

如何将 CCS 3.x 工程迁移至最新的 Code Composer Studio™ (CCS)



DSP Processors

摘要

本应用报告介绍了将可能在多年前开发的 Code Composer Studio™ 旧工程迁移到最新 CCS 和工具的要点。本文主要以 DSP 处理器为例进行阐述，但报告中的方法也可能适用于其他处理器。

内容

1 简介.....	2
2 CCS 迁移.....	2
2.1 弃用通知.....	2
2.2 CCS 内的 CCS 培训.....	3
2.3 CCS 旧工程导入向导.....	5
3 DSP/BIOS 与 SYS/BIOS.....	5
3.1 将旧工程迁移到最新工具.....	5
4 工具链：CGT、编译器、链接器.....	5
5 RTSC 和 XDC.....	5
6 COFF 与 ELF.....	6
7 Processor SDK.....	7
8 NDK 迁移.....	9
9 参考文献.....	10
10 修订历史记录.....	10

插图清单

图 2-1. TI Resource Explorer 中的 CCS 培训.....	3
图 2-2. 关于调试主题的 CCS 培训.....	4
图 5-1. XDC 用户.....	6
图 7-1. 最大化重复使用软件.....	7
图 7-2. 典型开发流程.....	8

商标

Code Composer Studio™ are trademarks of Texas Instruments.
所有商标均为其各自所有者的财产。

1 简介

这里并没有将 CCS 3.x 工程迁移到最新版本的常规步骤。所需步骤将取决于所涉及的器件以及所涉及的软件包。总体而言，建议注意以下几点：

- CCS 迁移
- DSP/BIOS 到 SYS/BIOS
- 工具链：编译器、链接器
- RTSC 和 XDC
- COFF 与 ELF
- Processor SDK
- NDK 迁移

2 CCS 迁移

[Code Composer Studio 下载](#)

授权：CCS V7 和更高版本均为技术软件公开可用 (TSPA)。CCS V4、V5、V6、V7、V8、V9 全都是免费的。

2.1 弃用通知

- Code Composer Studio 的所有 32 位版本都已被弃用。
 - Windows 从 [CCSv9.0.0](#) 开始
 - Linux 从 [CCSv6.2.0](#) 开始
 - macOS 从未有 32 位版本
- 64 位 CCS 版本不支持 Spectrum Digital XDS510USB JTAG 调试器和任何其他 32 位仿真器。
- [CCSv8](#) 的发行版不再支持 Ubuntu 14.04。
- [CCSv8](#) 的发行版不再支持 Ubuntu 12.04LTS。
- [CCSv7](#) 的发行版不再支持 Windows XP。

Eclipse 是计算机编程中使用的集成开发环境 (IDE)，包含一个基本工作区和一个用于定制环境的可扩展插件系统。Eclipse 主要是用 Java 编写的。

[CCS v3 迁移到 CCS v5](#)

2.2 CCS 内的 CCS 培训

现在，Code Composer Studio 的培训材料已集成到 CCS 内的 Resource Explorer 中或 dev.ti.com 中。在“Development Tools” → “Integrated Development Environments” → “Code Composer Studio”部分下，您可以浏览适用于 CCS 的所有培训材料，包括讲座、培训模块和视频。

依次点击“View”和“Resource Explorer”。图 2-1 显示了 CCS v9.3 的屏幕截图。

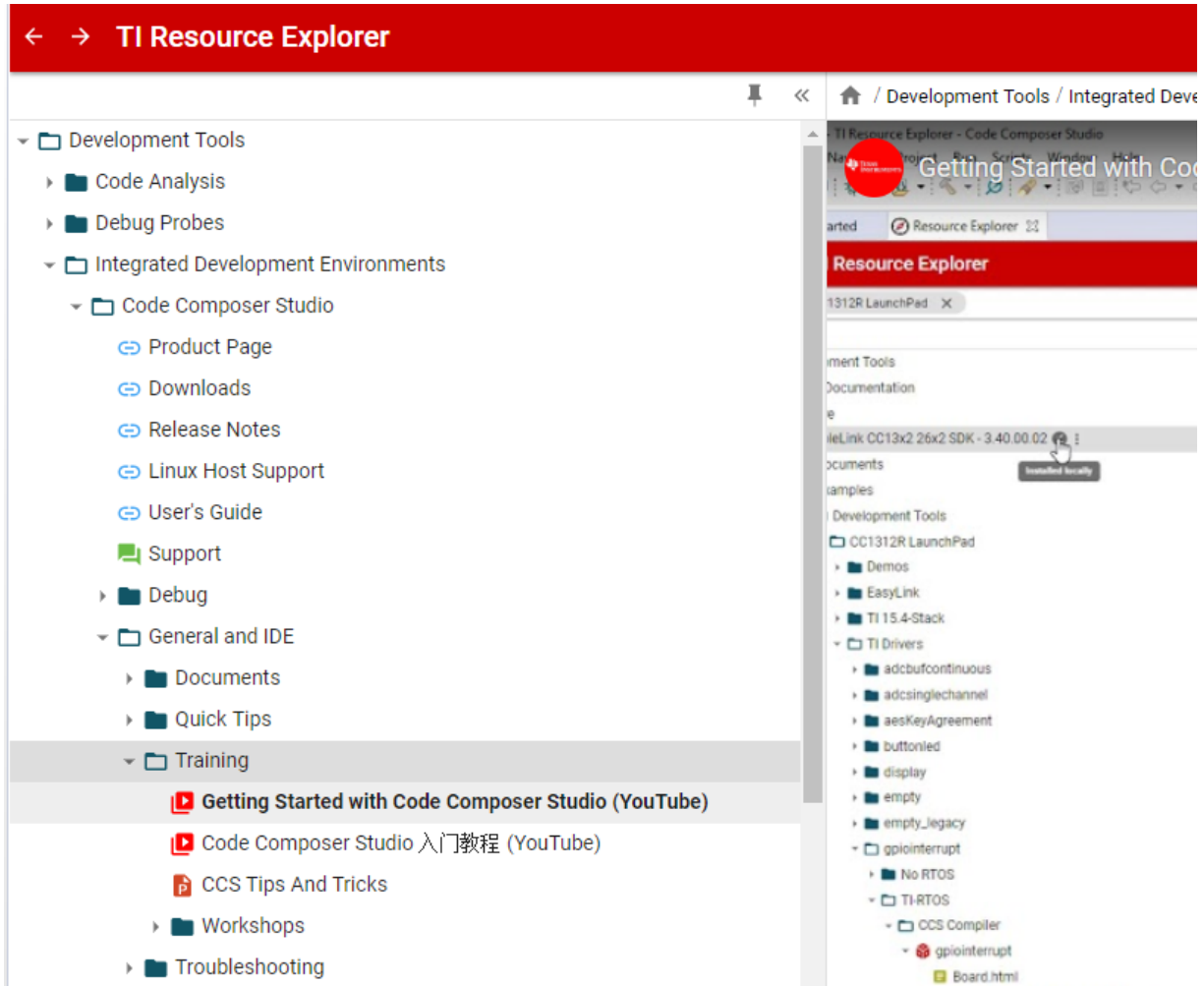


图 2-1. TI Resource Explorer 中的 CCS 培训

正在调试中：

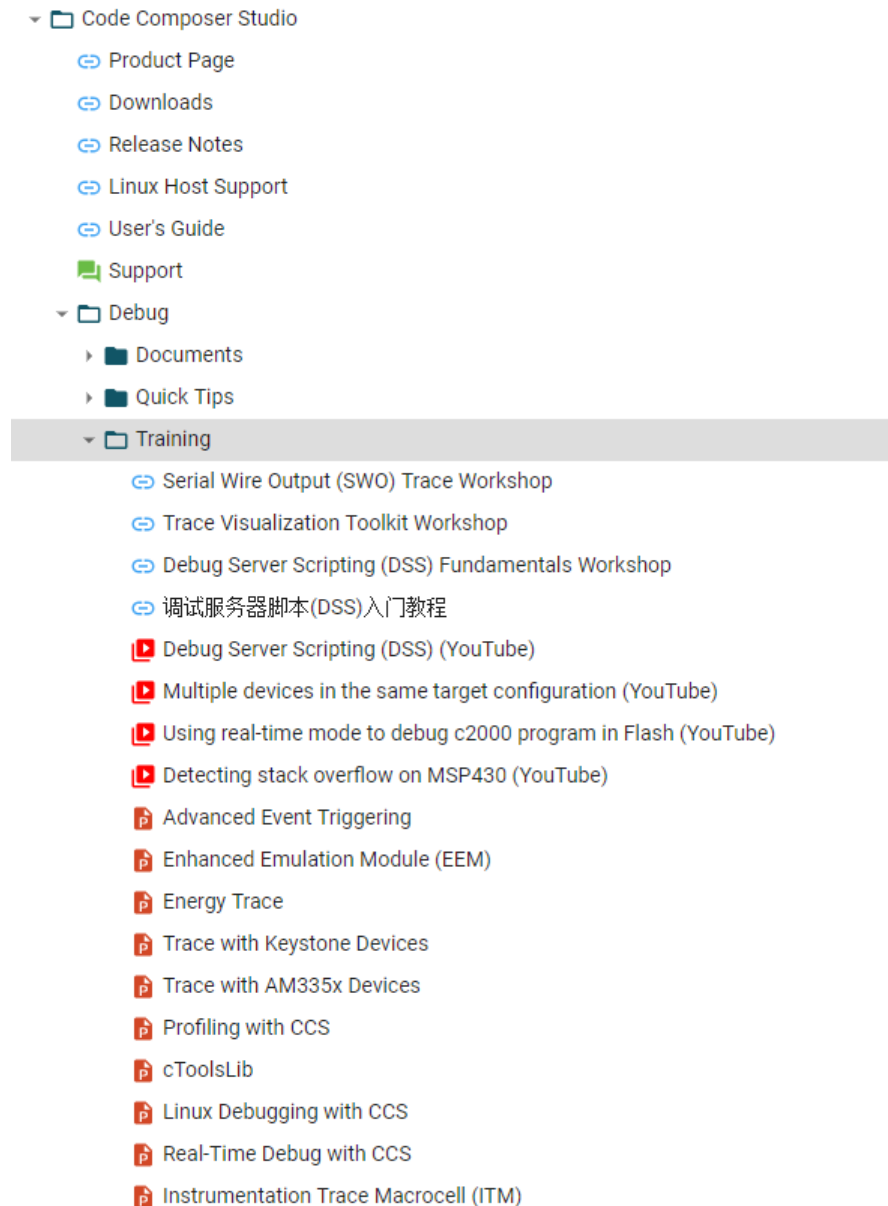


图 2-2. 关于调试主题的 CCS 培训

CCS v4 中引入了基于 Eclipse 的框架。

2.3 CCS 旧工程导入向导

有关 CCS 旧工程导入向导的更多详细信息，请参阅 https://software-dl.ti.com/ccs/esd/documents/ccs_legacy-project-import.html。

3 DSP/BIOS 与 SYS/BIOS

- [TI-RTOS \(SYS/BIOS\) 发行版](#)
- [《SYS/BIOS \(TI-RTOS 内核\) 用户指南》](#)

有关 TI-RTOS 下载信息，请参阅 https://software-dl.ti.com/dsp/dsp_registered_sw/sdo_sb/targetcontent/bios/index.html。

3.1 将旧工程迁移到最新工具

这里有一份介绍如何从 DSP/BIOS 迁移到 SYS/BIOS 的迁移指南。从“变化内容”角度来看，您会发现该指南很有用。如需了解更多信息，请参阅 [《将 DSP/BIOS 5 应用迁移到 SYS/BIOS 6》](#)。

[TI-RTOS 培训系列入门](#)

4 工具链：CGT、编译器、链接器

适用于德州仪器 (TI) 处理器的代码生成工具：[下载](#)

CCS 已附带 TI 编译器，但每个都有固定版本。

C6000 CGT v8.3 是一个新的编译器：

- v8.3 仅在 ELF EABI 模式下支持 C6400+、C6740 和 C6600
- v8.3 支持 C++14 标准 ISO/IEC 14882:2014，而不再支持 C++03
- 旧版编译器生成的 C++ obj 代码与 v8.0+ RTS obj 库不兼容
- v8.3 提供与 v7.4 相当的性能。性能可能因应用而异
- v7.4.x 将继续支持 (长期) ELF EABI 或 COFF ABI 模式下的所有处理器版本

符合以下条件的客户应使用 CGT v8.3：

- 使用 OpenCL、OpenMP 或 HPC-MCSDK 开发新应用
- 开发的新应用会利用仅在 v8.0 及更高版本中可用的新编译器功能 (例如，本机向量类型)

符合以下条件的客户应使用 CGT v7.4.x：

- 维护一个您不希望或不需要在近期内过渡到 v8.3 的现有代码库
- 开发新的应用或维护使用 COFF ABI 的现有应用
- 在 C6200、C6400、C6700、C6700+ 或 Tesla 上开发新的应用或维护现有应用

您可以将旧版本的编译工具与更高版本的 CCS 一起使用。

旧版本的编译器不会随附 CCS；必须另行安装。可以直接单独安装，也可以直接从 CCS IDE 安装旧版本：[安装新软件](#)。

5 RTSC 和 XDC

实时软件组件 (RTSC) 工程提供了基础工具和低级运行内容，以便使用面向所有嵌入式平台的 C 语言进行基于组件的开发。使用 RTSC，我们可以享受 RTSC 更高级别的编程和更高级别的性能。

RTSC 工程的起点将包括德州仪器 (TI) 当前免费提供的 XDC (eXpress DSP Components) 工具。XDC 工具产品包含创建、测试、部署、安装和使用 RTSC 组件所需的所有工具。

XDC 的主要好处是它可以标准化目标内容的交付，并使目标内容更容易包含在应用中。

XDC 用户分为开发者，我们称之为“消费者”和“生产者”。消费者集成内容包：将 DSP 算法、器件驱动程序、TCP/IP 堆栈、实时操作系统等集成到应用程序中。生产者创建消费者使用的包。

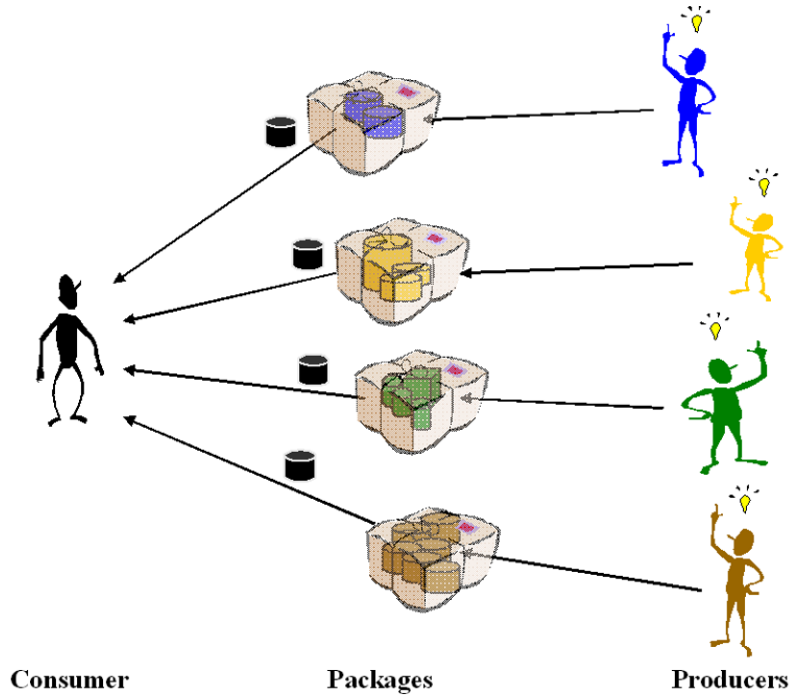


图 5-1. XDC 用户

XDC “包” 是一个指定的文件集合，其中的文件形成一个进行版本控制、更新和从生产者到消费者传递的单元。每个包都体现为文件系统中的特别命名的目录（及其内容）。包是在整个生命周期中管理内容的焦点。所有包都是作为一个单元进行编译、测试发布和部署。

此处是链接：<https://www.eclipse.org/rtsc/>，点击“User’s Guide”。

概述：http://rtsc.eclipseprojects.io/docs-tip/Overview_of_RTSC。

XDC 发行版：http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/rtsc/。

通常，所有 RTSC 工程也是 CCS 工程。唯一的区别是它们使用 SYMBIOS、TI-RTOS 或依赖于 RTSC 组件的 SDK。

6 COFF 与 ELF

术语 ABI 代表应用程序二进制接口。ABI 指定编译器和链接器应如何处理各种事项，例如寄存器分配、调用约定、类型大小和目标文件格式。ABI 指定的约定可以将单独编译的目标文件和库链接在一起，形成一个内聚的可执行文件。2010 年推出了名为 EABI（嵌入式应用程序二进制接口）的 ABI。

COFF ABI 和 EABI 之间的主要（但不是唯一的）区别是目标文件格式。COFF ABI 使用 COFF（通用对象文件格式），而 EABI 使用 ELF（可执行连接格式）。

支持 EABI 的 C6000 编译器的第一个版本是 7.2.0 版。支持 COFF ABI 的最后一个版本是 7.4.24 版。

可执行连接格式 (ELF) 比通用对象文件格式 (COFF) 具有更多的调试功能。

关于如何从 COFF ABI 更改为 EABI :

- 首先, 请确保您的所有库以及工程所依赖的任何其他软件都具有 EABI 变体。如果没有, 则不能使用这些库, 或只能继续停留于 COFF ABI。

TI 编译器开关: `--abi=eabi` 或 `-abi=coffabi`。

无法将 COFF 和 ELF 目标文件链接在一起。

有关将您自己的代码从 COFF ABI 迁移到 EABI 的信息, 请参阅 [C6000 EABI 迁移 Wiki](#)。

7 Processor SDK

如需查看发行版信息, 请参阅 E2E 主题: [Sitara 和 DSP 软件公告](#)。

NOTE

要即时获取最新的错误修复和特性, 建议点击 SDK 下载页面上的“Alert Me”, 以便在发布新的 SDK 时获得电子邮件通知。

推荐的做法是, 在下载 SDK 的特定产品页面上, 点击“Alert Me”以便在新版本发布时获得自动电子邮件通知。

有用的 SDK 培训材料位于: [Processor SDK 培训系列](#)。

特别是以下内容对于 RTOS 很有用: [1.5 使用 Processor SDK RTOS 进行应用开发的说明](#)

图 7-1 展示了如何保护嵌入式开发人员在现有 TI 器件上的软件投资, 因为未来 TI SoC 平台的所有软件版本都将使用这种软件方法。从应用角度来看, 即使基础 SoC 不同, 功能 CSL 和 LLD 的 API 接口也会保持不变, 因此, 即使基础软件可能不同, 也可以重复使用应用软件。凭借这种软件方法, 曾经开发过 TI 处理器软件的应用程序开发人员无需重新学习使用 TI 软件的知识。

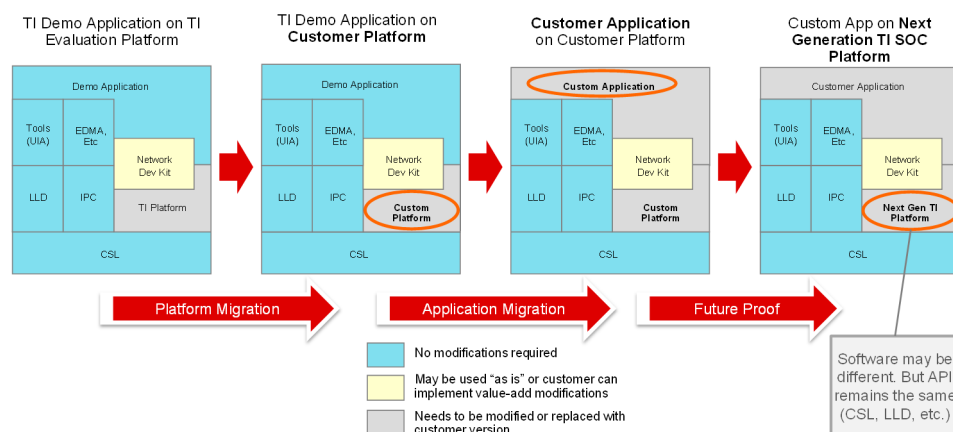


图 7-1. 最大化重复使用软件

图 7-2 显示了软件开发人员在使用 Processor SDK RTOS 时将经历的典型应用开发流程。当您浏览演示文稿时, 此开发流程将针对每个阶段的培训流程提供更多细节。

通常, 您首先购买一个已验证处理器功能的评估平台, 然后下载主机上所需的软件环境以便开始相关开发。

拥有评估模块后, 您需要执行一些常见的硬件设置步骤, 以便在评估平台上运行软件。例如, 通过连接仿真器来设置开发环境, 这可能会涉及连接电缆, 以便通过通用异步接收器/发送器 (UART)、USB 接口进行连接。通常, EVM 套件随附有快速入门指南说明, 其中提供了设置 EVM 并在平台上即开即用演示的步骤。

设置好硬件来运行软件后, 请完成设置主机开发环境的过程, 然后运行一些简单的示例代码以验证 EVM 是否正常运行。例如, 在内核上运行 hello world 示例, 学习运行 RTOS 应用并检查 EVM 上的一些基本功能, 例如使用通用输入/输出 (GPIO) 来闪烁 LED 以及检查 EVM 上的 UART、USB 和网络连接。

在 EVM 上检查了基本功能之后, 建议运行 Processor SDK 中提供的例程。这些例程集成了 SDK 的多个组件, 并通过创建实际系统用例来突出器件功能。

SDK 中包含使用驱动程序进行应用开发的更多详细信息。这里讨论了 SDK 中的一些关键元素，这些元素将帮助您创建自己的应用。

还讨论了从 TI 评估平台迁移到定制应用板时的应用程序可移植性方面，并介绍了 Processor SDK 的组件，使软件更易于移植。

最后，了解在迁移到定制平台后如何定制软件，并提供一些有助于系统集成的指导。

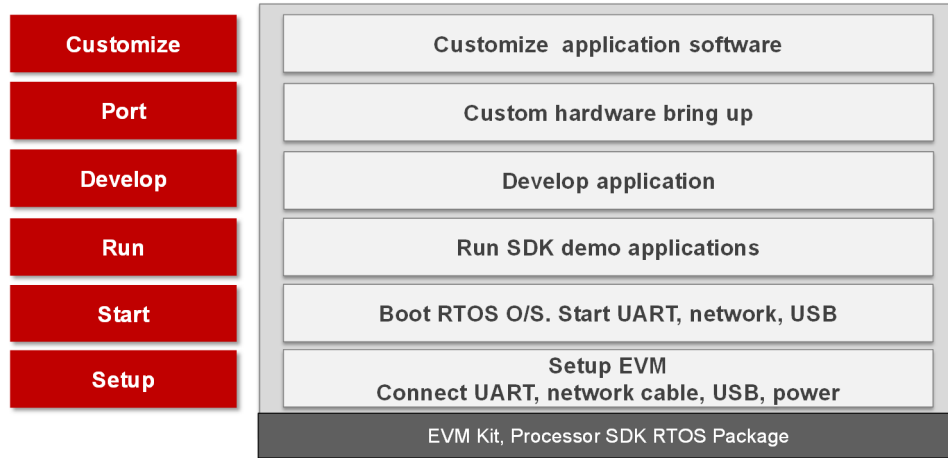


图 7-2. 典型开发流程

8 NDK 迁移

可在已安装的 Processor SDK 包中找到 NDK 迁移指南。例如：C:/ti/ndk_3_61_01_01/docs/ndk/NDK_2_to_3_Migration_Guide.html。

在 BIOS 配置上需要保留一些重要的配置。例如，在 C6657 平台上：

```
/* 加载 CSL 包 */
var devType           = "c6657";
var Csl               = xdc.useModule('ti.csl.Settings');
Csl.deviceType       = devType;
Csl.useCSLIntcLib    = true;
/* 加载 OSAL 包 */
var osType = "tirtos"
var Osal = xdc.useModule('ti.osal.Settings');
Osal.osType = osType;
/* 加载 QMSS 包 */
var Qmss           = xdc.loadPackage('ti.drv.qmss');
/* 加载 EMAC 包 */
var Emac = xdc.loadPackage('ti.drv.emac');
Emac.Settings.socType = devType;
var socType = "c6657";
var Nimu = xdc.loadPackage('ti.transport.ndk.nimu');
Nimu.Settings.socType = socType;
/*
** 使用此加载通过 RTSC 来配置 NDK 2.2 及更高版本。在以前版本的
** NDK 中，不支持 RTSC 配置，应将此内容注释掉。
*/
var Ndk = xdc.loadPackage('ti.ndk.config');
var Global = xdc.useModule('ti.ndk.config.Global');
/*
** 这允许创建检测信号（轮询功能），但不生成堆栈线程
**
** 在 cdoc（帮助文件）中查看可以配置哪些 CfgAddEntry 项。** 我们规定在主要任务线程 hpdspuaStart 中自行配置时，不要
** 创建任何堆栈线程（服务）。
*/
Global.enableCodeGeneration = false;
```

9 参考文献

- [培训](#)
- [处理器 E2E 论坛](#)
- [适用于 OMAPL138 处理器且支持 Linux 和 TI-RTOS 的 Processor SDK](#)
- [适用于 66AK2Ex 处理器的 Processor SDK - 支持 Linux 和 TI-RTOS](#)
- [适用于 66AK2Gx 处理器的 Processor SDK - 支持 Linux 和 TI-RTOS](#)
- [适用于 66AK2HX 处理器的 Processor SDK - 支持 Linux 和 TI-RTOS](#)
- [TMS320C6657 产品文件夹](#)
- [TMS320C6678 产品文件夹](#)
- 德州仪器 (TI) : [《SYS/BIOS \(TI-RTOS 内核 \) 用户指南》](#)
- 德州仪器 (TI) : [《将 DSP/BIOS 5 应用迁移到 SYS/BIOS 6》](#)

10 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision * (May 2020) to Revision A (February 2021)	Page
• 更新了整个文档的表、图和交叉参考的编号格式.....	1

重要声明和免责声明

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 产品的应用。严禁对这些资源进行其他复制或展示。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

Copyright © 2022，德州仪器 (TI) 公司