

Application Note

集成 TI SysConfig 的 AM13x 基于模型设计



Arun Munuswamy, Nilabh Anand, Val Singh

摘要

嵌入式控制应用日益复杂，要求开发方法能够加速算法设计、支持早期测试并缩短产品上市时间。采用 MATLAB/Simulink 的基于模型设计 (MBD) 已成为应对这一挑战的强大范式，使工程师能够设计、仿真并自动生成可用于生产的嵌入式 C 代码，从而在开发周期早期发现设计错误。

与此同时，德州仪器 (TI) 的系统配置工具 (SysConfig) 已成为覆盖整个 TI 微控制器 (MCU) 产品系列的统一外设与时钟配置图形界面。通过提供从引脚多路复用到驱动程序设置的单一、一致的配置体验，SysConfig 省去了手写初始化样板代码，并确保硬件配置意图被清晰捕获，且可在工程与团队之间共享和复用。本应用手册介绍了这两种互补技术在德州仪器 (TI) AM13x Embedded Coder 支持包中的集成。

内容

1 简介.....	2
2 前提条件与入门.....	2
3 SysConfig 与 MATLAB 和 Simulink 的集成.....	2
4 模型结构与 SysConfig 入口点.....	3
5 构建与部署概述.....	5
5.1 代码生成结构.....	5
5.2 在 CCS 中调试和定制 Simulink 生成的代码.....	6
6 一种参考电机控制设置的 SysConfig.....	7
6.1 AM13x Simulink 模型.....	7
6.2 SysConfig 中的 PWM 配置.....	8
6.3 SysConfig 中的 ADC 配置.....	9
7 总结.....	10
8 参考资料.....	10

商标

所有商标均为其各自所有者的财产。

1 简介

实现这一集成的一个核心要素在于 MATLAB/Simulink 模型与 Code Composer Studio (CCS) IDE 工程之间共享的 `syscfg` 配置文件。这一共享配置可确保外设设置、引脚分配和时钟定义在两种工作流程中保持一致并同步，从而消除冗余设置，降低仿真与硬件之间配置不匹配的风险。

这种互操作性为团队从传统手写代码开发过渡到基于模型的设计搭建了一座天然桥梁，反过来也可让基于模型的开发者和在 CCS 中工作的嵌入式软件工程师进行交接或协作。在 Simulink 中开发并验证过的算法 IP 可无缝迁移到基于 CCS 的集成环境中，反之亦然，无需重新定义底层硬件配置。

最终形成了一个兼具凝聚力和灵活性的开发生态系统，显著减少了集成阻力，支持跨学科的并行开发，并最终加快了 AM13x 器件从概念到经过验证、可部署的嵌入式应用的整个过程。

2 前提条件与入门

- 安装步骤及所需工具的详细说明见 [GitHub - TexasInstruments/am13x-hsp](#)
- 视频教程和常见问题解答见 [【常见问题解答】德州仪器 \(TI\) AM13x Embedded Coder 支持包](#)

3 SysConfig 与 MATLAB 和 Simulink 的集成

传统上，模型设置也会配置外设和其他资源。由于 SysConfig 提供了所有这些功能，现在用户只需提供一个 `.syscfg` 文件，该文件可来自现有的 SDK 示例，也可在 Code Composer Studio 中以交互方式创建。

Embedded Coder 构建过程会启动 SysConfig 工具，以生成所有必要的外设初始化代码。SysConfig 生成的代码与 Simulink 生成的算法和调度器代码链接在一起，最终生成完整的应用程序二进制文件。

4 模型结构与 SysConfig 入口点

这个 LED 示例是说明与 SysConfig 集成的最简单示例之一

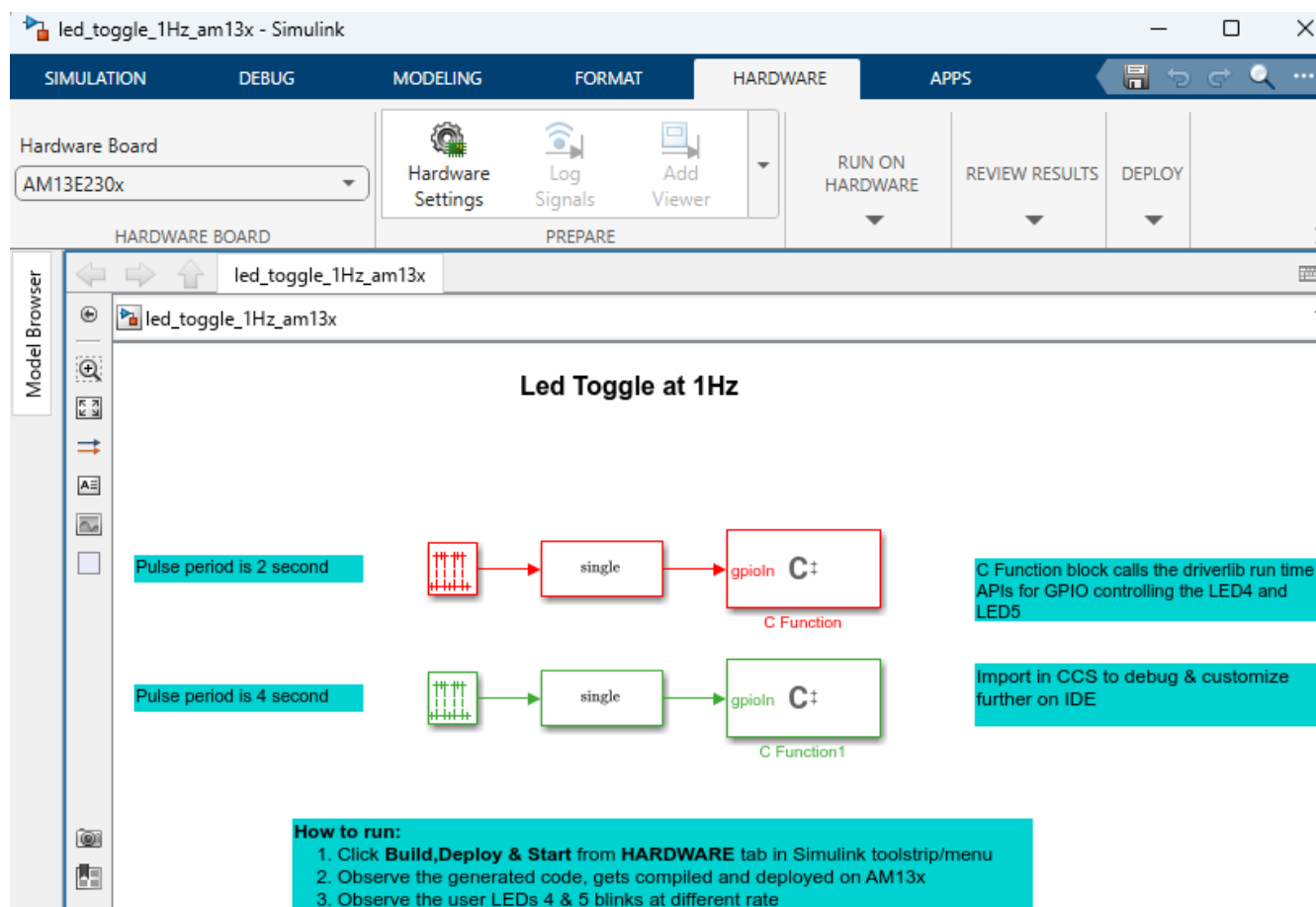


图 4-1. 用于 LED 切换程序的 Simulink 模块

显示 SysConfig 入口点的硬件设置/模型设置：

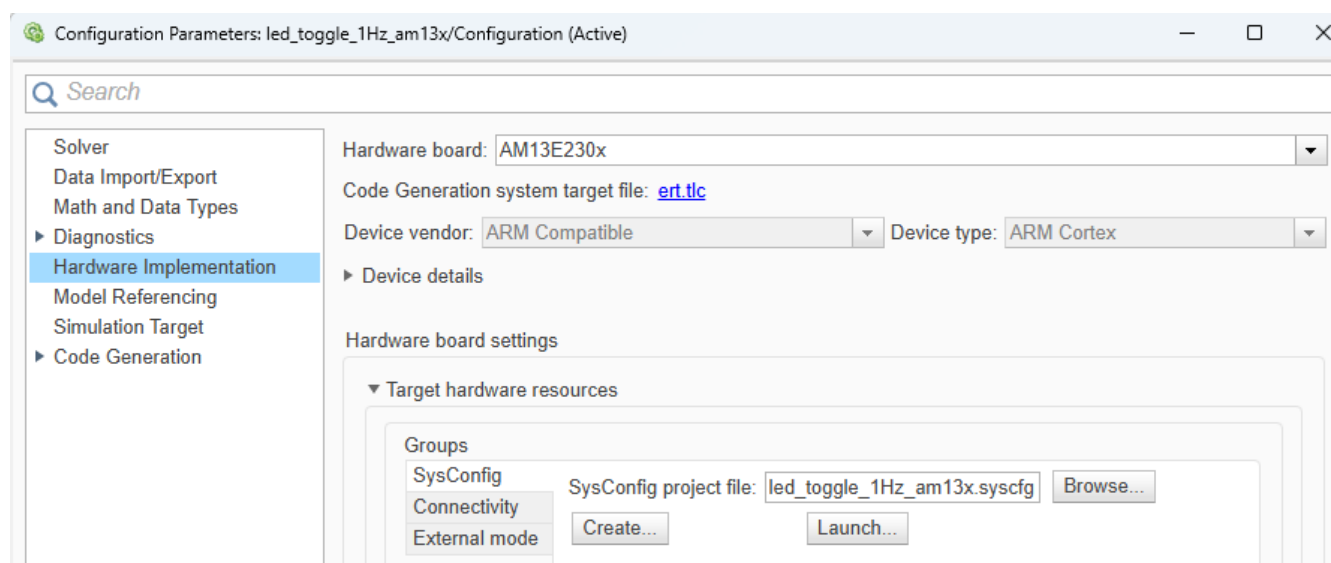


图 4-2. 硬件目标选择

从 Simulink 启动的 SysConfig 显示双 LED 配置

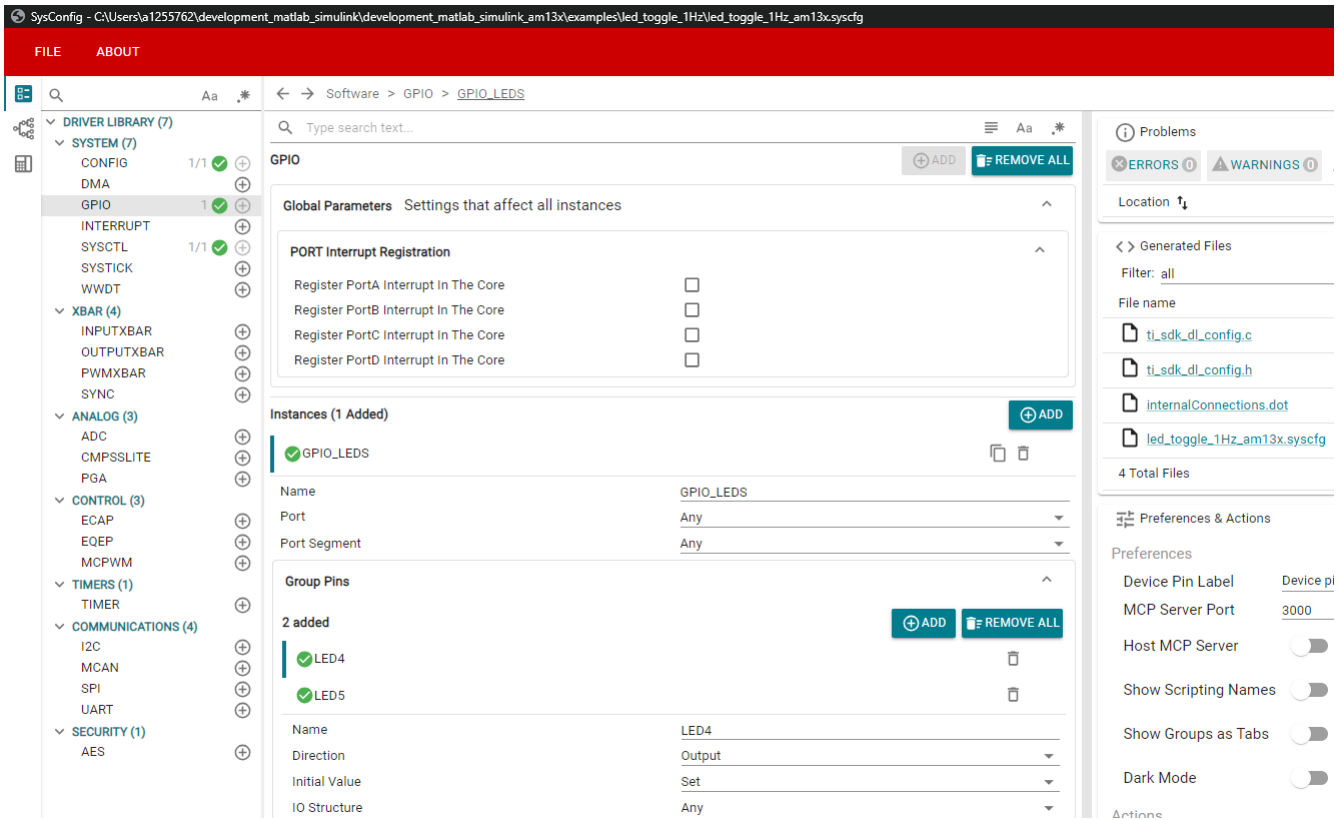
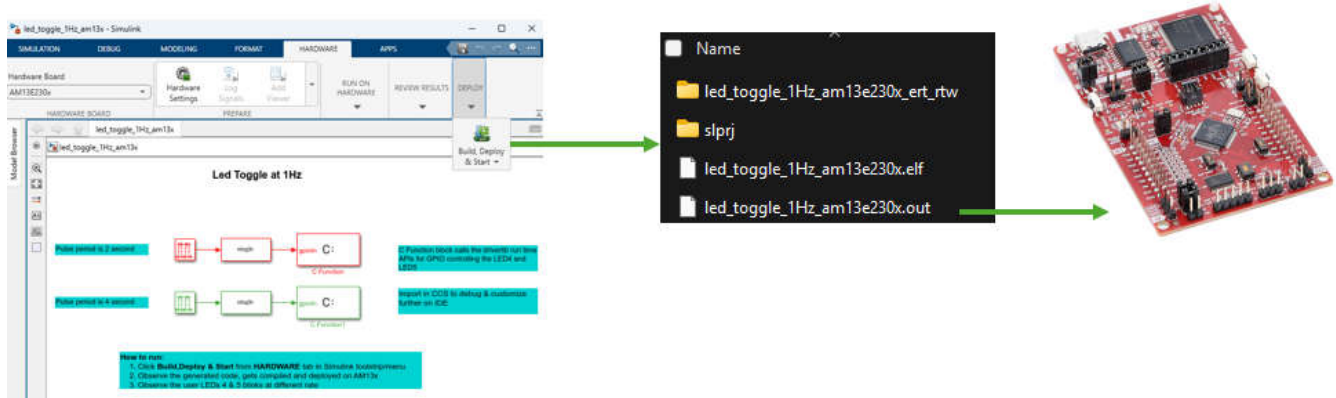


图 4-3. GPIO SysConfig

5 构建与部署概述

当用户在 Simulink 中点击 **Build, Deploy & Start** 时，代码生成、编译以及部署到连接的 AM13x Launchpad 会在一次操作中自动完成。生成的输出文件夹包含完整的应用程序构建所需的所有产物，结构如下：



5.1 代码生成结构

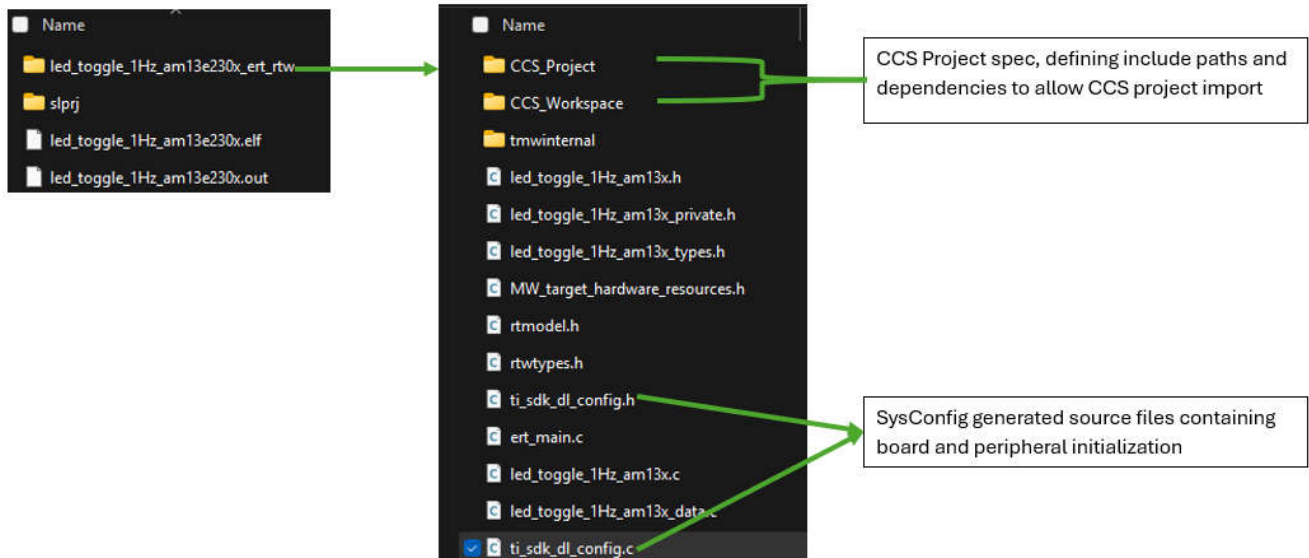


图 5-1. CCS 工程结构

以下是对 `ert_main.c` 中定义的应用程序入口点 `main` 函数的简化说明

```
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 }
```

图 5-2. 程序流程

5.2 在 CCS 中调试和定制 Simulink 生成的代码

本节在 [验证和确认](#) 中说明。

与 LED 示例类似，[电机控制示例](#)包含基于 FoC 的更复杂电机控制应用的模型和 SysConfig 文件，其中使用了更多外设，如 ADC、PWM、中断、GPIO。

SIMULATION DEBUG MODELING FORMAT HARDWARE APPS

Hardware Board: foc_sensorsless_am13x

Stop Time: 10 Monitor & Tune Data Inspector Build, Deploy & Start DEPLOY

HARDWARE BOARD RUN ON HARDWARE REVIEW RESULTS

Model Browser

foc_sensorsless_am13x

AM13x Motor Control using Sensorless Observer

Serial Interrupt Handler
 Function Call
 Generate every 0.1 second
 Serial Receiver
 Debug_signals
 speedRef_PU
 speedRef_N
 speedRef_R
 StartOpenLoop

Speed loop
 SpeedRef_PU
 SpeedRef_N
 SpeedRef_R
 Iref_PU
 Iref_N
 Iref_R

Current loop
 SpeedRef_PU
 Iref_PU
 Iref_N
 Iref_R

Motor and Inverter
 Duty
 Motor

Additional Product Requirement:
 - Motor Control Blockset

How to run:

- Click **Build, Deploy & Start from HARDWARE** tab in Simulink tooling menu
- Observe the generated code, gets compiled and deployed on AM13x
- Observe the motor starts in open loop
- Switch to closed loop from host model or CCS IDE.

Prerequisites:

- DRV 8325BH - 3 phase 3 shunt mode
- Taltech 2210P

Monitor signals using host model:

- Open foc_host_serial_logger to control the signals such as Speed reference, selection of debug signals
- By default Speed reference and feedback are sent via serial to MATLAB. You can change at run time what two signals you can monitor
- Additionally import in CCS to control global variables. Some key global variables
 -> speedRef_PU
 -> speedRef_N
 -> speedRef_R

Preload Shooting:

- Check PWM frequency
- Check FOC handler is occurring at desired frequency
- Check SysConfig project configuration, this model uses almost same configuration used in AM132x BDI's motor control examples
- Check open loop currents by monitoring in host model, Id and Iq has to be closer to desired, if not change the starting Vd/Iq boost value by increasing or decreasing gradually
- Propose loop gains and if necessary, add selective switch to closed loop

图 6-1. 用于电机控制 FoC 的 Simulink 模块

6.2 SysConfig 中的 PWM 配置

The screenshot displays the SysConfig application window. The left sidebar shows a tree view of components, with 'MCPWM' selected under the 'CONTROL' category. The main configuration area is titled 'MCPWM (1 of 5 Added)' and shows the 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, the 'MCPWM Submodules' section lists various submodules with checkboxes for adding or removing them. The 'MCPWM Time Base' section contains settings 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.

Generated Files:

- ti_sdk_dl_config.c
- ti_sdk_dl_config.h
- internalConnections.dot
- foc_sensorless_am13x.syscfg

4 Total Files

图 6-2.

6.3 SysConfig 中的 ADC 配置

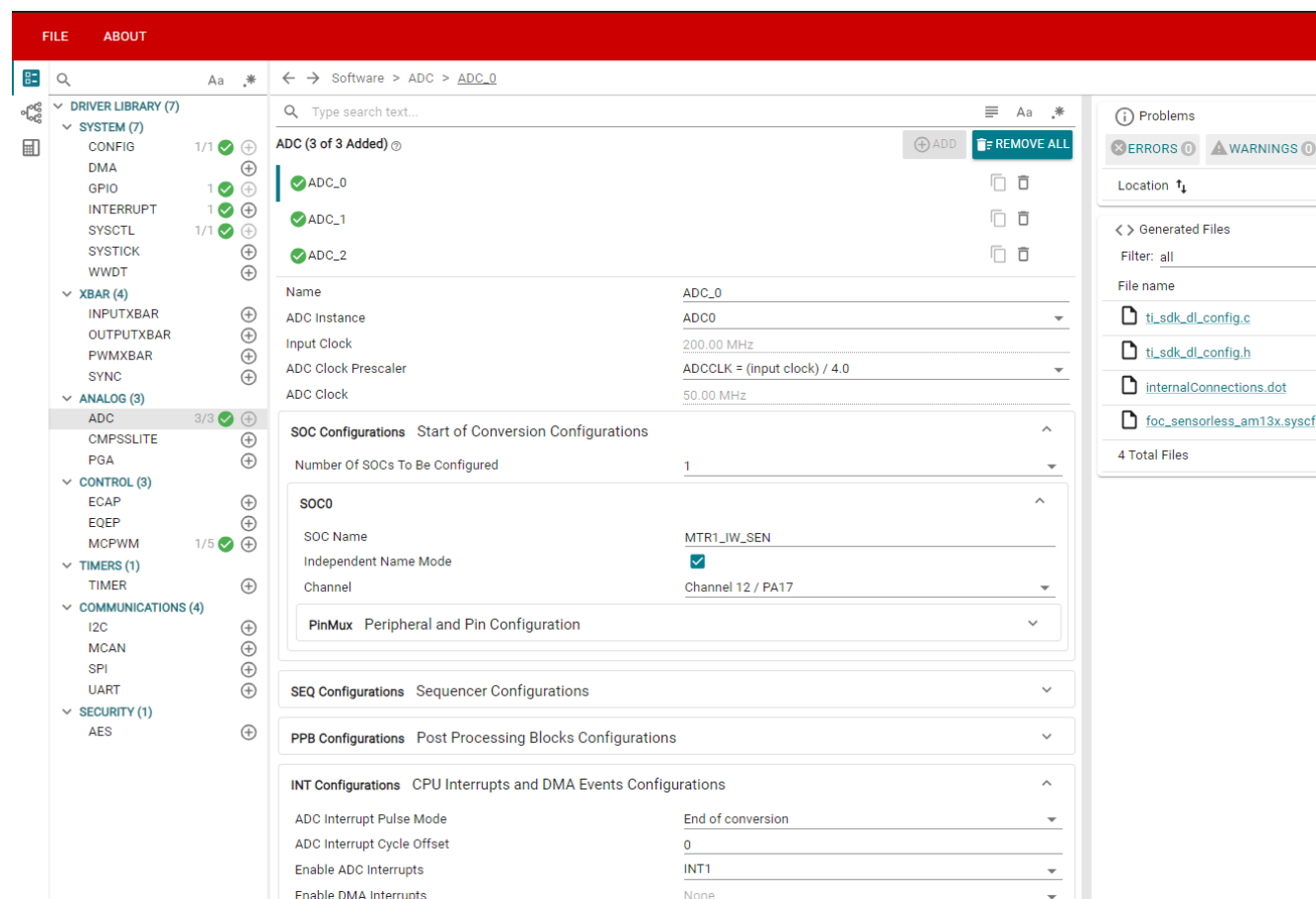


图 6-3.

7 总结

本应用手册介绍了一种将 TI SysConfig 与面向 AM13x 微控制器系列的 MATLAB/Simulink 基于模型设计工作流程相集成的结构化方法。SysConfig 作为所有外设/MCU 配置的单一来源，可验证手写代码与模型生成嵌入式软件之间的一致性。

8 参考资料

德州仪器 (TI)，[Matlab 模块集培训](#)，培训。

德州仪器 (TI)，[【常见问题解答】德州仪器 \(TI\) AM13x Embedded Coder 支持包](#)，常见问题解答。

德州仪器 (TI)，[AM13E2X-SDK](#) 产品页面。

德州仪器 (TI)，[AM13E2X-HSP](#)，硬件支持包。

重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、与某特定用途的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他安全、安保法规或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#)、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月