

Application Note

《C2000 IDE Assist 工具特性指南》



Nima Eskandari, Aishwarya Rajesh, Delaney Woodward, Shashank Madineni

摘要

C2000™ IDE Assist 是集成在 Visual Studio Code™ 和 Code Composer Studio™ 中的首创工具，旨在增强德州仪器 (TI) C2000 MCU 的开发。C2000 IDE 利用您开发环境中已有的功能，提供了一种更简便的方法来编写、调试和迁移 C2000 应用程序代码。这款多合一工具可加速软件生产过程的各个步骤，从代码初始化到运行时事件再到调试系统配置，比如轻松迁移、有针对性的配套资料查找能力、软件模板支持等。集成在 Code Composer Studio 中的 C2000 IDE 可让您轻松访问各种工程工具、从而使该扩展成为 C2000 应用实现的全面解决方案。

内容

1 简介.....	2
2 入门.....	2
2.1 软件安装.....	2
2.2 导入工程.....	4
2.3 检测项目.....	4
2.4 基本命令.....	4
2.5 全局设置.....	5
3 概述.....	6
4 特性.....	8
4.1 定向配套资料交付.....	8
4.2 开发人员效率工具.....	9
4.3 迁移支持.....	10
4.4 命令列表.....	14
5 总结.....	16
6 参考资料.....	17

商标

C2000™, Code Composer Studio™, and E2E™ are trademarks of Texas Instruments.

Visual Studio Code™ is a trademark of Microsoft Corporation.

所有商标均为其各自所有者的财产。

1 简介

德州仪器 (TI) C2000 软件开发套件 (SDK) 提供了一套完整的软件和文档，旨在加速实时控制应用程序的开发。这些资源包括器件专用驱动程序、库和外设示例、可帮助开发人员简化项目。这些 SDK 旨在适应经验水平各异的用户，并会提供适合初学者的功能以指导首次使用的用户。SDK 中包含的 C2000 SysConfig 工具通过直观的可视化界面简化了器件和外设的复杂性，从而完善了设置过程。为了进一步改善初始化、运行时和调试的用户体验，C2000 IDEA 将工具和配套资料集中在单一环境中，以实现更高效的工作流程。

本功能指南讨论了如何上手并利用 C2000 IDE Assist 来简化与 C2000 SDK 一起使用时的整体软件实施。

2 入门

本部分介绍了 C2000 IDEA 工具的基础知识，例如如何设置开发环境以及如何使用扩展的基本命令。

2.1 软件安装

本部分详细介绍了在您的开发环境中启用扩展的软件安装过程。

2.1.1 安装 Code Composer Studio IDE

按照以下步骤安装 TI 的微控制器和处理器集成开发环境 (IDE)：

1. 下载 [Code Composer Studio \(CCS\)](#) 的最新离线安装程序：
 - a. 默认推荐路径 - C:/ti/ccs200X
2. 看到“选择组件”后，请选择“C2000 实时 MCU”。
3. 继续执行这些步骤，直至安装完成。
4. 启动 CCS。

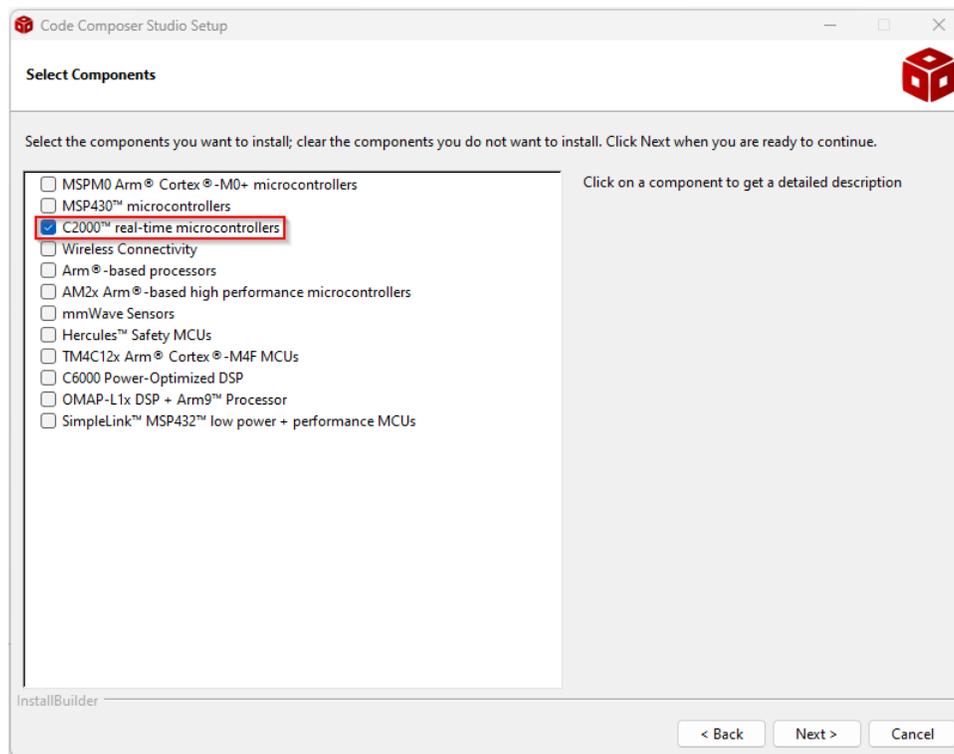


图 2-1. CCS 设置

备注

此扩展仅支持 Code Composer Studio v20.0.0 及更高版本。

2.1.2 安装 C2000 IDE 扩展

遵循以下说明将扩展安装到 CCS：

1. 导航至左侧栏面板中的扩展选项卡。
2. 搜索 C2000 IDE，然后选择“安装”。

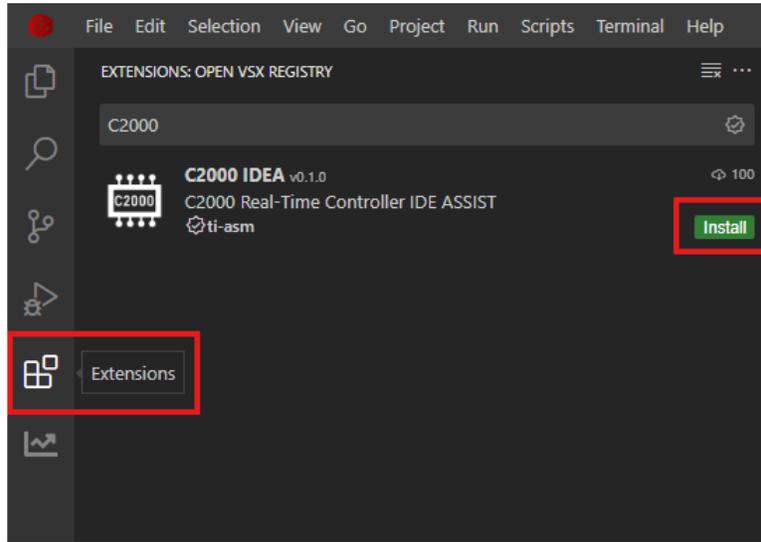


图 2-2. CCS 中的 C2000 IDE 设置

- a. 如果“扩展市场”中没有显示任何结果、则可以将连接配置为使用代理服务器。若要设置代理设置，请执行以下操作：
 - i. 单击左侧栏面板中的齿轮（管理）图标、然后选择设置。
 - ii. 导航至“应用程序”>“代理”，然后输入代理信息。

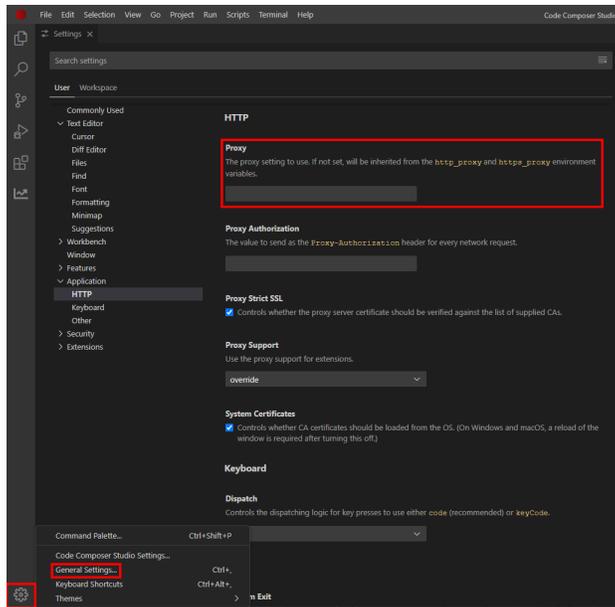


图 2-3. 代理设置

遵循以下说明将扩展安装到 VS Code ：

1. 从 [Open-VSX](#) 下载 VSIX。
 - a. 注意：确保始终下载最新版本。有关该工具的更多信息，请参阅 [C2000-IDEA GitHub 存储库](#)。
2. 导航至左侧栏面板中的 *扩展* 选项卡。
3. 单击三个点并选择 *使用 VSIX 安装*。在文件资源管理器中选择下载的 VSIX。

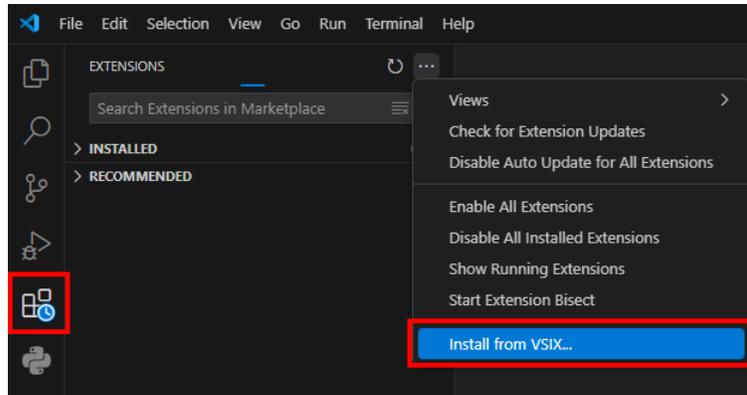


图 2-4. VS Code 中的 C2000 IDEA 设置

刷新 IDE。当屏幕左侧栏面板中出现 *C2000 IDE Assist* 图标时，表示扩展已成功安装。

2.2 导入工程

通过扩展设置代码编辑器平台后，即可将该工具用于 CCS 工程。

如要导入工程，请按照以下步骤操作：

1. 打开 CCS 或 VS Code IDE。
2. 正常打开新的或现有工作区。
3. 选择左侧栏面板上的图标，导航到 CCS 扩展。

2.3 检测项目

只需点击“*获取项目*”元素即可检测到所有使用 C2000 器件的项目。通过此步骤，扩展可以识别每个项目的器件并跟踪每个文件所属的项目。项目检测实时可实现扩展的完整功能，例如跨器件的 *Driverlib* 迁移。某些扩展功能仅在项目检测（或默认器件设置，如 [节 2.5.1](#) 中所述）时可用，而其他功能可以通过要求输入当前器件来在单个文件上运行。有关可用功能和命令的完整列表，请参阅 [节 4.4](#)。

如要检测项目，请按照以下步骤操作：

1. 如要检测工作区中的所有项目、请使用以下两个选项之一：
 - a. 输入 CTRL+SHIFT+P，输入并点击 *C2000: Get Projects*。
 - b. 点击扩展树的 *C2000 IDEA - 功能* 窗格中的“*项目检测*”>“*获取项目*”。
2. 运行后，扩展检测到的工作区中的所有项目都会出现在扩展树的“*C2000 IDEA - 项目*”窗格中。查看您的项目并验证器件型号和当前器件详细信息是否符合预期。

备注

建议在 CCS 内专门使用 C2000 IDEA，以同时利用扩展的所有功能以及 CCS 中提供的完整调试功能。

2.4 基本命令

CCS 和 VS Code 提供了强大的命令面板，允许用户仅使用键盘执行大多数（并非全部）任务。可以使用 CTRL+SHIFT+P 激活这些命令。请参阅下表了解 C2000 IDEA 扩展启用的一些易于使用的命令。或者，可以在 C2000 IDEA 功能窗格中选择有对应 UI 按钮的功能，这会自动运行相关命令来激活该功能。

备注

请参阅本文档中的 [节 4.4](#) 以获取完整的命令列表或在命令面板中搜索“C2000”。

表 2-1. C2000-IDEA 基本命令

用途	命令	器件	项目检测
打开工具演示	C2000 : 实用演示	GEN2、GEN3、GEN4	不需要
运行 Driverlib Register Vision	C2000 : 对当前文件运行 Driverlib Register Vision	GEN3、GEN4	不需要
运行 Bitfield Register Vision	C2000 : 对当前文件运行 Bitfield Register Vision	GEN3、GEN4	不需要
启用 Register Coder	C2000 : 启用寄存器代码写入/读取	GEN3、GEN4	必需
启用 Interrupt Coder	C2000 : 启用中断代码模板	GEN3、GEN4	必需
启用持续迁移检查	C2000 : 对当前文件启用持续迁移检查	GEN3、GEN4	必需

2.5 全局设置

可以根据用户的项目需求使用全局设置。在“扩展 > C2000 IDEA”选项卡的特定 IDE 设置中，有几个配置选项。

2.5.1 无工程检测

当系统未自动检测到器件或未运行工程检测时，用户可以手动选择要使用的默认器件。通过设置默认器件，即使文件并不属于所检测到的工程，也可以使用该工具的大多数功能。换句话说、当在无工程的文件中未自动检测到器件时，该工具会使用用户配置的默认器件。

按照以下步骤在 CCS 和 VS Code 中设置默认 C2000 器件：

1. 点击左侧栏面板中的齿轮（管理）图标，然后选择“设置”选项。
2. 导航至“用户” > “扩展” > “C2000 IDEA”。
3. 在“项目：默认器件”中，选择目标器件。

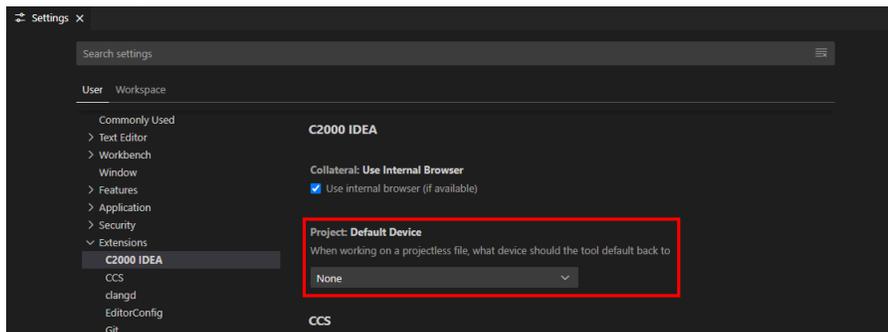


图 2-5. 默认器件选择

备注

TI 建议在利用此扩展进行开发时始终使用最新的 SDK，因为 C2000 IDEA 会与最新的可用 SDK 相关联。

3 概述

此部分概述了 C2000 IDEA 界面以及如何轻松浏览该工具。该工具的功能分别位于屏幕左侧的四个面板中。以下各图展示了使用样本 C2000WARE 示例时该工具的概览。

1. **C2000 配套资料**：该面板展示了关键配套资料，包括技术参考手册和数据表，这些资料与项目和/或当前打开的文件相关联的器件相对应。

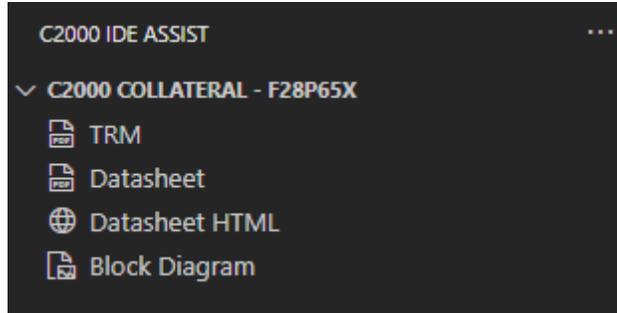


图 3-1. C2000 配套资料

2. **C2000 其他资源**：此面板列出了与特定于器件和外设的项目相关的所有配套资料。我们已将这些配套资料分为基础、入门和专家资料。

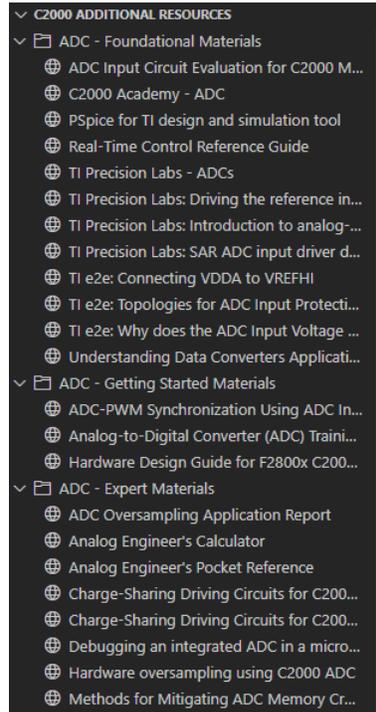


图 3-2. C2000 其他资源

3. **C2000 Idea - 功能**：该面板主要在使用 C2000 IDEA 时使用，因为所有软件开发工具（包括该工具的演练）都托管在这里。

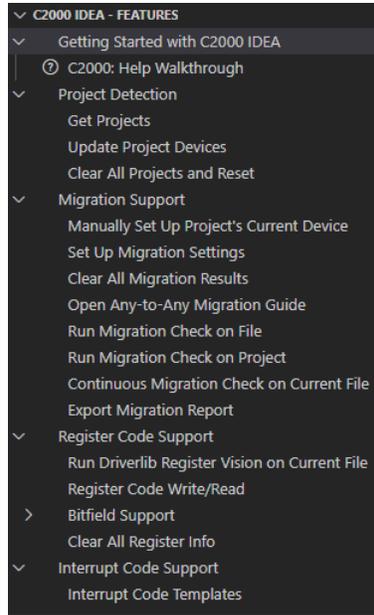


图 3-3. C2000 IDEA - 功能

4. **C2000 IDEA - 项目**：该面板可显示当前在工作区中打开的所有项目。每个项目都有关联的器件型号、当前器件以及迁移器件，以指示要从当前器件迁移到哪个器件。

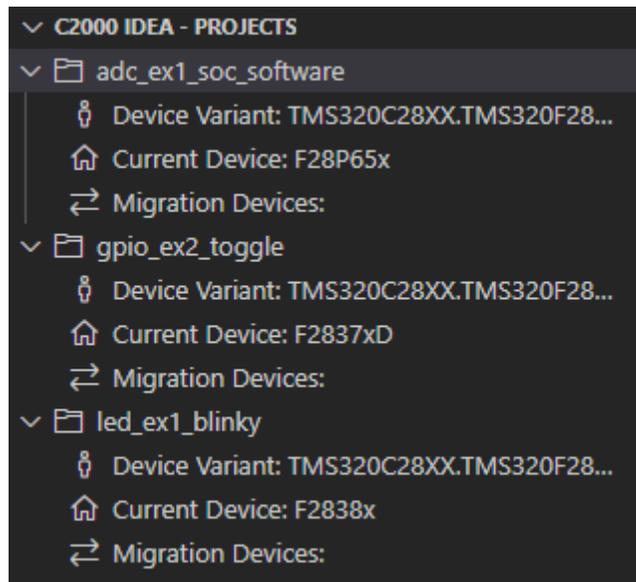


图 3-4. C2000 IDEA - 项目

4 特性

本章详细介绍了上一节讨论的特性的全部功能以及如何在开发软件时优化这些功能。

4.1 定向配套资料交付

通过使用 C2000 IDEA 工具，您可以随时高效访问各种配套资料。通过提供有针对性的配套材料、用户只需点击一下即可获得最相关的信息，而无需搜索各种文档。该工具会自动检测当前主题并提供即时建议，包括器件文档和寄存器定义等关键资源的直接链接。

4.1.1 关键配套资料

当编辑检测到的项目时，指向该项目器件的关键配套资料的链接会自动显示在扩展的“C2000 配套资料”窗格中。此功能会生成指向器件数据表和技术参考手册 (TRM) 的顶层链接，以及指向此文档内关键部分的链接，例如器件功能方框图。

4.1.2 其他资源

当用户光标移动到活动文件中时，配套资料推荐会自动更改，以突显“C2000 配套资料”窗格中关键文档之外的其他实用配套资料。其中包括应用报告、C2000 Academy 页面、实验、E2E™ 论坛答案以及与用户当前正在编辑的字或短语（例如：C2000 外设名称）相关的视频。这些资源的链接位于扩展树的“C2000 其他资源”窗格中。

4.1.3 Register Vision

Register Vision 功能支持轻松访问文件中所有检测到的寄存器的 TRM 寄存器描述。运行该命令后，系统会突出显示特定 C2000 器件的所有检测到的寄存器，并提供指向器件 HTML TRM 中的寄存器描述的直接链接。

按照以下步骤启用 Driverlib Register Vision：

1. 打开 C2000 应用 C-Code 文件。
2. 通过以下任一方式运行 Driverlib Register Vision：
 - a. 按 CTRL+SHIFT+P，输入并选择 *C2000: Run Driverlib Register Vision on Current File*。
 - b. 在扩展树的 **C2000 IDEA - 功能**窗格中，单击 *寄存器代码支持 > 对当前文件运行 Driverlib Register Vision*。
3. 如果没有检测到项目（或者文件不属于项目），扩展会提示输入与文件对应的 C2000 器件。
4. 将鼠标悬停在以黄色突出显示的已识别寄存器上。
5. 点击“在 TRM 中查看寄存器描述”链接，访问在线 HTML TRM 中的寄存器描述。

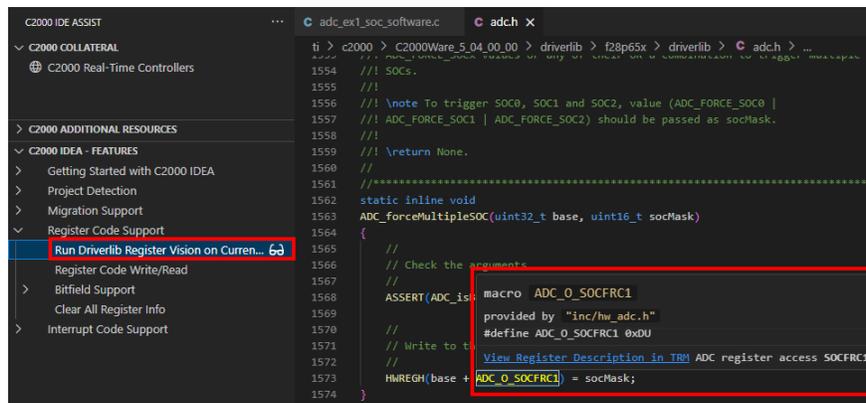


图 4-1. 对 F28P65x 使用 Driverlib Register Vision

C2000 IDEA 还支持对位域式软件开发的寄存器访问。

按照以下步骤启用 Bitfield Register Vision：

1. 打开 C2000 应用 C-Code 文件。
2. 通过以下任一方式运行 Bitfield Register Vision：
 - a. 按 CTRL+SHIFT+P，输入并选择 *C2000: Run Bitfield Register Vision on Current File*。

- b. 在扩展树的 **C2000 IDEA - 功能** 窗格中，点击 **寄存器代码支持 > 位域支持 > 对当前文件运行 Bitfield Register Vision**。
3. 如果没有检测到项目（或者文件不属于项目），扩展会提示输入与文件对应的 C2000 器件。
4. 将鼠标悬停在以黄色突出显示的已识别寄存器上。
5. 点击“在 TRM 中查看寄存器描述”链接，访问在线 HTML TRM 中的寄存器描述。

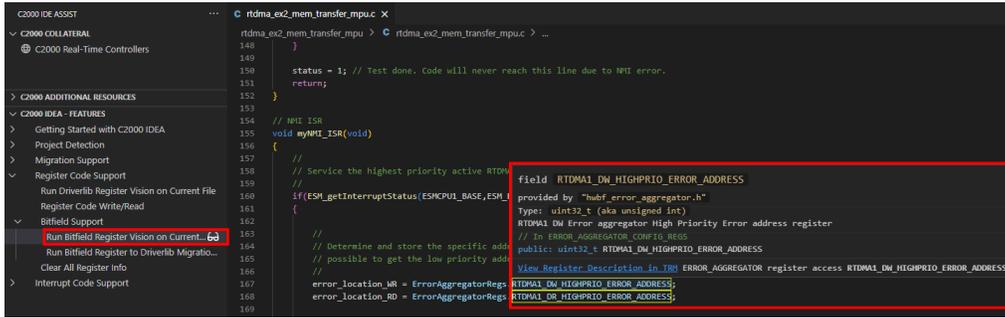


图 4-2. 对 F29H85x 使用 Bitfield Register Vision

4.2 开发人员效率工具

C2000 IDE 通过提供软件模板支持来提高开发人员的效率，使用户可以轻松插入中断定义和寄存器访问等代码。此功能不仅可以加快开发速度，还会指导用户如何编写采用 C2000 的高效软件。

4.2.1 寄存器编码器

寄存器编码功能会自动生成要从特定寄存器或寄存器字段进行写入或读取的代码。自动生成的代码可用作写入或读取寄存器的模板。

要启用寄存器编码器，请按照以下步骤操作：

1. 通过以下任一方式运行寄存器编码器：
 - a. 按 CTRL+SHIFT+P，输入并选择 **C2000: Enable Register Code Write/Read**。
 - b. 单击扩展树的 **C2000 IDEA - Features** 窗格中 **Register Code Support > Register Code Write/Read** 中的复选切换按钮。
2. 启用后，从检测到的项目中打开 C2000 应用 C-Code 文件。
3. 开始输入外设、寄存器或字段名称。寄存器编码器功能按字母顺序显示了可用于为当前器件执行读写操作的所有可用寄存器和寄存器字段。
4. 选择一个选项自动生成模板代码。按需按键盘上的 **Tab** 键编辑基址字段、要读取的变量和/或要写入的值。

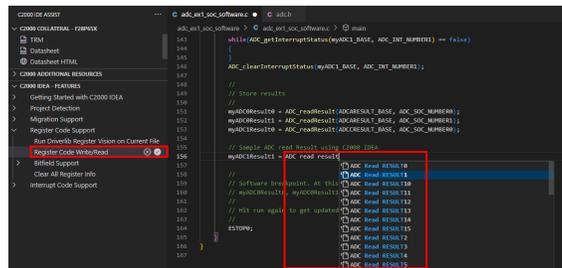


图 4-3. 使用寄存器编码器读取 F28P65x

4.2.2 Interrupt Coder

中断代码模板功能会为器件上可用的任何中断自动生成模板中断服务例程 (ISR) 函数。

要启用 Interrupt Coder，请按照以下步骤操作：

1. 通过以下任一方式运行 Interrupt Coder：
 - a. 按 CTRL+SHIFT+P，输入并选择 **C2000: Enable Register Code Write/Read**。
 - b. 点击“**C2000 IDEA - 功能**”窗格中的“**中断代码支持**” > “**中断代码模板**”。
2. 启用后，打开 C2000 应用 C-Code 文件以添加中断服务例程 (ISR)。

- 首先，在代码编辑器中输入短语 “*interrupt handler [interrupt name]*”。Interrupt Coder 功能会按字母顺序显示当前器件所有可用中断的列表。
- 选择一个选项来自动生成模板中断代码。请填写具体应用所需的其余代码。

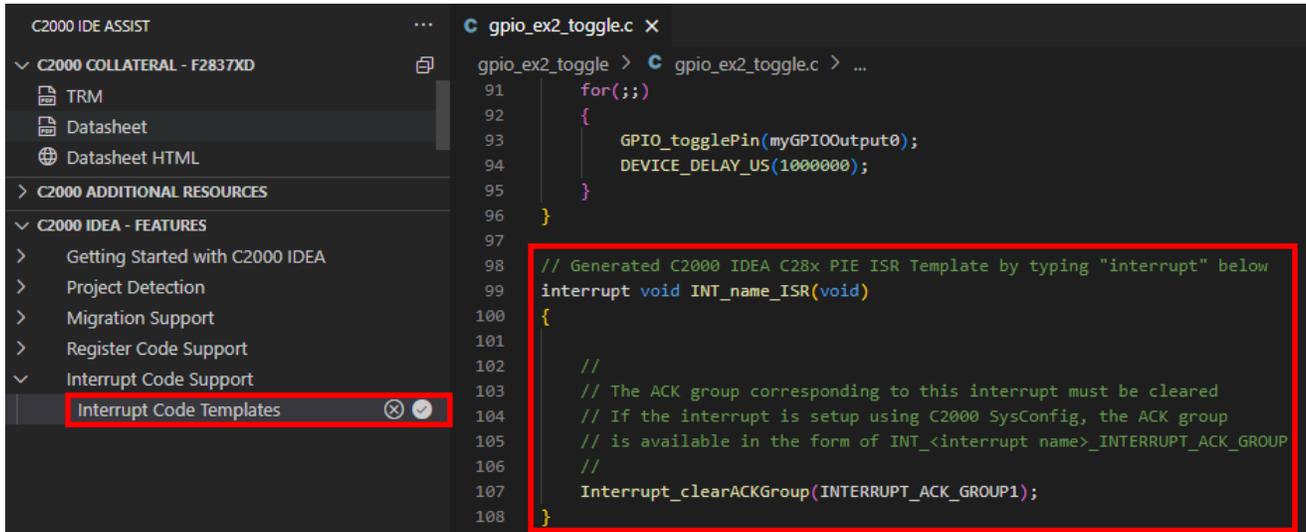


图 4-4. 面向 C28x 的 Interrupt Coder

备注

对于 C28x 器件，系统会自动填充给定外设中断的对应 ACK 组。确保在此行之前添加应用特定代码。

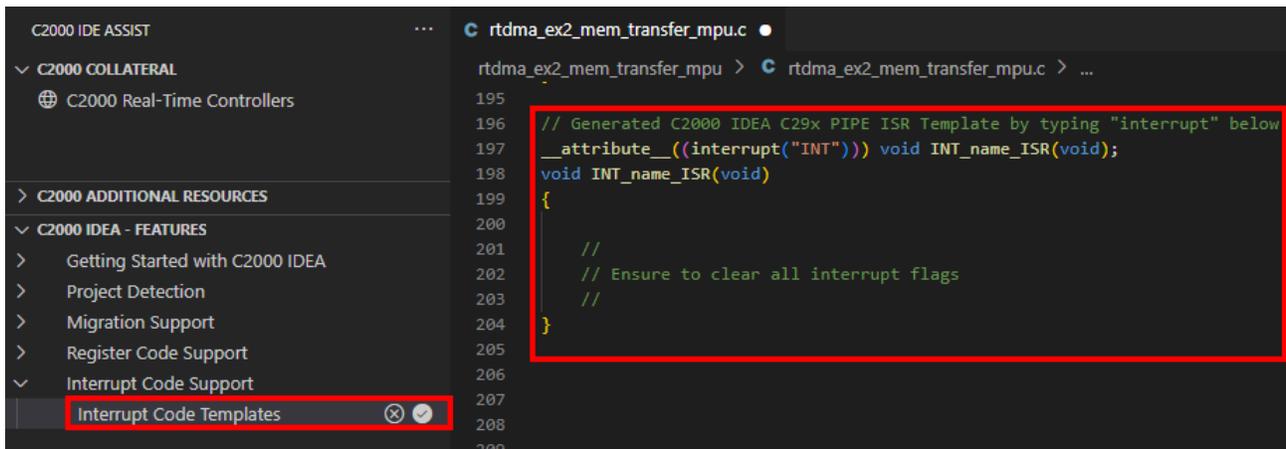


图 4-5. 面向 C29x 的 Interrupt Coder

备注

对于 C29x 器件，系统会自动在其中加入中断类型属性来指示正在定义的中断类型。默认情况下，扩展会将中断定义为 *INT*，但您可以将此属性修改为 *RTINT*。

4.3 迁移支持

C2000 IDEA 通过自动检测 *driverlib* 代码中的更改（例如添加、删除或修改的寄存器和寄存器字段）简化了 C2000 器件之间的代码迁移，从而实现无缝的迁移过程。此支持范围已得到扩展，涵盖 *driverlib* 迁移过程中检测到的项目和独立文件。对于选定的迁移路径、还可以进行位域代码迁移。此外、该工具还可协助完成 F28x 到 F28x 和 F28x 到 F29x 器件之间的迁移、突出二者之间的差异并提供建议的解决方案。

备注

有关每个命令支持的特定器件的完整列表、请参阅 [节 4.4](#)；有关 driverlib 样式、位域样式与代码之间的差异、请参阅 [软件驱动程序文档](#)。

4.3.1 Driverlib 迁移

C2000 IDE 扩展可用于在使用 driverlib 样式代码编写的项目或文件上运行 F28x 到 F28x 或 F28x 到 F29x 迁移检查。这种代码样式的特点是使用对 driverlib 源文件中定义的函数的调用 (如 : Device_init()) 和/或包含 _O_ 语法的寄存器访问。

4.3.1.1 基于文件的迁移检查

按照以下步骤对文件启用 Driverlib 迁移：

1. 打开 C2000 应用 C-Code 文件。
2. 通过以下任一方式运行迁移检查：
 - a. 按 CTRL+SHIFT+P，输入并选择 *C2000: Run Migration Check on File*。
 - b. 在扩展树的“**C2000 IDEA - 功能**”窗格中，单击“*迁移支持*” > “*对项目运行迁移检查*”
3. 选择将文件中的代码应用于哪一个现有的 C2000 器件。
4. 选择要将文件迁移至哪一个 C2000 器件。
5. 完成后，屏幕右下角的状态栏会显示“*已完成迁移检查*”。文件中所有主要的迁移问题均已用红色波浪下划线标出。文件中的所有其他迁移警告均已用黄色波浪下划线标出。
6. 查看并解决整个问题。当鼠标悬停在带下划线的代码上时，系统会提供以下选项：
 - a. 选择 *查看问题*，快速循环浏览文件中检测到的问题。

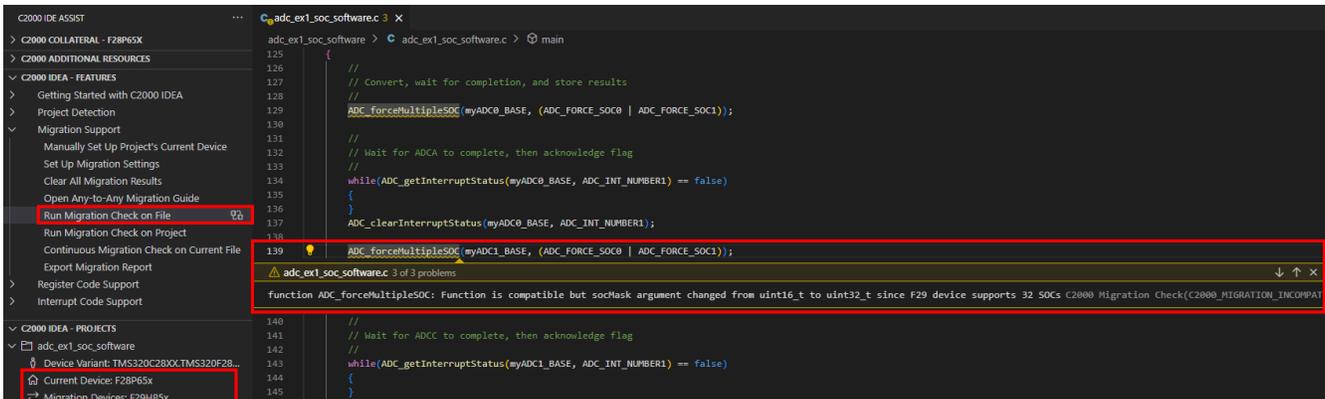


图 4-6. 迁移检查 - 查看问题

- b. 选择 *快速修复*，缓解迁移问题。选择以下选项之一：
 - i. *查看从 [当前器件] 到 [迁移器件] 的迁移配套资料* - 此选项会打开使用最新版本的 C2000WARE 的特定迁移路径的在线迁移资料的链接。
 - ii. *为 [当前器件] 和 [迁移器件] 打包器件专属 #IFDEF* - 此选项会在代码行周围自动生成预处理器包装器，以便为新器件编译代码的更新版本。用修改后的代码填写带有“*//输入替换代码*”注释的行，并在文件的对应位置为当前器件添加一个 #define。
 - iii. *忽略与代码相关的错误* - 此选项会忽略此迁移问题。

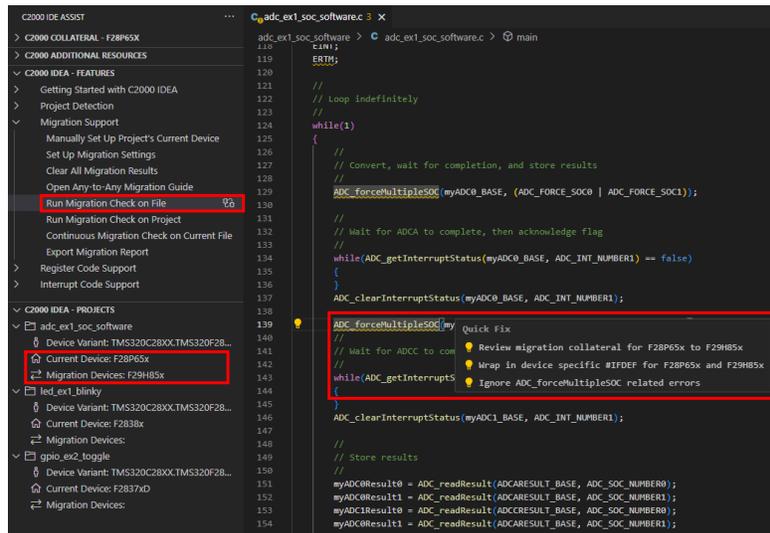


图 4-7. 迁移检查 - 快速修复

7. 迁移报告包含已检测到的迁移问题的列表。如要将迁移报告导出到目标文件路径，请执行以下操作之一：
- 输入 CTRL+SHIFT+P，输入并选择 *C2000: Export Migration Report*。
 - 单击扩展树的 **C2000 IDEA - 功能** 窗格中的“迁移支持” > “导出迁移报告”。

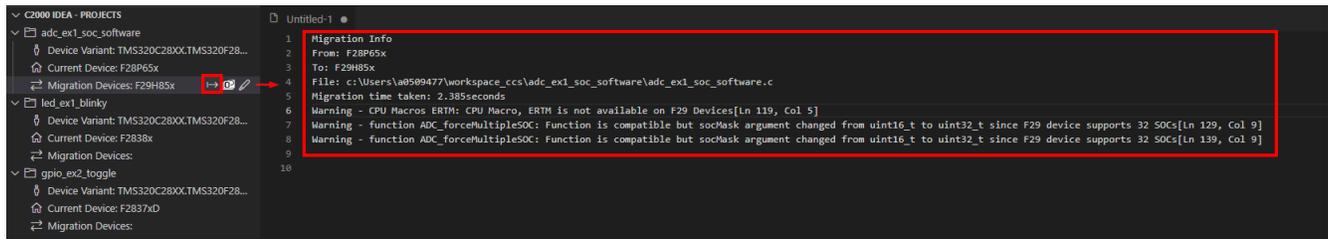


图 4-8. 迁移报告

4.3.1.2 基于项目的迁移检查

按如下步骤在项目上启用 Driverlib 迁移：

- 运行项目检测（请参阅 节 2.3 中的说明）。
- 设置要迁移的来源和目标 C2000 器，并自定义要忽略迁移检查的文件夹/文件，方法如下：
 - 单击扩展树的 **C2000 IDEA - 项目** 窗格中检测到的项目下的迁移器件旁边的以下图标。
 - 按 CTRL+SHIFT+P，输入并选择 *C2000: Set Up Migration Settings*。
 - 在扩展树的 **C2000 IDEA - 功能** 窗格中单击“迁移支持” > “设置迁移设置”。

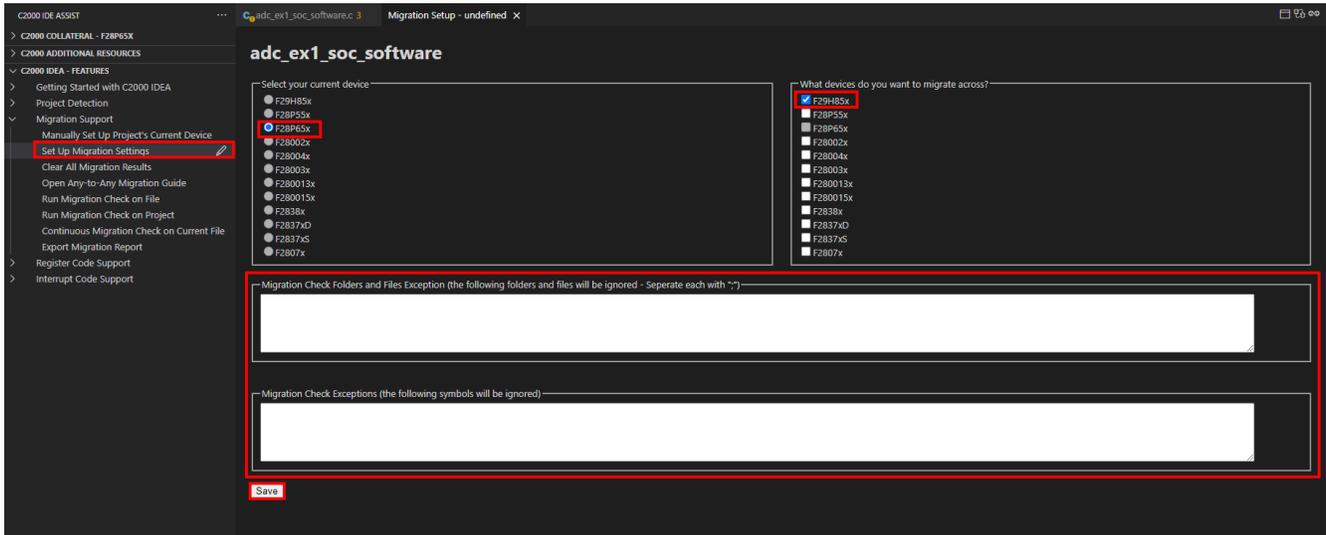


图 4-9. 迁移设置设置

3. 通过以下任一方式运行迁移检查：

- 单击扩展树的 **C2000 IDEA - 项目** 窗格中检测到的项目下的迁移器件旁边的以下图标。
- 按 CTRL+SHIFT+P，输入并选择 **C2000: 对项目运行迁移检查**。
- 在扩展树的 **C2000 IDEA - 功能** 窗格中，单击迁移支持 > 对项目运行迁移检查。

备注

如果迁移文件夹中的文件夹或文件与迁移设置页面中要忽略的文件信息存在重叠，则扩展会抛出错误。

备注

当工具已经在对项目运行迁移检查时，请勿使用这些迁移功能。等待屏幕右下角的 [项目名称] 迁移检查完成，然后再启用其他迁移功能。运行检查所需的时间完全取决于项目中存在多少个文件、多少行和多少代码更改。迁移报告包含每个文件所用的时间。

- 在工作区中选择要运行迁移检查的项目。
- 完成后，屏幕右下角的状态栏会显示“已完成 [项目名称] 的迁移检查”。文件中所有主要的迁移问题均已用红色波浪下划线标出。文件中的所有其他迁移警告均已用黄色波浪下划线标出。
- 查看并解决整个文件中的问题。当鼠标悬停在带下划线的代码上时，系统会提供以下选项：
 - 选择查看问题，快速循环浏览文件中检测到的问题。
 - 选择快速修复，缓解迁移问题。选择以下选项之一：
 - 查看从 [当前器件] 到 [迁移器件] 的迁移资料 - 此选项会打开使用最新版本的 C2000WARE 的特定迁移路径的在线迁移资料的链接。
 - 为 [当前器件] 和 [迁移器件] 打包器件专属 #IFDEF - 此选项会在代码行周围自动生成预处理器包装器，以便可以为新器件编译代码的更新版本。用修改后的代码填写带有“//输入替换代码”注释的行，并在文件的某处为当前器件添加一个 #define。
 - 忽略与代码相关的错误 - 此选项会忽略此迁移问题。
- 迁移报告包含检测到的迁移问题的列表、完成迁移所需的时间以及用户自定义忽略的任何设置。如要将迁移报告导出到目标文件路径，请执行以下操作之一：
 - 单击扩展树的 **C2000 IDEA - 项目** 窗格中检测到的项目下的迁移器件旁边的以下图标。
 - 输入 CTRL+SHIFT+P，输入并选择 **C2000: Export Migration Report**。
 - 单击扩展树的 **C2000 IDEA - 功能** 窗格中的“迁移支持” > “导出迁移报告”。

4.3.2 位域迁移

C2000 IDEA 扩展可用于在使用 bitfield 样式代码编写的文件上运行 F28x 到 F28x 或 F28x 到 F29x 迁移检查。这种代码样式的特点是使用对 bitfield 源文件中定义的函数的调用 (如 : InitsysCtrl()) 和/或包含 [base

name].[register name].all 或 [base name].[register name].bit.[field name] 语法的寄存器访问。

如要启用位域迁移，请按照以下步骤操作：

1. 打开 C2000 应用 C-Code 文件。
2. 通过按 CTRL+SHIFT+P，输入并选择 “C2000: Run Bitfield Migration Check on File” 来运行迁移检查
3. 选择将文件中的代码应用于哪一个现有的 C2000 器件。
4. 选择要将文件迁移至哪一个 C2000 器件。
5. 完成后，屏幕右下角的状态栏会显示 “已完成从 [当前器件] 到 [迁移器件] 的位场迁移”。文件中所有迁移问题均已用红色波浪下划线标出。
6. 查看并解决整个问题。当鼠标悬停在带下划线的代码上时，系统会提供以下选项：
 - a. 选择 *查看问题*，快速浏览文件中检测到的问题。
 - b. 选择 *快速修复*，缓解迁移问题。选择以下选项之一：
 - i. *查看从 [当前器件] 到 [迁移器件] 的迁移配套资料* - 此选项会打开使用最新版本的 C2000WARE 的特定迁移路径的在线迁移资料的链接。
 - ii. *为 [当前器件] 和 [迁移器件] 打包器件专属 #IFDEF* - 此选项会在代码行周围自动生成预处理器包装器，以便可以为新器件编译代码的更新版本。用修改后的代码填写带有 “//输入替换代码” 注释的行，并在文件的对应位置为当前器件添加一个 #define。
 - iii. *忽略与代码相关的错误* - 此选项会忽略此迁移问题。

4.4 命令列表

请参阅下表、了解可在 C2000 IDE Assist 工具中使用的命令的完整列表。

表 4-1. 表需要标题

命令名称	支持的器件	支持的软件	项目检测或默认器件设置
C2000：实用演示	不适用	不适用	
C2000：Interrupt Coder 实用演示	不适用	不适用	
C2000：Register Coder 实用演示	不适用	不适用	
C2000：Register Vision 实用演示	不适用	不适用	
C2000：打开 C2000-IDEA 扩展视图	不适用	不适用	
C2000：手动设置项目的当前器件	GEN3、GEN4	不适用	必需
C2000：设置迁移设置	GEN3、GEN4	不适用	必需
C2000：获取项目	GEN3、GEN4	Bitfield + Driverlib	
C2000：清除所有项目并重置	GEN3、GEN4	Bitfield + Driverlib	
C2000：更新项目器件	GEN3、GEN4	Bitfield + Driverlib	
C2000：打开任意版本间的迁移指南	从 GEN3、GEN4 更改为 GEN3、GEN4	不适用	
C2000：导出迁移报告	从 GEN3、GEN4 更改为 GEN3、GEN4	Driverlib	
C2000：对文件运行迁移检查	从 GEN3、GEN4 更改为 GEN3、GEN4	Driverlib	
C2000：对当前文件启用持续迁移检查	从 GEN3、GEN4 更改为 GEN3、GEN4	Driverlib	
C2000：对当前文件禁用持续迁移检查	从 GEN3、GEN4 更改为 GEN3、GEN4	Driverlib	

表 4-1. 表需要标题 (续)

命令名称	支持的器件	支持的软件	项目检测或默认器件设置
C2000 : 清除所有迁移结果	从 GEN3、GEN4 更改为 GEN3、GEN4 & 从 F2803x 更改为 F280013x	Bitfield + Driverlib	
C2000 : 对项目运行迁移检查	从 GEN3、GEN4 更改为 GEN3、GEN4	Driverlib	
C2000 : 对文件运行 Bitfield 迁移检查	从 GEN3、GEN4 更改为 GEN3、GEN4 & 从 F2803x 更改为 F280013x	位域	
C2000 : 对当前文件运行 Driverlib Register Vision	GEN3、GEN4	Driverlib	推荐
C2000 : 对当前文件运行 Bitfield Register Vision	GEN3、GEN4、F2803x	位域	推荐
C2000 : 对当前文件运行从位域寄存器到 Driverlib 的迁移 (测试版)	GEN3、GEN4、F2803x	位域	推荐
C2000 : 清除所有寄存器信息	GEN3、GEN4、F2803x	Driverlib + Bitfield	
C2000 : 启用寄存器代码写入/读取	GEN3、GEN4	Driverlib	必需
C2000 : 禁用寄存器代码写入/读取	GEN3、GEN4	Driverlib	必需
C2000 : 启用中断代码模板	GEN3、GEN4	Driverlib	必需
C2000 : 禁用中断代码模板	GEN3、GEN4	Driverlib	必需
C2000 : 项目树状图刷新	GEN3、GEN4	Bitfield + Driverlib	

5 总结

德州仪器 (TI) C2000 SDK 集成了驱动程序和库等基本工具，可帮助各个级别的开发人员构建实时控制应用程序。其中包括可简化代码初始化和外设配置的 C2000 SysConfig。C2000 IDEA 工具是在这些基础功能的基础上构建而成，用于集中开发环境，以便更轻松地进行编码和调试。IDE Assist 有助于实现跨器件组合和从旧代码到新代码的迁移，帮助用户浏览独特的 C2000 功能，并通过在开发的每个阶段提供实时协助来加快软件实施。

6 参考资料

工具:

- 德州仪器 (TI), [C2000 IDEA Open VSX \(VSIX 下载 \)](#)
- 德州仪器 (TI), [C2000 Idea GitHub 存储库 \(VSIX 下载 \)](#)
- 德州仪器 (TI), [Code Composer 20 IDE](#)
- 德州仪器 (TI), [C2000WARE \(F28x SDK\)](#)
- 德州仪器 (TI), [F29H85X-SDK \(F29x SDK\)](#)

文档 :

- 德州仪器 (TI), [C28x Academy - 迁移资源](#)
- 德州仪器 (TI), [C29x Academy - 迁移资源](#)
- 德州仪器 (TI), [F28x 到 F29x 软件迁移指南](#)
- 德州仪器 (TI), [将应用软件迁移到 C29 CPU 应用说明。](#)
- 德州仪器 (TI), [将 TMS320F2837x、TMS320F2838x、TMS320F28P65x 迁移至 TMS320F29H85x 用户指南](#)
- 德州仪器 (TI), [C2000 设计与开发](#)

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、某特定用途方面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功能安全、信息安全、监管或其他要求。

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。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/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。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
版权所有 © 2025，德州仪器 (TI) 公司