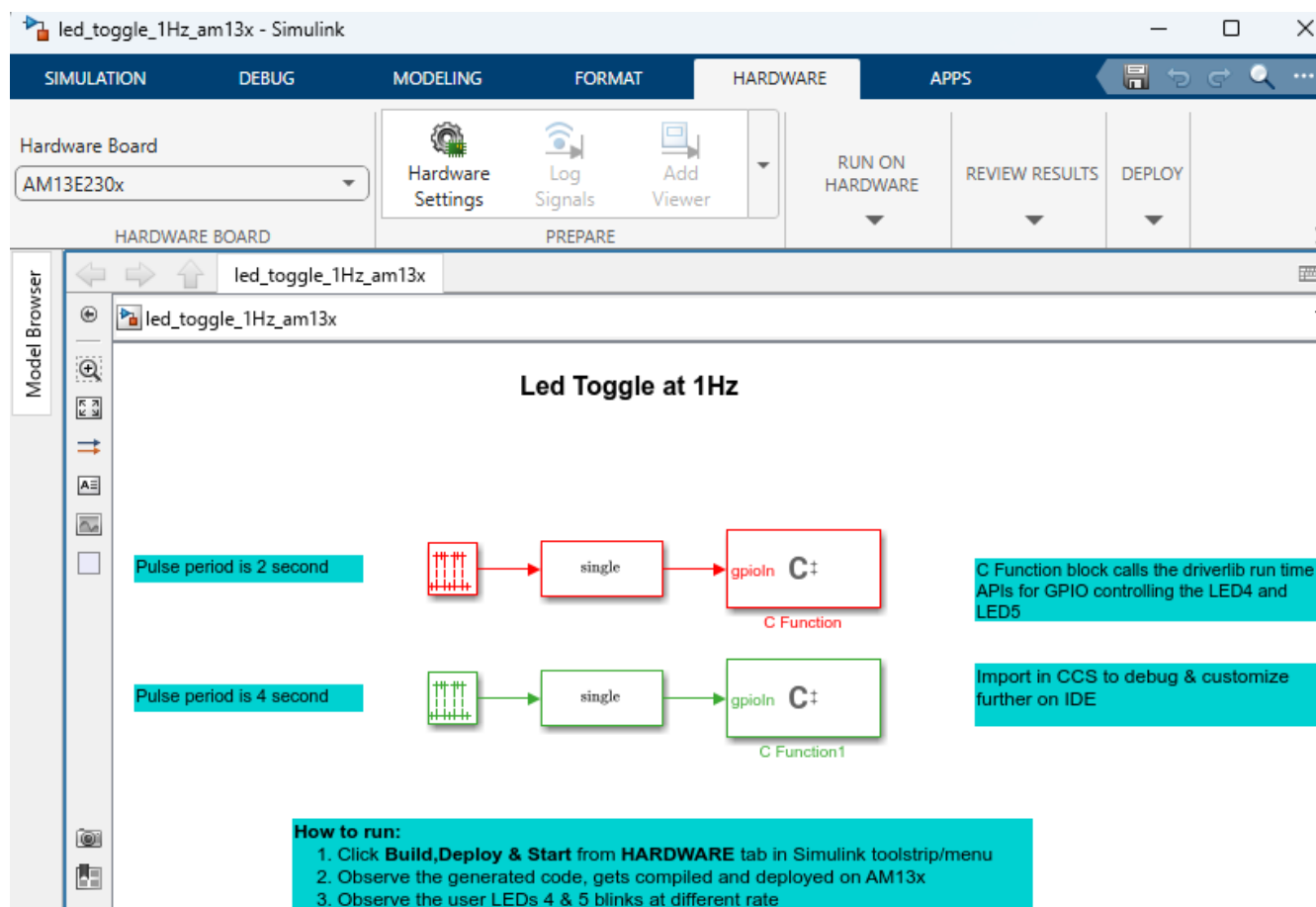


図 4-1. Led トグル プログラムの Simulink ブロック

SysConfig エントリ ポイントを示すハードウェア設定/モデル設定:

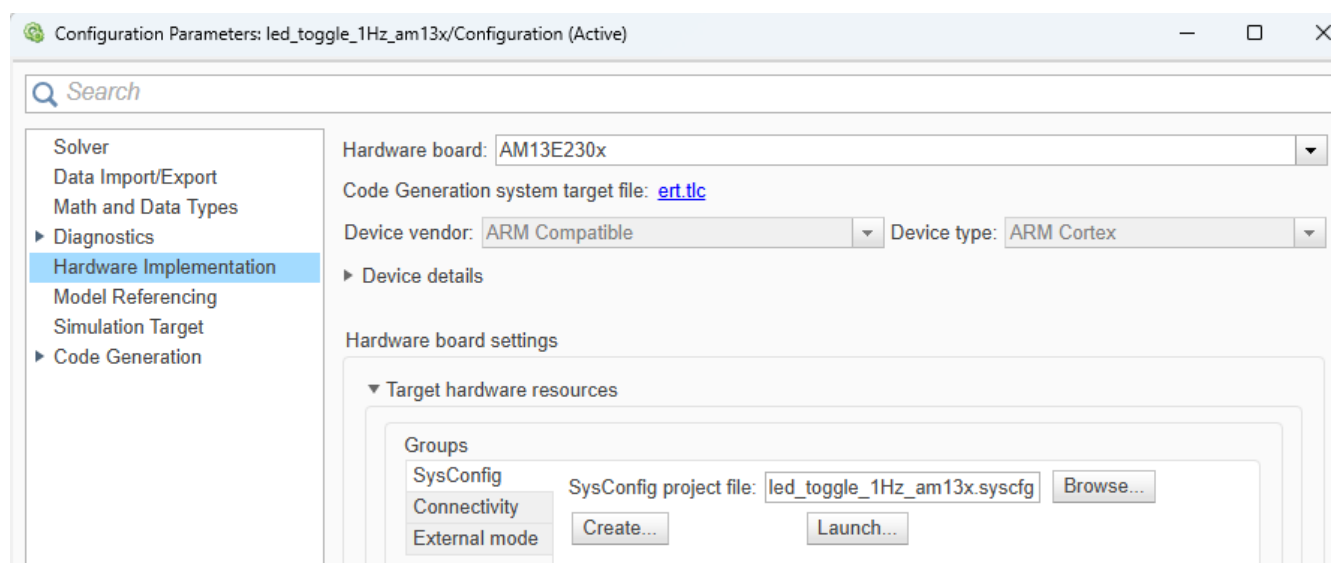


図 4-2. ハードウェア ターゲットの選択

SysConfig を Simulink から起動し、2 つの LED 構成を表示します

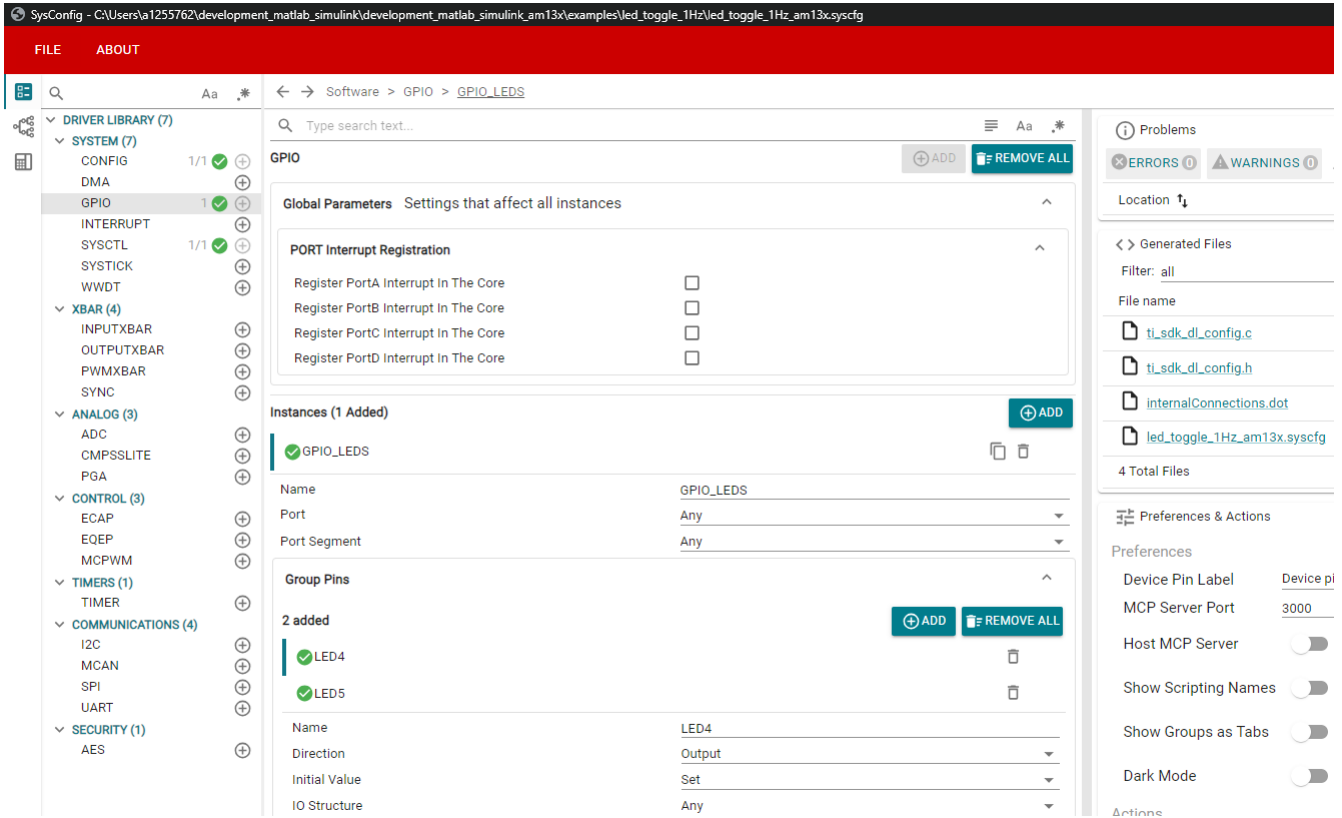
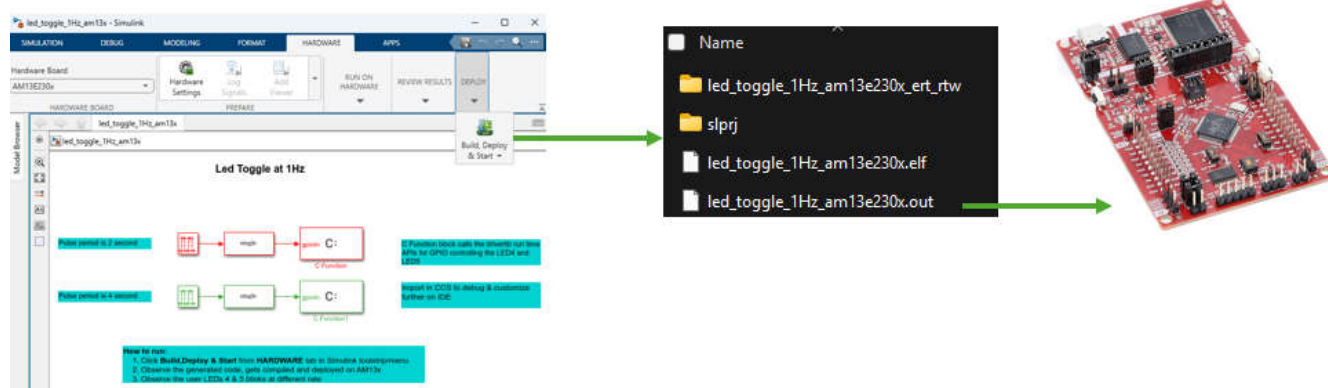


図 4-3. GPIO Sysconfig

5 ビルドと導の概要

ユーザーが Simulink で「**Build, Deploy & Start**」をクリックすると、接続されている AM13x Launchpad へのコード生成、コンパイル、展開が 1 回の操作で自動的に実行されます。生成された出力フォルダには、アプリケーションのビルド全体に必要なすべてのアーティファクトが含まれています。



5.1 Codegen の構造

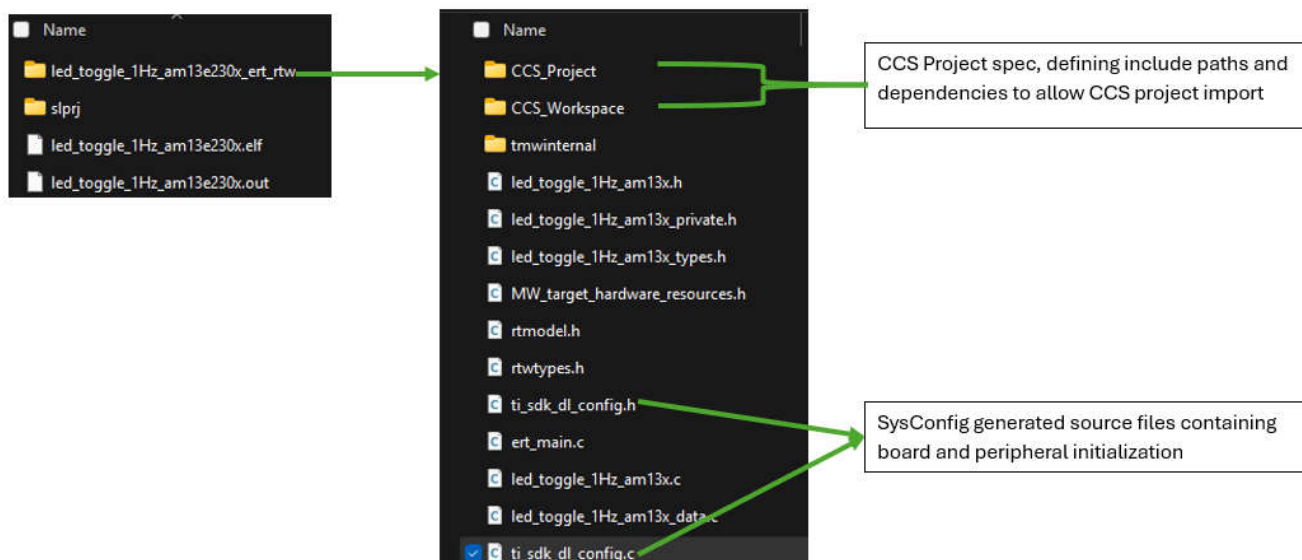


図 5-1. CCS プロジェクトの構造

ert_main.c で定義されているアプリケーション エントリ ポイント main 関数の簡単な説明を示します

```
1  int main(void)
2  {
3      /* Step 1: SysConfig-generated peripheral initialization */
4      SYSCFG_DL_init();
5
6      /* Step 2: Simulink-generated model initialization */
7      led_toggle_initialize();
8
9      /* Step 3: HSP scheduler initialization */
10     am13x_tick_init(BASE_RATE_TICKS);
11
12     /* Step 4: Enable interrupts – execution driven by SysTick_Handler */
13     __enable_irq();
14
15     while (1)
16     {
17         /* Idle loop */
18     }
19 }
20
21 /* Simulink-generated scheduler ISR (am13x_tick.c) */
22 void SysTick_Handler(void)
23 {
24     rt_OneStep(); /* Invokes led_toggle_step() at base sample time */
25 }
```

図 5-2. プログラム フロー

5.2 CCS での Simulink 生成コードのデバッグとカスタマイズ

このセクションの説明は、「[検証と確認](#)」に記載されています。

6 リファレンス モーター制御設定の SysConfig

Led の例と同様に、**モーター制御の例**には、ADC、PWM、割り込み、GPIO などの多くのペリフェラルが使用されるより複雑な FoC ベースのモーター制御アプリケーションのためのモデルと SysConfig ファイルが含まれています。

6.1 AM13x Simulink モデル

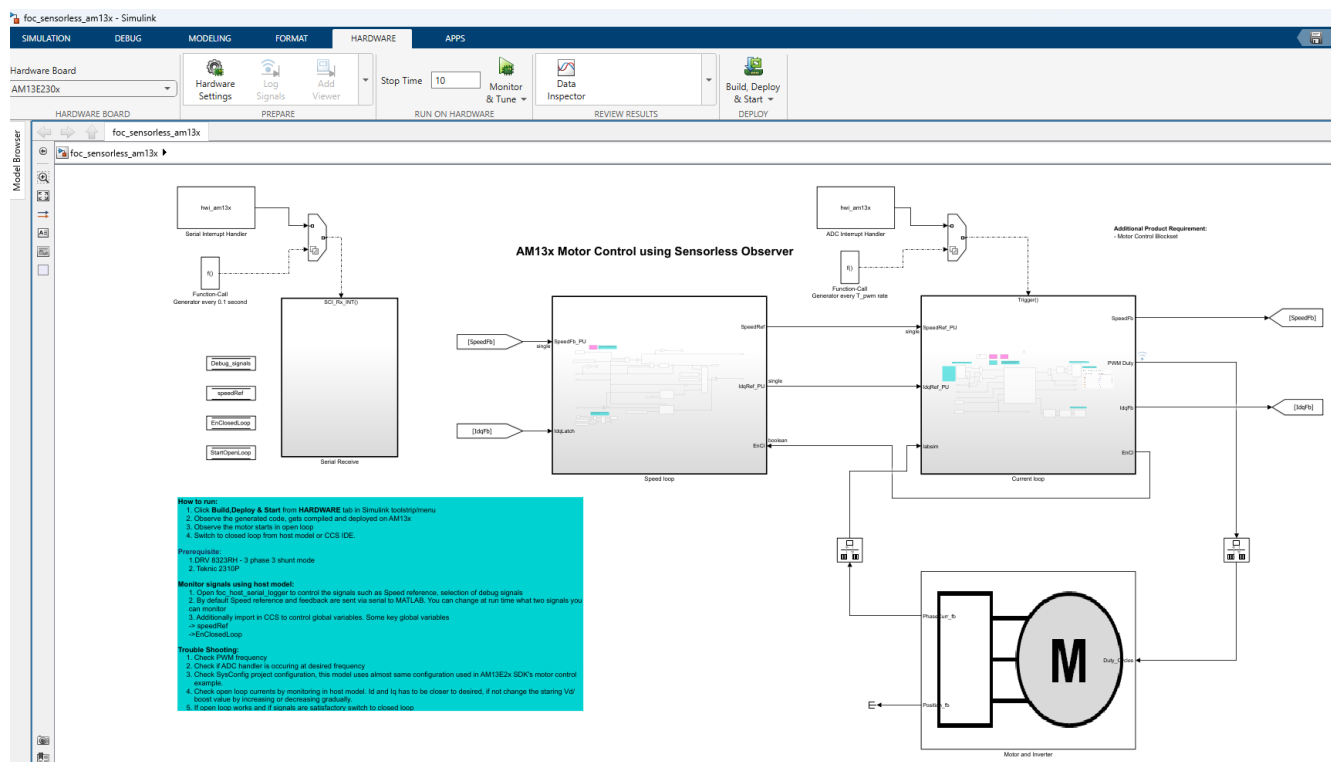


図 6-1. モーター制御 FoC の Simulink ブロック

6.2 SysConfig 内の PWM 構成

The screenshot displays the SysConfig application window. The top menu bar includes 'FILE' and 'ABOUT'. The left sidebar shows a tree view of components under 'DRIVER LIBRARY (7)', 'SYSTEM (7)', 'XBAR (4)', 'ANALOG (3)', 'CONTROL (3)', 'TIMERS (1)', 'COMMUNICATIONS (4)', and 'SECURITY (1)'. The 'MCPWM' component is selected under 'CONTROL (3)'. The main configuration area shows the 'MCPWM (1 of 5 Added)' configuration for 'MTR1_MCPWM_BASE'. The configuration includes fields for Name, MCPWM Instance, Selected Instance, Add External Pins, and Register Interrupt In The Core. Below this is the 'MCPWM Submodules' section with a table for adding/removing submodules. The 'MCPWM Time Base' section includes fields for Time Base Clock Divider, Time Base Clock Frequency, Counter Mode, Counter Direction After Sync, Time Base Period, PWM Frequency, Time Base Counter Initial Value, Enable Phase Shift Load, Sync In Pulse Source, SYNCPER Source Select, and Sync Out Pulse.

MCPWM Submodules	
Add / Remove Submodules	
Add Action Qualifiers for PWM1A	☒
Add Action Qualifiers for PWM1B	☒
Add Action Qualifiers for PWM2A	☒
Add Action Qualifiers for PWM2B	☒
Add Action Qualifiers for PWM3A	☒
Add Action Qualifiers for PWM3B	☒
Add Dead Band submodule	☒
Add Trip Zone submodule	☐
Add Event Trigger submodule	☒

MCPWM Time Base	
Time Base Clock Divider	Divide clock by 1
Time Base Clock Frequency	200.00 MHz
Counter Mode	Up-down count mode
Counter Direction After Sync	Count down after sync event
Time Base Period	5000
PWM Frequency	20.00 kHz
Time Base Counter Initial Value	0
Enable Phase Shift Load	☐
Sync In Pulse Source	Disable Sync-in
SYNCPER Source Select	Sync Peripheral Signal Pulse is generated when time ba...
Sync Out Pulse	Sync pulse is disabled

The right sidebar shows 'Problems' with 0 errors and 0 warnings. Below it, 'Generated Files' are listed: [ti_sdk_dl_config.c](#), [ti_sdk_dl_config.h](#), [internalConnections.dot](#), and [foc_sensorless_am13x.syscfg](#). A total of 4 files are generated.

図 6-2.

6.3 SysConfig 内の ADC 構成

The screenshot displays the TI SysConfig interface for configuring the ADC. On the left, the component tree shows 'DRIVER LIBRARY (7)', 'SYSTEM (7)', 'XBAR (4)', 'ANALOG (3)', 'CONTROL (3)', 'TIMERS (1)', 'COMMUNICATIONS (4)', and 'SECURITY (1)'. The 'ANALOG' section is expanded, showing 'ADC' (3/3 added), 'CMPSSLITE', and 'PGA'. The 'ADC' component is selected, and its configuration is shown in the main area. The configuration includes 'ADC (3 of 3 Added)' with 'ADC_0', 'ADC_1', and 'ADC_2' instances. The 'SOC0' configuration is detailed, showing 'SOC Name' as 'MTR1_IW_SEN', 'Independent Name Mode' checked, and 'Channel' as 'Channel 12 / PA17'. The 'INT Configurations' section shows 'ADC Interrupt Pulse Mode' as 'End of conversion', 'ADC Interrupt Cycle Offset' as '0', 'Enable ADC Interrupts' as 'INT1', and 'Enable DMA Interrupts' as 'None'. The right sidebar shows 'Generated Files' including 'ti_sdk_dl_config.c', 'ti_sdk_dl_config.h', 'internalConnections.dot', and 'foc_sensorless_am13x.syscfg'.

図 6-3.

7 まとめ

このアプリケーション ノートでは、AM13x マイコン ファミリーを対象として、TI SysConfig を MATLAB/Simulink モデル ベース設計ワークフローに統合するための構造化手法について説明しました。SysConfig は、すべてのペリフェラル/マイコン構成の単一の定義として機能し、手書きコードとモデル生成コードによる組込みソフトウェア間の整合性を検証します。

8 参考資料

テキサス インストルメンツ、『[MATLAB ブロックセットトレーニング](#)』、トレーニング。

テキサス インストルメンツ、『[\[FAQ\] テキサス インストルメンツ AM13x 向け Embedded Coder サポート パッケージ](#)』、FAQ

テキサス インストルメンツ、[AM13E2X-SDK](#)、製品ページ。

テキサス インストルメンツ、『[AM13E2X-HSP](#)』ハードウェア サポート パッケージ。

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