

适用于 **SimpleLink™ MSP432™** 微控制器的 **Code Composer Studio™ IDE 7.1+**

本手册介绍 TI Code Composer Studio™IDE 7.1 及更高版本的使用方法，该产品适用于 SimpleLink™ MSP432™低功耗微控制器。本手册仅介绍适用于 Windows®操作系统的 Code Composer Studio IDE。适用于 Linux®和 OS X®操作系统的 Code Composer Studio IDE 版本与之类似，因此不再单独介绍。

本指南中的大多数说明适用于 Code Composer Studio IDE 7.1 以下版本，但 SimpleLink 软件开发套件 (SDK) 要求使用 Code Composer Studio IDE 7.1 或更高版本。

内容

1	安装 Code Composer Studio IDE.....	4
2	更新 Code Composer Studio IDE.....	4
3	创建基本的 SimpleLink MSP432 项目	5
4	从 TI 资源浏览器导入 SimpleLink 示例	7
5	调试应用程序	8
6	使用串行线输出 (SWO) 硬件跟踪分析器	15
7	EnergyTrace™ 技术	17
8	器件安全	29
9	低功耗调试	34
10	常见问题解答.....	37
11	有关 Code Composer Studio IDE 的附加信息.....	38
12	参考文献	39

附图目录

1	检查更新	4
2	创建一个新 Code Composer Studio IDE 项目	5
3	新建项目向导	6
4	新建项目文件	6
5	新建 CCS 项目，打开资源浏览器	7
6	在 TI 资源浏览器中选择一个示例	8
7	项目属性	9
8	选择调试器连接.....	10
9	GNU 编译器和 MSP-FET 的调试设置	11
10	ROM 调试的预定义符号	12
11	启动调试会话.....	13
12	调试会话	14
13	目标配置	15
14	连接属性	15
15	脉冲密度和电流流量	17
16	EnergyTrace™ 技术首选项	18
17	EnergyTrace™ 技术控制条	19
18	带 EnergyTrace+ 图形的调试会话	20
19	系统配置窗口.....	21

20	状态窗口	21
21	能量窗口	22
22	电能窗口	22
23	带 EnergyTrace 图形的调试会话	23
24	EnergyTrace 系统配置窗口	24
25	放大功率窗口	24
26	当前的系统配置（蓝色）与记录的系统配置（黄色）	24
27	恢复（黄线）和自由运行（绿线）中同一程序的电能系统配置	25
28	在 EnergyTrace+ 模式下比较系统配置	26
29	在 EnergyTrace 模式下比较系统配置	26
30	显示目标配置视图	29
31	目标配置列表	30
32	启动所选目标配置	30
33	启动目标配置后的调试视图	30
34	显示所有内核 (Show all cores)	31
35	MSP432 中的所有内核列表	31
36	手动连接到 DAP	32
37	DAP 已连接	32
38	执行恢复出厂设置脚本	32
39	批量擦除脚本控制台输出	33
40	Code Composer Studio IDE 工具 (Tools) → GEL 文件 (GEL Files)	33
41	带密码的恢复出厂设置 GEL 文件	34
42	属性 (Properties) 菜单	35
43	启用低功耗运行	36
44	CPU 内核状态显示指示深度休眠模式	36
45	程序计数器位于 WFI 指令处	37
46	将调试器设置更改为 SWD	37

商标

Code Composer Studio, SimpleLink, MSP432, E2E, LaunchPad are trademarks of Texas Instruments.
CoreSight is a trademark of ARM Ltd.
ARM) is a registered trademark of ARM Ltd.
OS X is a registered trademark of Apple Inc.
IAR Embedded Workbench is a registered trademark of IAR Systems.
Linux is a registered trademark of Linus Torvalds.
Windows is a registered trademark of Microsoft Corporation.
All other trademarks are the property of their respective owners.

前言：请先阅读以下内容

如何使用本手册

本手册仅介绍特定于 SimpleLink MSP432 低功耗微控制器 的那部分 Code Composer Studio IDE 功能，并未全面介绍 MSP432 微控制器或完整的开发软件和硬件系统。有关上述条目的详细信息，请参见[网站上的重要 MSP432 文档](#)中列出的相关 TI 文档。

网站上的重要文档

有关 MSP432 微控制器的主要信息来源包括器件特定的数据表和技术参考手册。[MSP432 网站](#)提供上述文档的最新版本。

有关介绍 Code Composer Studio 工具（Code Composer Studio IDE、汇编器、C 编译器、链接器和库管理器）的文档，请访问 www.ti.com/tool/cn/ccstudio。Code Composer Studio IDE 的 Wiki 页面 (FAQ) 链接如下：processors.wiki.ti.com/index.php/Category:CCS；[TI E2E™ 社区](#)支持论坛可为您提供更多帮助。

有关第三方工具（例如，IAR Embedded Workbench®（适用于 ARM）®或 Segger J-Link 调试探针）的文档，请分别访问各自企业的网站。

如果您需要协助

德州仪器 (TI) 产品信息中心 (PIC) 提供对 MSP432 器件和硬件开发工具的技术支持。有关 PIC 的联系信息，请访问 [TI 网站](#)。针对 MSP432 的 [TI E2E 社区](#)支持论坛提供了与同行工程师、TI 工程师和其他专家进行开放式沟通交流的机会。有关其他器件特定信息，请访问 [MSP432 网站](#)。

1 安装 Code Composer Studio IDE

可访问 [TI 网站](http://www.ti.com) 获取 Code Composer Studio IDE。

- Code Composer Studio IDE 6.1 及更高版本支持 MSP432 低功耗微控制器。先前版本不支持 MSP432 MCU。
- 要充分利用 SimpleLink 生态系统，需要使用 Code Composer Studio IDE 7.1 或更高版本。

2 更新 Code Composer Studio IDE

在开始使用 MSP432 低功耗微控制器之前，请使用 Code Composer Studio IDE 更新功能更新安装。要检查更新，请单击帮助 (Help) → 检查更新 (Check for Updates)。

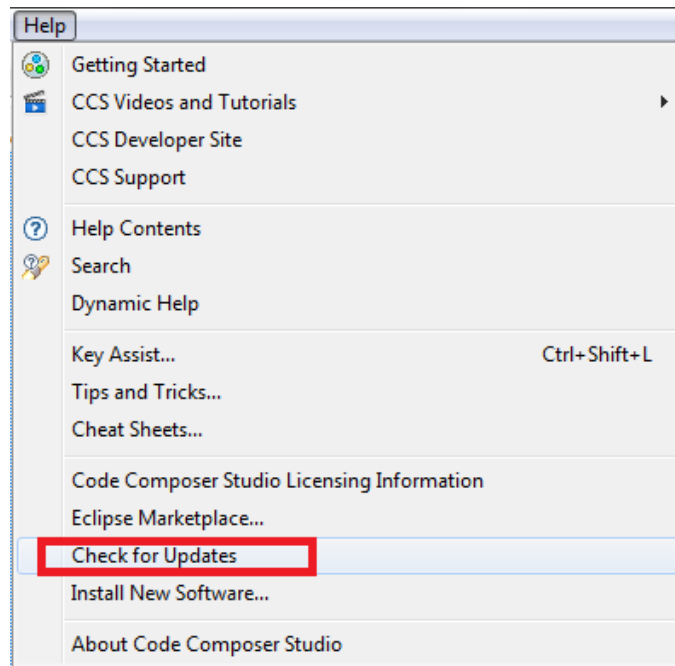


图 1. 检查更新

Code Composer Studio IDE 连接到 TI 更新服务器，并检索有关系统相关更新的信息。可以有选择地安装更新，但最好安装所有更新。

3 创建基本的 SimpleLink MSP432 项目

Code Composer Studio IDE 在工作区中编排项目。首次启动 Code Composer Studio IDE 时，会自动生成新的工作区。此工作区为空。

可以创建一个全新的项目或者使用 4 节中所述的 SimpleLink MSP432 SDK 示例之一。TI 建议使用示例之一。

要在当前工作区中创建一个新项目：

1. 单击文件 (**File**) → 新建 (**New**) → CCS 项目 (**CCS Project**)。

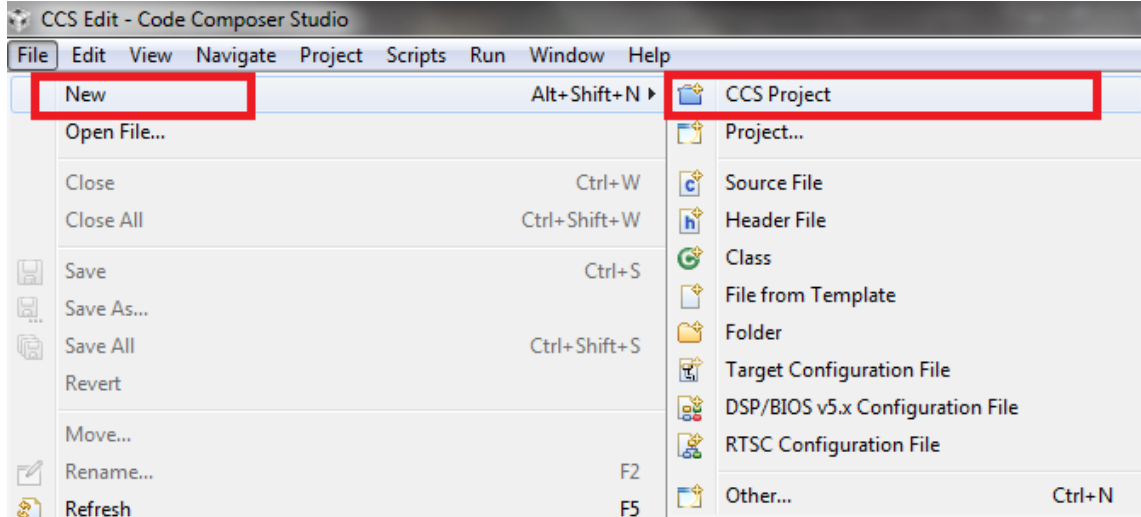


图 2. 创建一个新 Code Composer Studio IDE 项目

2. 选择 **MSP432** 作为目标系列（请参见图 3），这可以限制器件下拉列表中列出的器件型号。
3. 选择正在使用的器件；例如，**MSP432P401R**。
4. 选择要使用的调试连接。在以下示例中，调试连接为 **XDS110** 调试探针。稍后可以在项目属性中修改上述设置。
5. 键入唯一的项目名称。
6. 单击完成 (**Finish**)。新项目完成创建后，会有多个文件随之一一起复制到工作区中。

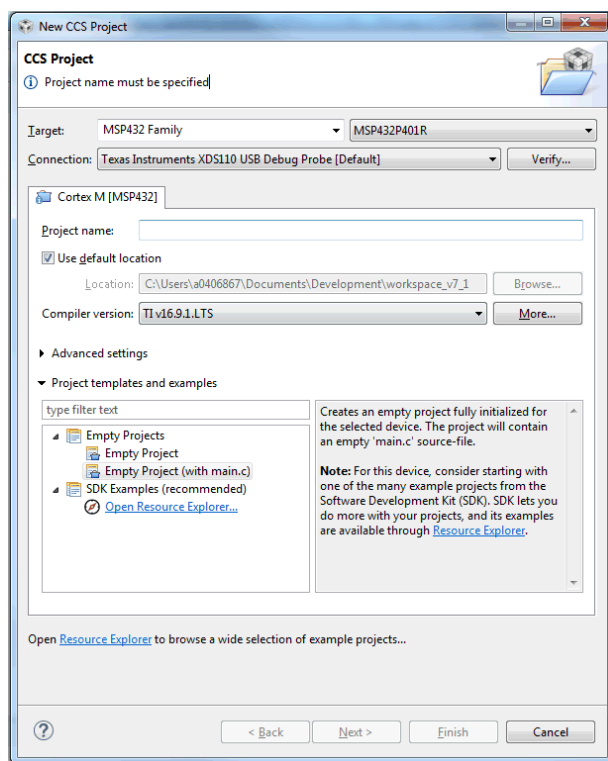


图 3. 新建项目向导

此时，工作区包含一个新创建的项目 ()，其中包括：

- 一个基本的 **main.c** 文件（前提是在新建 CCS 项目 (New CCS Project) 对话框中选择了该文件）
- 中断向量文件 **mcp432_startup_ccs.c**，其中预定义了所有中断处理程序
- 链接器命令文件 **mcp432p401r.cmd**
- 目标配置文件 **MSP432P401R.ccxml**，其中包含指向 XDS-110 调试探针的链接

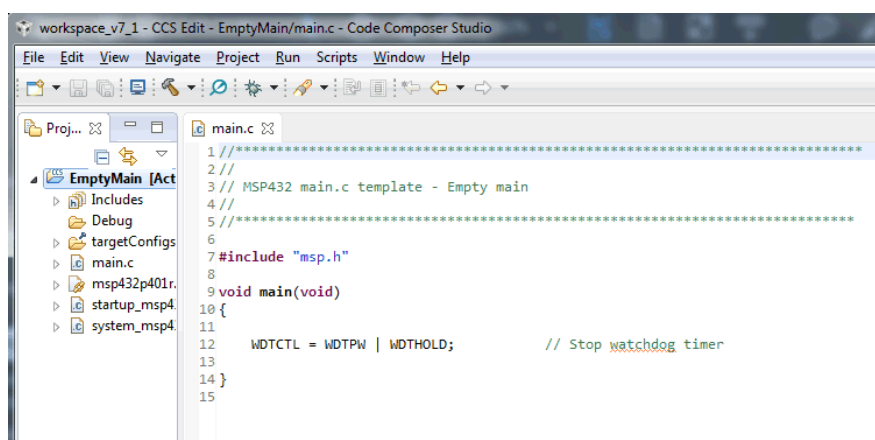


图 4. 新建项目文件

现在即可着手开发代码。

4 从 TI 资源浏览器导入 SimpleLink 示例

在 Code Composer Studio IDE 中启动新项目时，建议使用 TI 资源浏览器中提供的示例并进行调整。TI 资源浏览器使 SimpleLink MSP432 SDK 中的所有示例和文档都可以在 Code Composer Studio IDE 中使用，其中包含驱动程序、RTOS 以及从裸机到高级 API 的示例。

可通过访问 <http://dev.ti.com/tirex/> 在线获取 TI 资源浏览器。此外，Code Composer Studio IDE 7.0 及更高版本中随附 TI 资源浏览器。在 Code Composer Studio IDE 7.1+ 中，可选择查看 (View) → 资源浏览器 (Resource Explorer) 或创建一个新项目并选择 SDK 示例 (SDK Examples) → 打开资源浏览器 (Open Resource Explorer) (请参见图 5)。

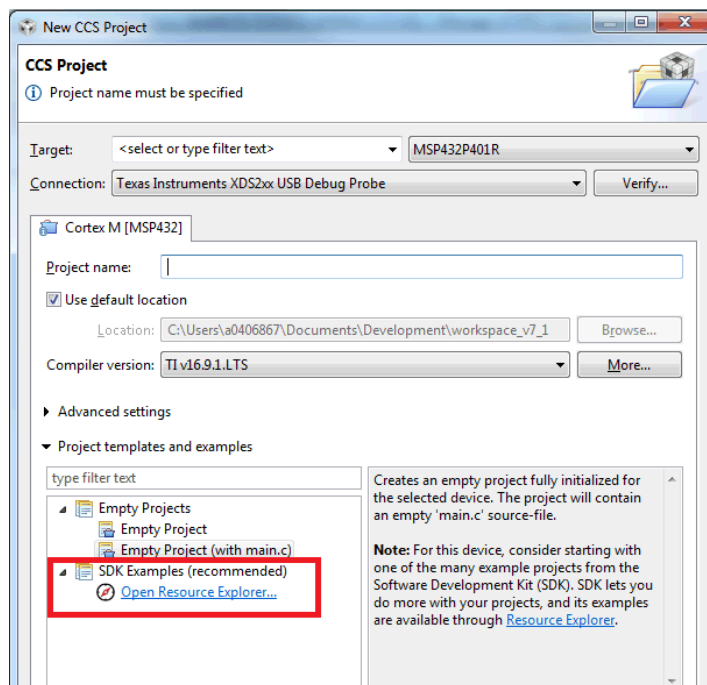


图 5. 新建 CCS 项目，打开资源浏览器

在 TI 资源浏览器中，在搜索字段中选择 MSP432 器件可以只列出适用于该器件的示例，以便用户可以更轻松找到适合的示例。例如，可在该层级结构中找到基本 ADC 示例 (请参见图 6)。

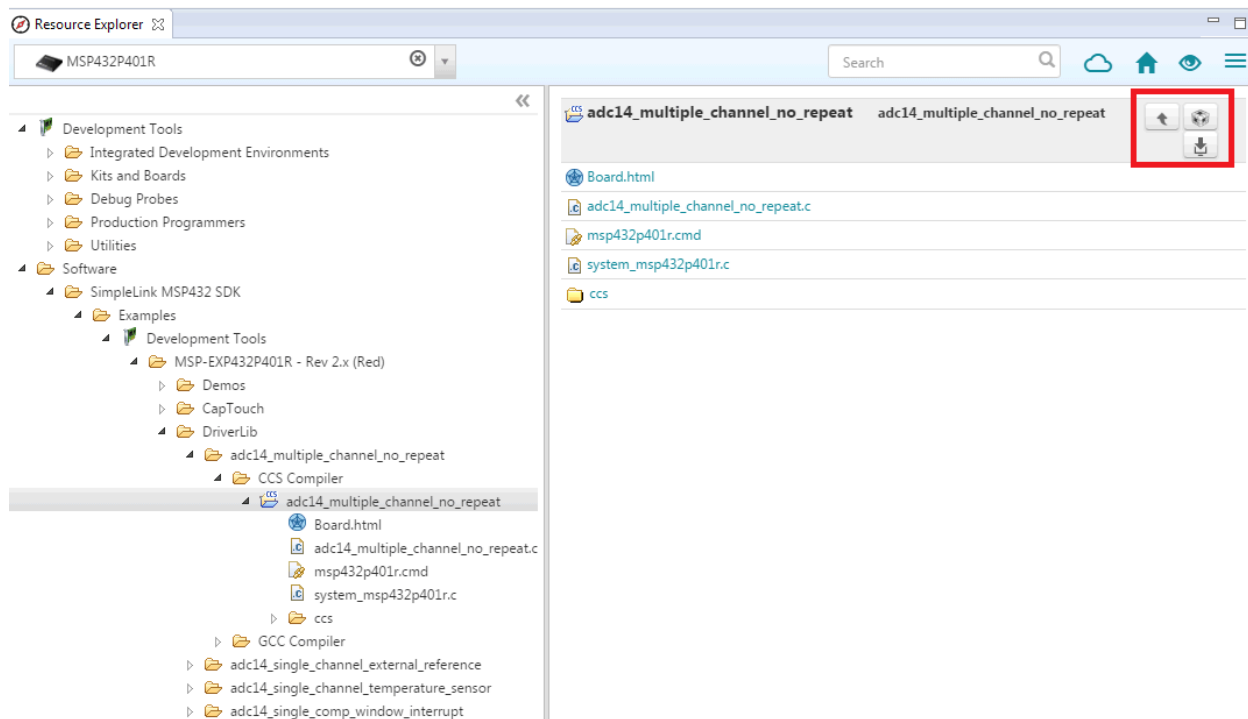


图 6. 在 TI 资源浏览器中选择一个示例

单击右上角的一个按钮（请参见图 6）下载 SDK 并将示例导入 Code Composer Studio IDE 中。当 SDK 安装到本地默认位置后，所有示例均可直接从已安装的 [SimpleLink MSP432 SDK](#) 中获取。

此外，可以使用项目浏览器中的导入对话框直接从 SDK 安装路径下导入示例。导入后，可以继续编译或扩展示例。

SDK 的所有文档均存储在 docs 文件夹的 SDK 安装路径下。从该文件夹中打开 [Documentation_Overview.html](#)，然后导航至 IDE 的快速入门指南。

一次性集成 SimpleLink 平台后，用户可以将产品组合中器件的任何组合添加至您的设计中。SimpleLink 平台的最终目标是确保设计要求变更时，完全重复使用代码。更多详细信息，请访问 www.ti.com/simplelink。

5 调试应用程序

以下调试探针已使用 Code Composer Studio IDE 成功测试。

- 德州仪器 (TI) XDS100v2、XDS100v3、XDS200、XDS110（包括 [XDS110 独立探针](#)）
- 德州仪器 (TI) MSP-FET（有关详细信息，请访问 [MSP-FET 页面](#)）
- Segger J-Link（有关详细信息，请访问 [TI J-Link 支持仿真器 wiki 页面](#)）

在 Code Composer Studio IDE 6.1.3.x 中，可从 Code Composer Studio IDE 应用程序中心直接链接 Segger 安装说明。

5.1 调试器设置

创建项目后，可以更改用于调试应用程序的调试器。要更改调试器，请在工作区中右键单击项目，然后选择属性 (Properties)（请参见图 7）。

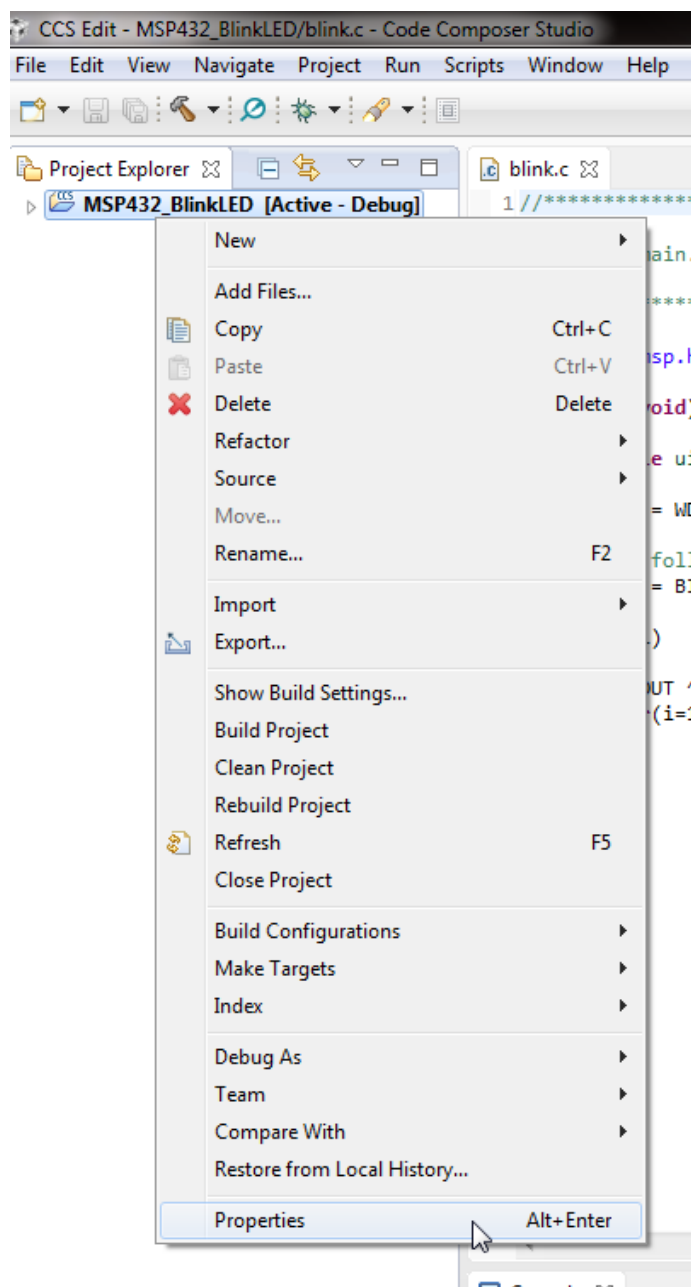


图 7. 项目属性

在项目属性窗口中，确保位于常规 (**General**) 选项窗格。在此可以看到目标连接的下拉列表。从此列表中选择调试器，然后单击确定 (**OK**)。

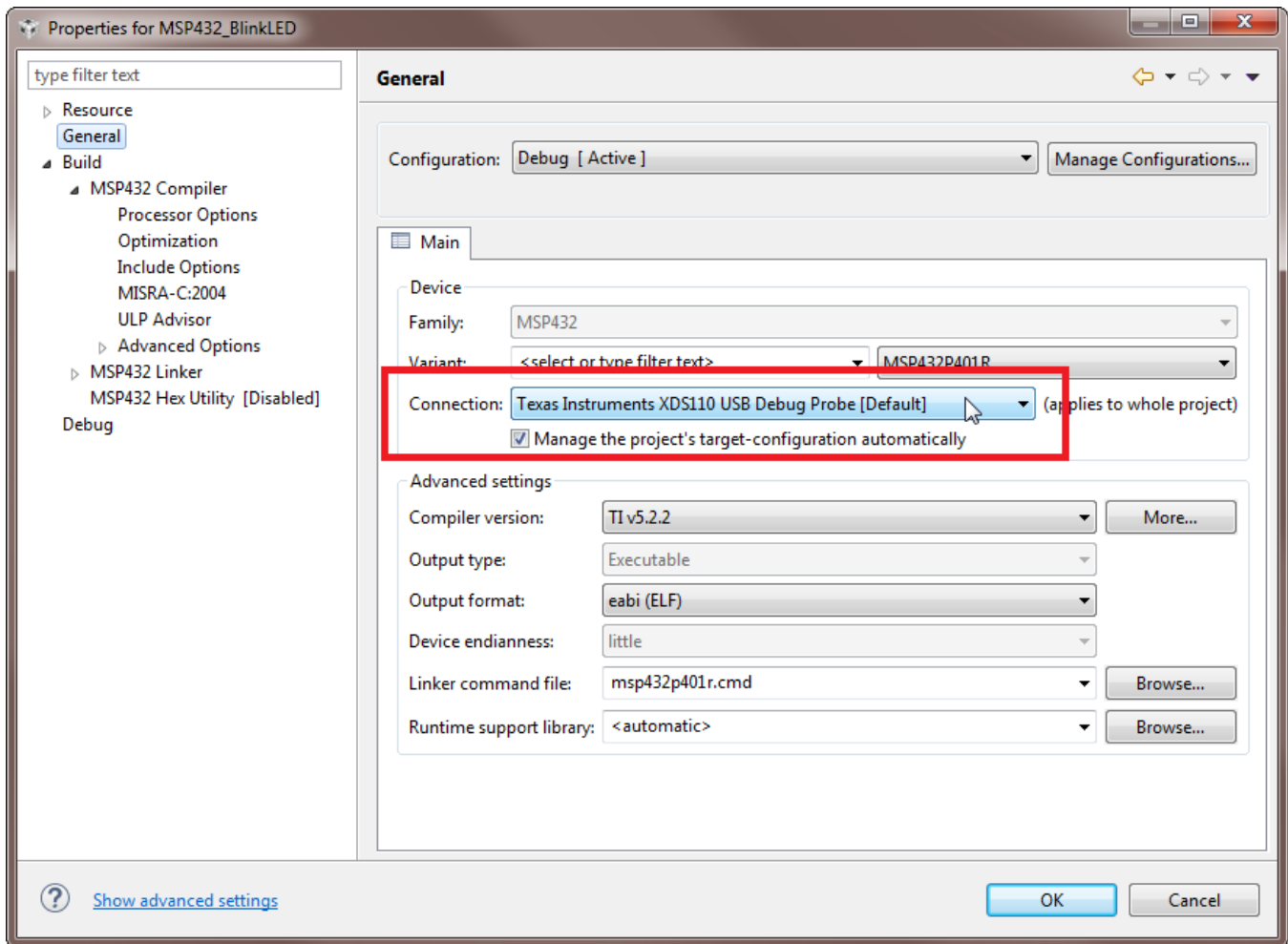


图 8. 选择调试器连接

更多调试器设置可在目标配置设置中完成。例如，从 JTAG 切换到使用较少引脚的串行线调试 (SWD)。有关上述设置的详细信息，请参见 6 节。

注：图 9 显示了使用 MSP-FET 和 GNU Linaro 编译器时调试所需的附加设置（转到项目属性 (project properties) → 调试 (Debug) → 程序/存储器加载选项 (Program/Memory Load Options)）。

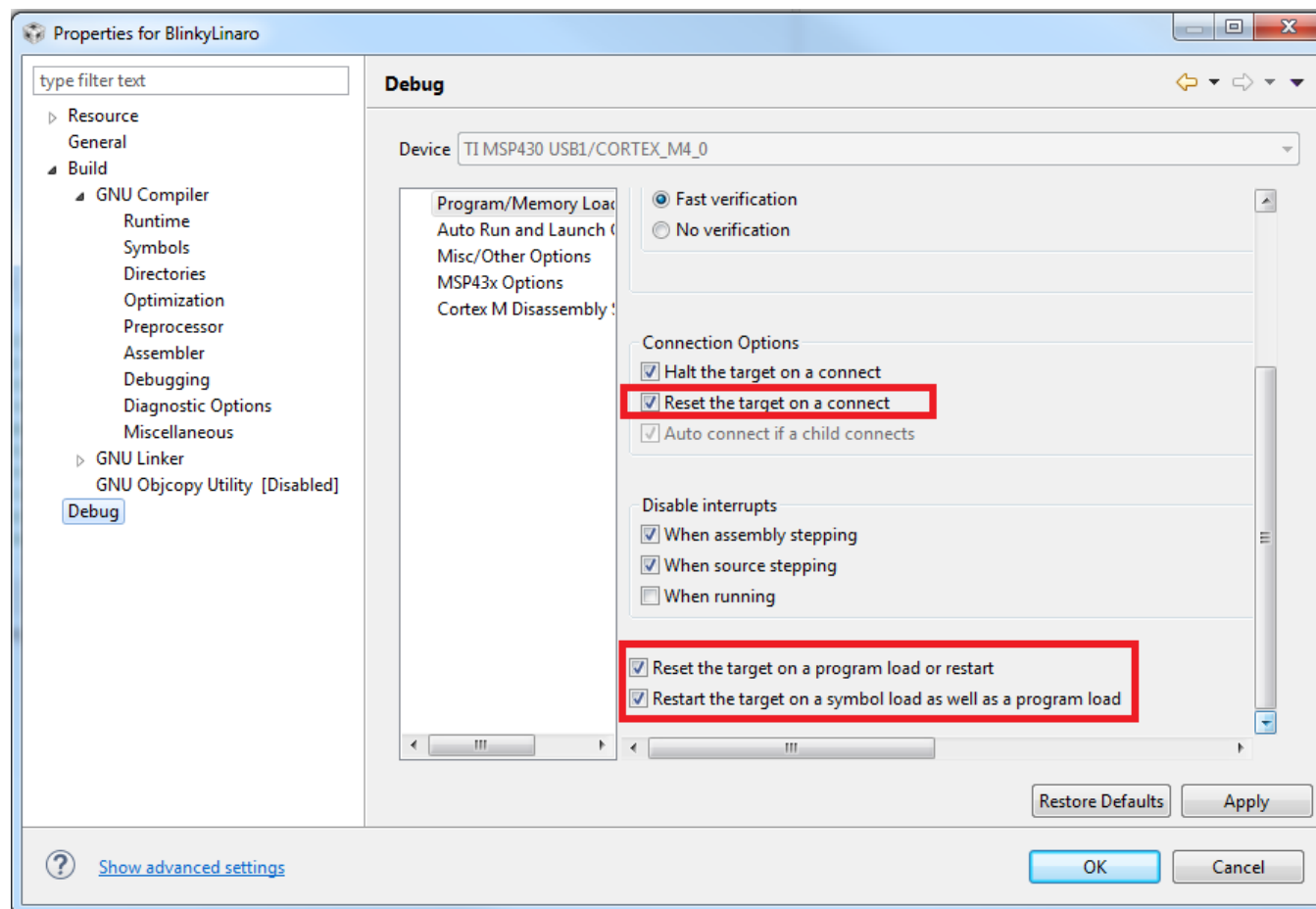


图 9. GNU 编译器和 MSP-FET 的调试设置

5.2 调试 ROM 驱动程序库

MSP432P4xx 系列包含一个完全集成在 ROM 存储器中的完整外设驱动程序库 (DriverLib)。开发人员可以利用 ROM DriverLib 实现诸多优势，包括访问高度稳健耐用且经过测试的 API，具有较低功耗的单周期 ROM 执行速度，以及释放存储器空间用于存储更多应用程序代码。开发人员可以通过将 DriverLib 头文件添加到项目并链接到预编译的库来访问 ROM API。

此时，驱动程序库源代码成为 SimpleLink MSP432 SDK 的一部分。驱动程序库是 TI 驱动程序（也是 SDK 的一部分）下面的低级软件层。在 SDK 中，DriverLib 源代码位于 `<SDK_InstallationPath>\source\ti\devices\msp432p4xx\driverlib` 下。

有关 MSP432P4xx 驱动程序库的更多信息以及 ROM DriverLib 中提供的内容，请参见 [SimpleLink MSP432 SDK](#) 中的文档。

要调试 ROM 驱动程序库，请确保 `TARGET_IS_MSP432P4XX` 已在高级选项 (Advanced Options) → 预定义符号 (Predefined Symbols) 中列出（请参见图 10）。

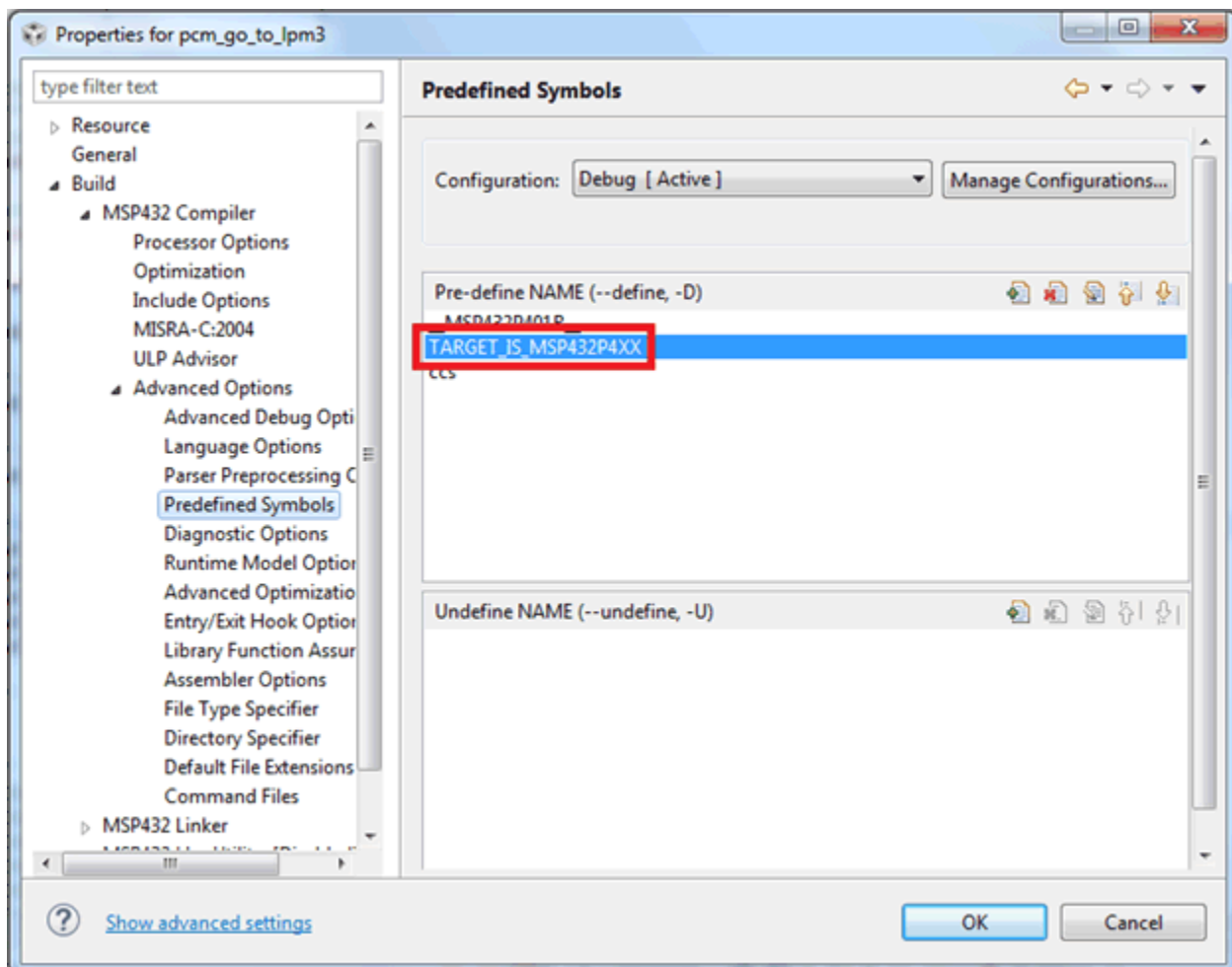



图 10. ROM 调试的预定义符号

5.3 启动调试会话

为项目设置正确的调试器后，即可开始调试应用程序。要启动调试会话，请单击 IDE 顶部工具栏中的  图标（请参见图 11）。

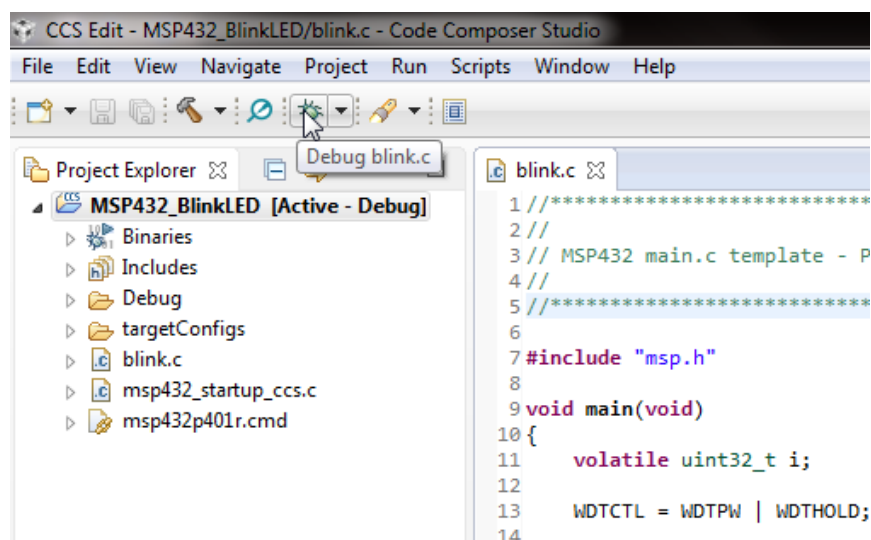


图 11. 启动调试会话

IDE 完成编译、链接和将代码下载到器件后，将启动调试会话，此时用户即可开始调试应用程序。

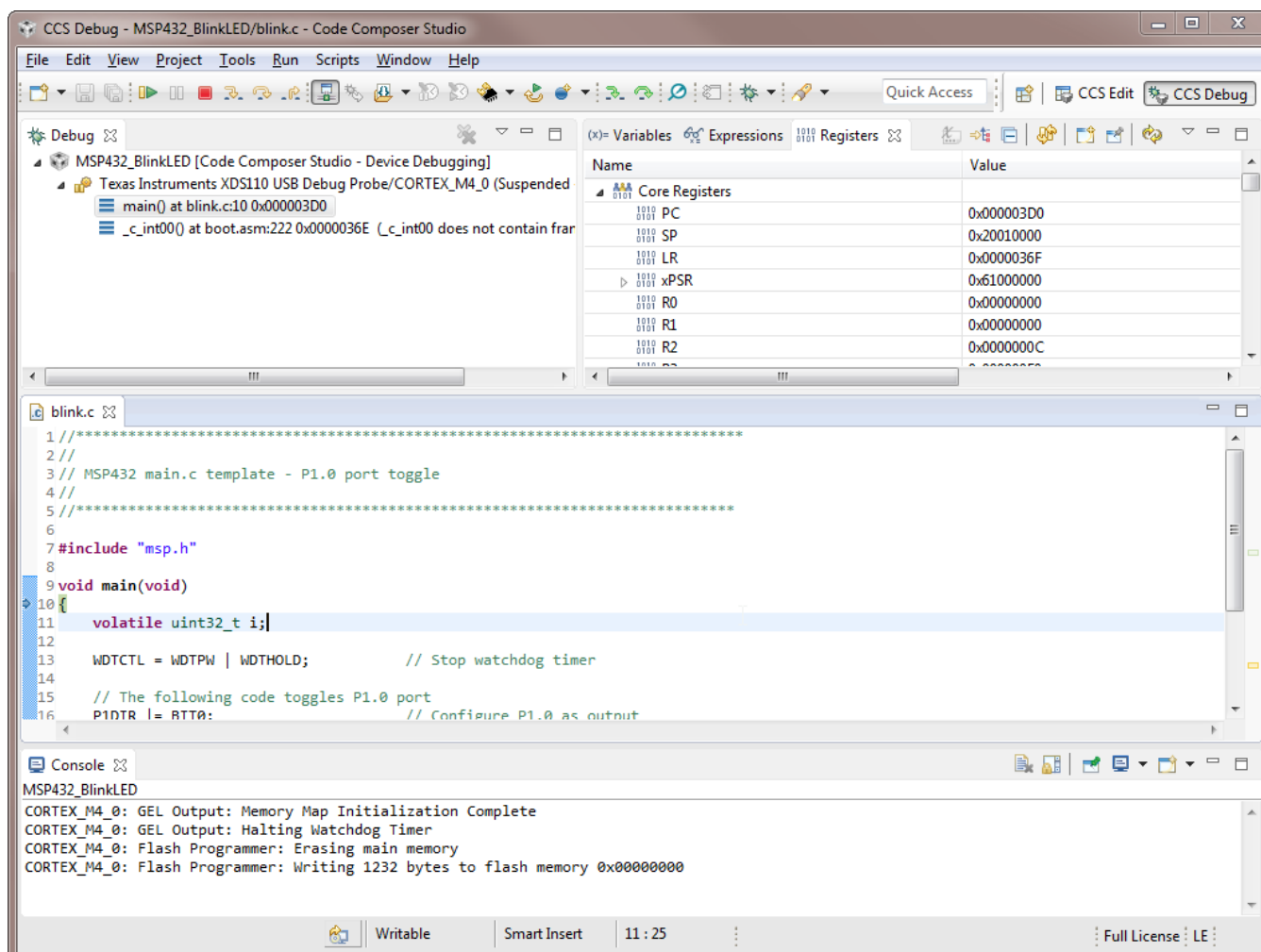


图 12. 调试会话

6 使用串行线输出 (SWO) 硬件跟踪分析器

在 Code Composer Studio IDE 中, MSP432 MCU 的 SWO 跟踪工具 通过使用 ARM CoreSight™组件的功能来实现, 尤其是仪器跟踪宏单元 (ITM) 以及数据观测点和跟踪单元 (DWT) (MSP432 MCU 中目前尚未实现 ETM)。

本用户指南仅介绍如何在 Code Composer Studio IDE 中启用 SWO 跟踪。有关跟踪硬件和示例用例的详细信息, 请参见参考文献 [3]。

6.1 为 SWO 跟踪配置项目

要启用 SWO 跟踪:

1. 在项目浏览器中展开项目, 然后打开 targetConfigs 文件夹中的 *.ccxml 项目文件。选择正确的调试探针 (请参见图 13)。

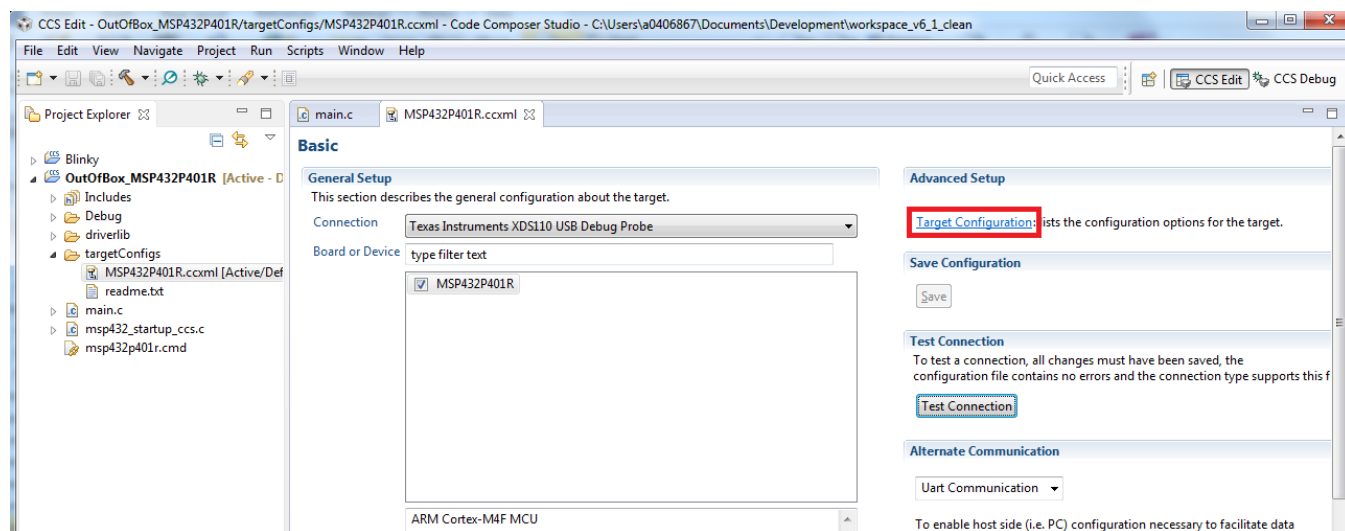


图 13. 目标配置

2. 单击高级设置 (Advanced Setup) 区域中的目标配置 (Target Configuration), 然后单击所选的调试探针。
3. 图 14 显示了连接属性 (Connection Properties)。在 JTAG/SWD 模式下, 选择 SWD 模式 (SWD Mode) - 辅助 COM 端口为目标 TDO 引脚 (Aux COM port is target TDO pin)。

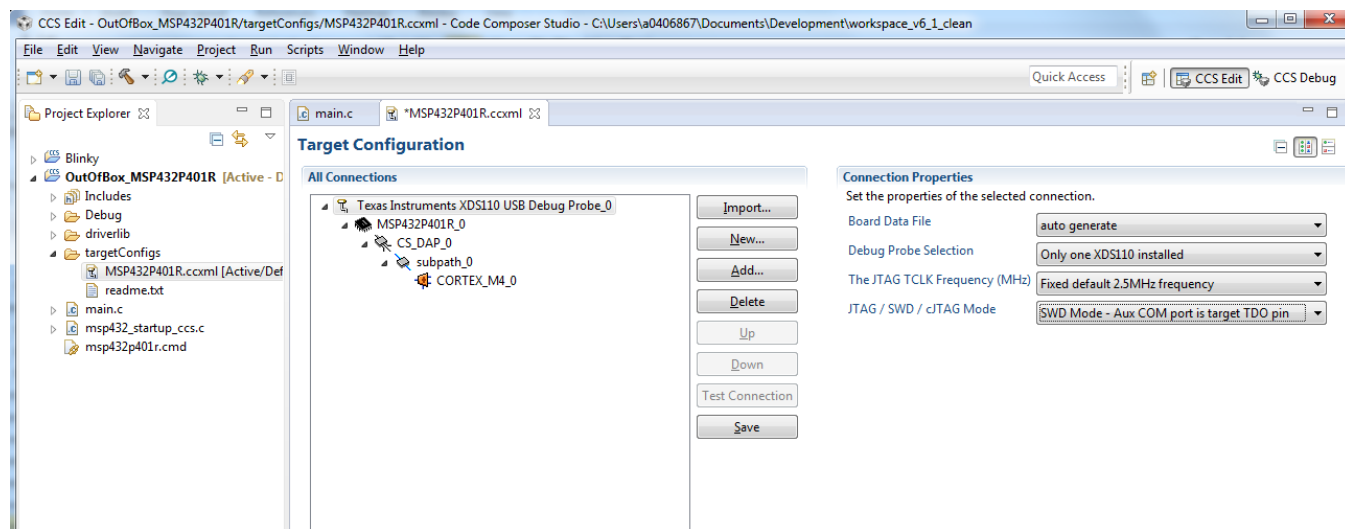


图 14. 连接属性

6.2 运行串行线跟踪

完成 SWO 配置后，启动调试会话以开始跟踪：

1. 通过从项目的快捷菜单中选择**编译项目 (Build Project)** 或选择锤子图标来编译项目。
2. 输入调试会话：右键单击项目名称，选择调试为 **(Debug As) → Code Composer Studio 调试会话 (Code Composer Studio Debug Session)**。或者，单击缺陷图标。
3. 在调试模式下，转到工具 **(Tools) → 硬件跟踪分析器 (Hardware Trace Analyzer)**，并选择以下选项之一：
 - 统计函数分析，用于分析每个函数的调用频率以及 CPU 周期的使用百分比
 - 数据变量跟踪，用于借助 ITM 获取有关变量值的图形
 - 中断分析，用于获取与何时发生中断以及它们如何相互干扰有关的数据。
 - 定制核心跟踪，用于在指定的代码位置跟踪用户定义的字符串

有关详细信息，请参见 [MSP432™ 调试工具：使用 CCS 跟踪分析器实现的串行线输出](#)。

7 EnergyTrace™ 技术

EnergyTrace™ 技术是基于电能的代码分析工具，用于测量和显示应用的电能系统配置，并帮助优化应用以实现超低功耗。

MSP432 器件采用内置 **EnergyTrace+[CPU 状态]**（简称 **EnergyTrace+**）技术，可在执行用户程序代码的同时实时监测内部器件状态。精品 MSP432 器件和调试器支持 EnergyTrace+ 技术。

EnergyTrace 模式（不带“+”号）是 **EnergyTrace** 技术的基础，支持通过模拟电能测量来确定应用的能耗，但不会将能耗与内部器件信息相关联。EnergyTrace 模式适用于所有带精品调试器的 MSP432 器件（包括 Code Composer Studio IDE）。

7.1 电能测量

具有 EnergyTrace 技术支持的调试器含有一个全新且独特的方法来持续测量施加到目标微控制器的电能，该方法与在不连续的时间内通过放大和采样分流电阻器上的压降的常见方法有很大不同。一个由软件控制的 dc-dc 转换器用于生成目标电源。此直流-直流转换器充电脉冲的时间密度与等于目标微控制器的能耗。一个内置高速数据传输错误纠正校准电路定义与一个单直流-直流充电脉冲等效的电能。

图 15 显示能量测量原理。每个时间单位内充电脉冲数量少的周期表示低能耗，因此电流低。每个时间单位内充电脉冲数量多的周期表示高能耗，因此电流高。每个电荷脉冲导致一个输出电压 $V_{\text{输出}}$ 的上升，这将引起一个对于所有直流-直流转换器常见的不可避免的电压纹波。

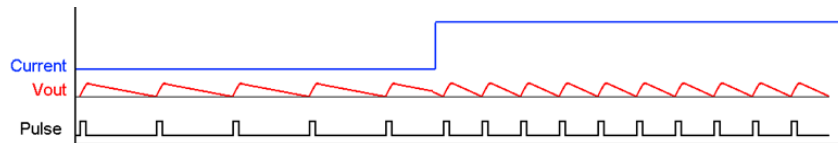


图 15. 脉冲密度和电流流量

持续采样的优势是很明显的：即使是耗能的最短器件活动也会对总体记录能耗产生影响。非并联测量系统可实现这一目的。

7.2 与 Code Composer Studio IDE 相集成

EnergyTrace 技术会作为德州仪器 (TI) 适用于 MSP432 微控制器的 Code Composer Studio IDE 的一部分提供。调试应用期间，如果硬件支持 EnergyTrace 技术，可提供额外的窗口。

7.3 启用 EnergyTrace 技术并选择默认模式

默认情况下，Code Composer Studio 首选项中禁用 EnergyTrace 技术功能。要启用该功能，请转到窗口 (Window) → 首选项 (Preferences) → Code Composer Studio → 高级工具 (Advanced Tools) → EnergyTrace™ 技术 (EnergyTrace™ Technology)（请参见图 16）。

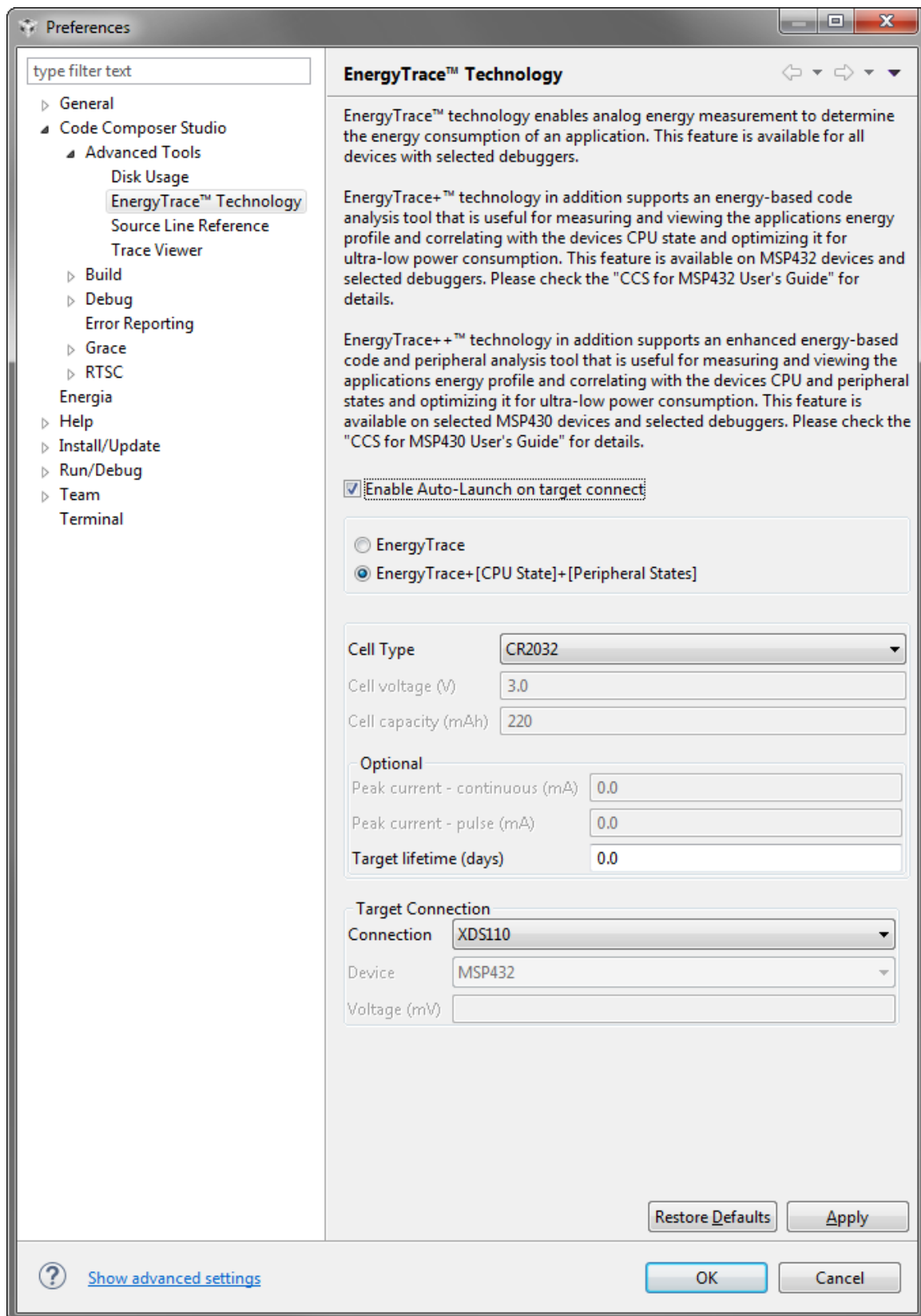



图 16. EnergyTrace™ 技术首选项

对于目标连接，请选择 XDS110，或者如果使用了支持 EnergyTrace 技术的 MSP-FET，则可选择 USB 连接之一。

支持以下两种捕获模式。

- 功能全面的 **EnergyTrace+[CPU 状态]** 模式，可提供实时器件状态信息以及电能测量数据
- **EnergyTrace** 模式，仅提供电能测量数据

使用单选按钮选择要在调试会话启动时启用的模式。如果 MSP432 器件不支持器件状态捕获，选择将被忽略，Code Composer Studio 将以 **EnergyTrace** 模式启动。

当调试会话激活时，单击系统配置 (Profile) 窗口中的  图标可以在两种模式间切换。

- 注：
- 如果在调试会话启动时未打开 EnergyTrace 技术窗口，请验证以下各项：
 - 硬件（调试器和器件）是否支持 EnergyTrace 技术？要确定所选器件是否支持 EnergyTrace 技术，请参见器件特定的数据手册或评估板随附的用户指南。
 - 是否在窗口 (Window) → 首选项 (Preferences) → Code Composer Studio → 高级工具 (Advanced Tools) → EnergyTrace™ 技术 (EnergyTrace™ Technology) 中全局启用了 EnergyTrace 技术？

7.4 控制 EnergyTrace 技术

EnergyTrace 技术可通过配置文件 (Profile) 窗口中的控制栏图标来控制（请参见图 17）。表 1 介绍了各个按钮的功能。

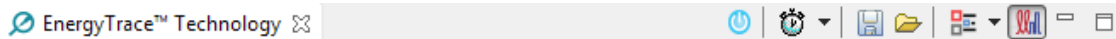


图 17. EnergyTrace™ 技术控制条

表 1. EnergyTrace™ 技术控制条图标

	启用或禁用 EnergyTrace 技术。当被禁用时，图标变成灰色。
	设定捕获周期：5 秒、10 秒、30 秒、1 分钟或 5 分钟。时间周期停止后，数据采集停止。然而，程序在调试控制窗口中的 Pause（暂停）按钮被点击前继续执行。
	将系统配置保存至项目目录。保存 EnergyTrace+ 系统配置时，默认文件名以“EnergyTrace_D”开头，后跟一个时间戳。保存 EnergyTrace 系统配置时，默认文件名以“EnergyTrace”开头，后跟一个时间戳。
	载入之前存入的系统配置进行比较。
	恢复曲线图或打开 Preferences 窗口。
	在 EnergyTrace+ 模式和 EnergyTrace 模式之间切换

7.5 EnergyTrace+ 模式

调试具有内置 EnergyTrace+ 支持的器件时，**EnergyTrace+** 模式提供有关目标微控制器的能耗和内部状态的信息。以下窗口在一个调试会话启动时打开：

- 系统配置
- 状态
- 电源
- 能源

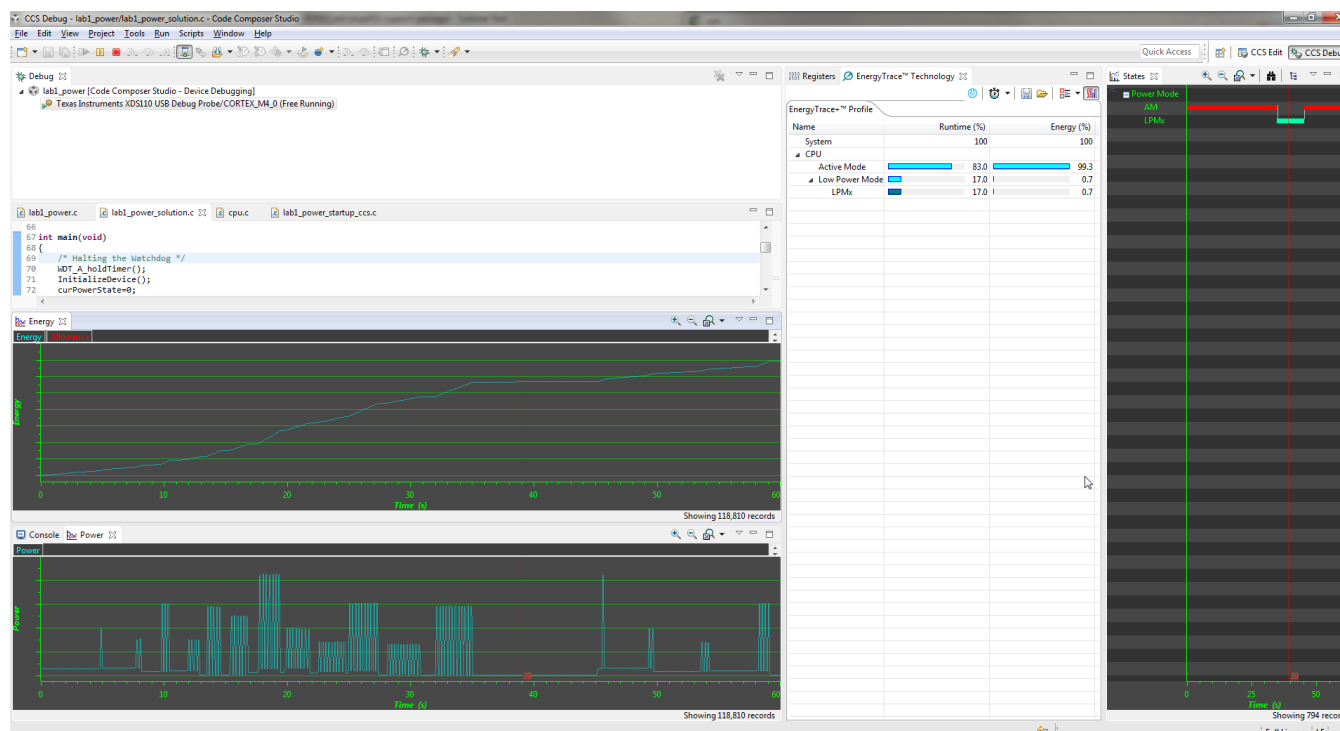


图 18. 带 EnergyTrace+ 图形的调试会话

系统配置 (**Profile**) 窗口 (请参见图 19) 是 EnergyTrace+ 的控制界面。它可被用来设定捕获时间或用来保存已捕获的数据供以后参考。**Profile** 窗口还显示了一个已捕获数据的压缩视图，并且可实现与之前数据的比较。

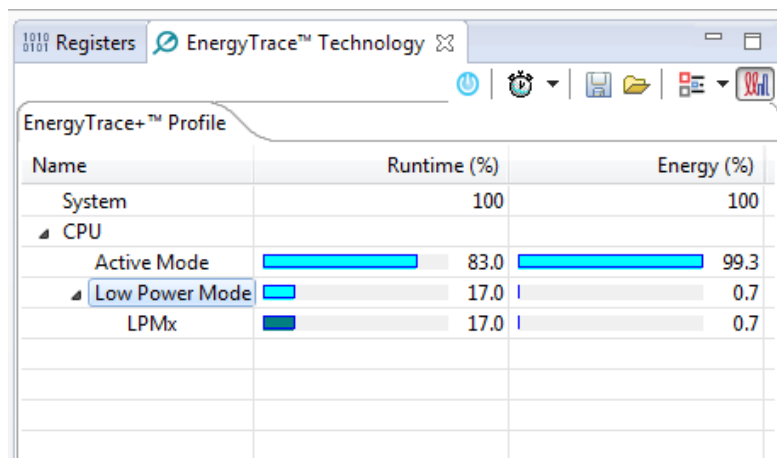


图 19. 系统配置窗口

系统配置 (Profile) 窗口提供了已配置应用的资源使用情况的快速概览。

- CPU: 显示与程序执行相关的信息
 - 低功耗模式: 显示低功耗模式使用摘要。
 - 激活模式: 显示激活模式期间, 哪些函数已经被执行。实时库中的函数在 `_RTS_ subcategory` 中分别列出。如果器件支持 IP 封装, 将显示一条标记为 <受保护> (<Protected>) 的线, 指示在 IP 封装内存外执行的时间。

States 窗口 (请见图 20) 显示捕获会话期间目标微控制器内部状态的实时跟踪。状态信息包括功率模式、外设模块的开关状态和系统时钟的状态。

图 20 显示了器件进入低功耗模式, 然后又返回到工作模式。通过状态 (States) 窗口可直接验证应用是否表现出预期的运行状态, 例如, 某个外设特定活动后被禁用。

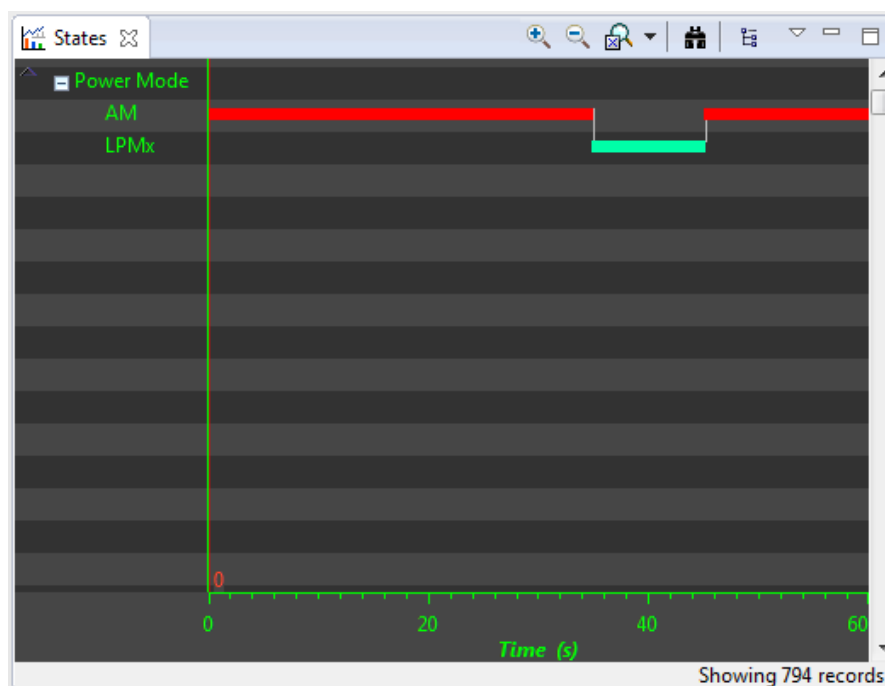


图 20. 状态窗口

Power 窗口（请见图 21）显示目标在时间段内的动态功耗。当前的系统配置用淡蓝色标出，而之前记录的系统配置已经被重新载入，以便与黄色标出的内容相比较。

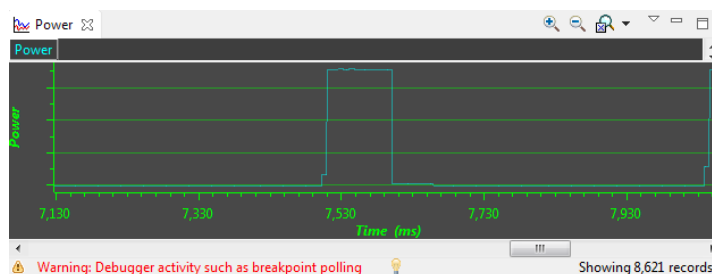


图 21. 能量窗口

Energy 窗口（请见图 22）显示目标在时间段内的累积能耗。当前的系统配置用淡蓝色标出，而之前记录的系统配置已经被重新载入，以便与黄色标出的内容相比较。

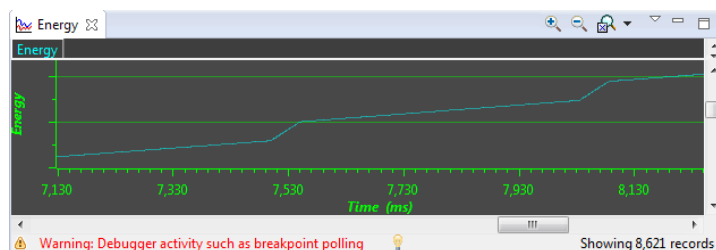


图 22. 电能窗口

注： 内部状态捕获期间，通过 JTAG 或 Spy-Bi-Wire 调试逻辑持续访问目标微控制器。这些调试访问耗能；因此，未在功率 (Power) 和电能 (Energy) 曲线图垂直轴上显示绝对功率数。要查看应用的绝对功率数，建议结合使用 **EnergyTrace** 模式与自由运行 (**Free Run**) 选项。在此模式下，测量能耗时不会访问目标微控制器的调试逻辑。

对器件来说，自由运行模式为非侵入式模式，但仍然使用调试端口，这会影响电能测量结果。为了获得最高精度，运行 **EnergyTrace** 时不要进行调试或采用自由运行模式。另请注意，无法进行精确测量时，电能和功率的图表显示为蓝色，而不是绿色。

7.6 EnergyTrace 模式

此模式允许对没有内置 EnergyTrace+ 支持的 MSP432 微控制器单独使用电能测量功能。它还可用于验证应用的能耗，而无需调试器活动。如果在首选项 (Preferences) 窗口中选择了 **EnergyTrace** 模式，则启动调试会话时将打开以下窗口（请参见图 23）：

- 系统配置
- 电源
- 能源

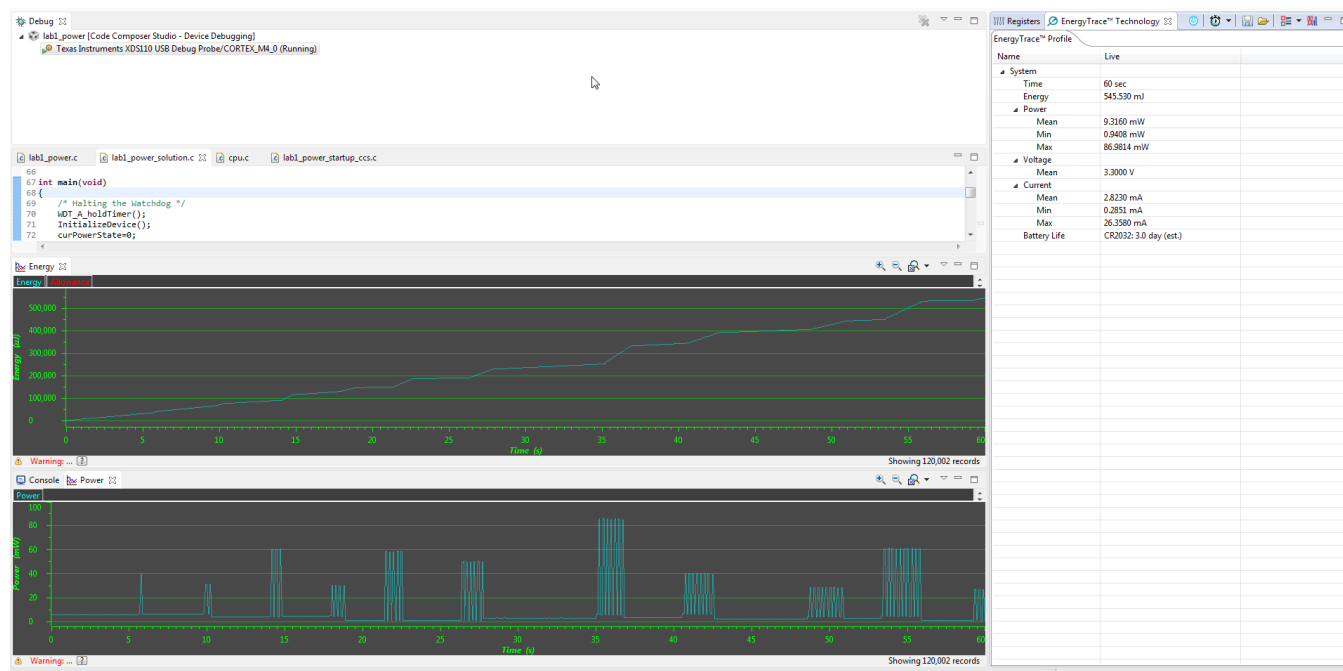


图 23. 带 EnergyTrace 图形的调试会话

在 **EnergyTrace** 模式下，系统配置 (Profile) 窗口将显示已配置应用的统计数据（请参见图 24）。以下参数显示：

- 已捕获时间
- 应用的总能耗（以 mJ 为单位）
- 最小、平均和最大功率（以 mW 为单位）
- 平均电压（单位为 V）
- 最小、平均和最大电流（单位为 mA）
- 这对捕获电能系统配置的 CR2032 电池估计的使用寿命（以天计）

注： 用来计算电池使用寿命的公式假定一个理想的 3V 电池，并且未将温度、老化、峰值电流和其它会对电池容量产生负面影响的因素考虑在内。更改目标电压时（例如，从 3.6V 到 3V）可能会使模拟电路运行方式有所不同，电路将以更高效或更低效的状态运行，因而会减少或增加能耗。系统配置 (Profile) 窗口中显示的值不能替代真实硬件上的测量值。

EnergyTrace™ Profile	
Name	Live
System	
Time	10 sec
Energy	14.61 mJ
Power	
Mean	1.75 mW
Min	1.068 mW
Max	7.943 mW
Voltage	
Mean	3.59 V
Current	
Mean	0.49 mA
Min	0.298 mA
Max	2.213 mA
Battery Life CR2032: 18.8 day (est.)	

图 24. EnergyTrace 系统配置窗口

功率 (Power) 窗口 (请参见图 25) 显示目标在时间段内的动态功耗。当前的系统配置用淡蓝色标出，而之前记录的系统配置已经被重新载入，以便与黄色标出的内容相比较。

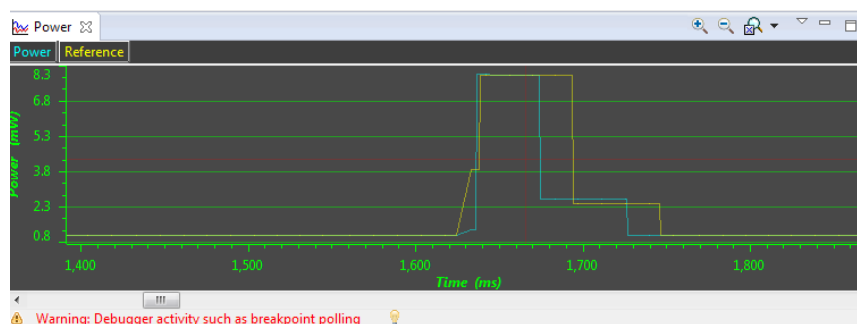


图 25. 放大功率窗口

Energy 窗口 (请参见图 26) 显示目标在时间段内的累积能耗。当前的系统配置用淡蓝色标出，而之前记录的系统配置已经被重新载入，以便与黄色标出的内容相比较。

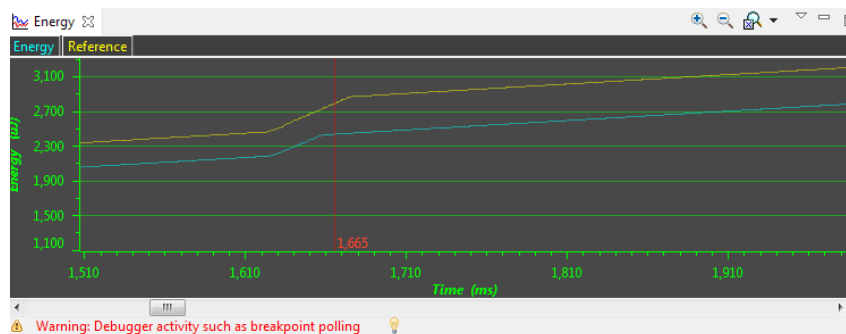


图 26. 当前的系统配置 (蓝色) 与记录的系统配置 (黄色)

注：使用调试器视图的恢复 (Resume) 按钮执行程序期间，通过 JTAG 或 Spy-Bi-Wire 协议持续访问目标微控制器，以检测何时命中断点。不可避免的是，这些调试访问将会在目标域内耗能并更改电能和功率图形所示的结果。要查看某个应用的绝对功耗，TI 建议使用自由运行模式。在自由运行模式下，不访问目标微控制器的调试逻辑。有关调试访问对能耗的影响的示例，请参见图 27。黄色曲线在恢复 (Resume) 模式下记录，绿色曲线在自由运行 (Free Run) 模式下记录。

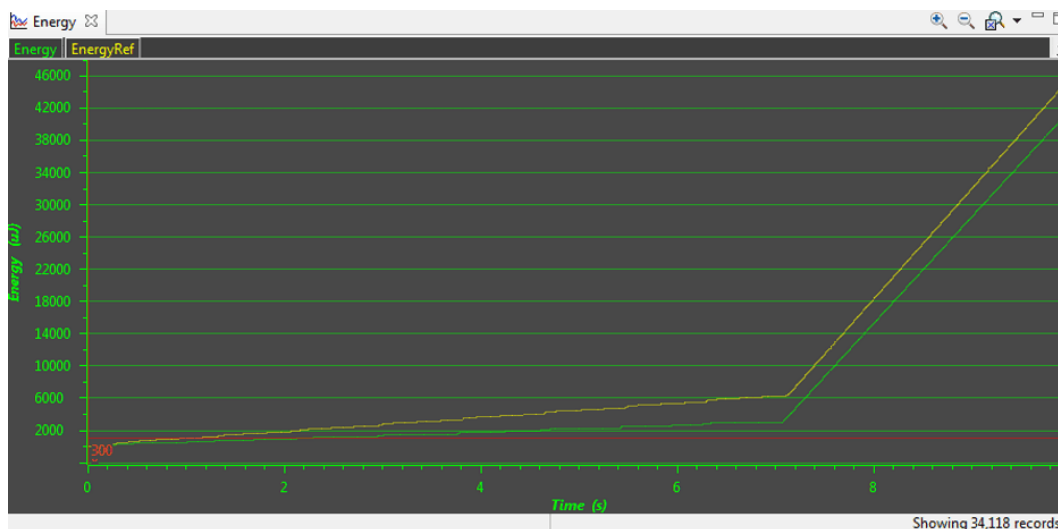


图 27. 恢复 (黄线) 和自由运行 (绿线) 中同一程序的电能系统配置

7.7 将捕获到的数据与基准数据相比较

EnergyTrace 技术有多种使用方法。一种方法是将器件在时间段内的内部状态与预期的运行方式相比较并且纠正任何错误运行方式；例如，由定期使用后未将外设禁用所导致的错误。另一种方法是将捕获的数据与之前捕获的数据进行比较。在以下讨论中，之前捕获的数据称为基准数据。

加载基准数据后，将在功率 (Power) 和能量 (Energy) 窗口中绘制黄色基准图形。功率 (Power) 窗口显示运行时间内两个数据集的功率系统配置，并且有助于确定静态功耗中的任何变化；例如，由于使用更深的低功耗模式或禁用未使用的外设而引起的变化。它还显示动态功耗是如何从一个测量值变化为另外一个测量值的；例如，由于执行了超低功耗顾问提示 (ULP advisor hint)。能量 (Energy) 窗口显示随时间累积的能耗并指示哪种系统配置的能效更高。

在 EnergyTrace+ 模式下，在系统配置 (Profile) 窗口中显示捕获数据和基准数据的压缩视图（请参见图 28）。您可以快速查看两次捕获会话之间总能耗以及功率模式、外设和时钟的使用是如何变化的。总的来说，变好的参数用绿条显示，而变坏的参数用红条显示。例如，激活模式中花费的时间通常被视为负面的。因此，如果一个代码变化使得应用花费在激活模式中的时间变少，那么负增量显示为绿条，并且在低功耗模式下花费的额外时间也显示为绿条。

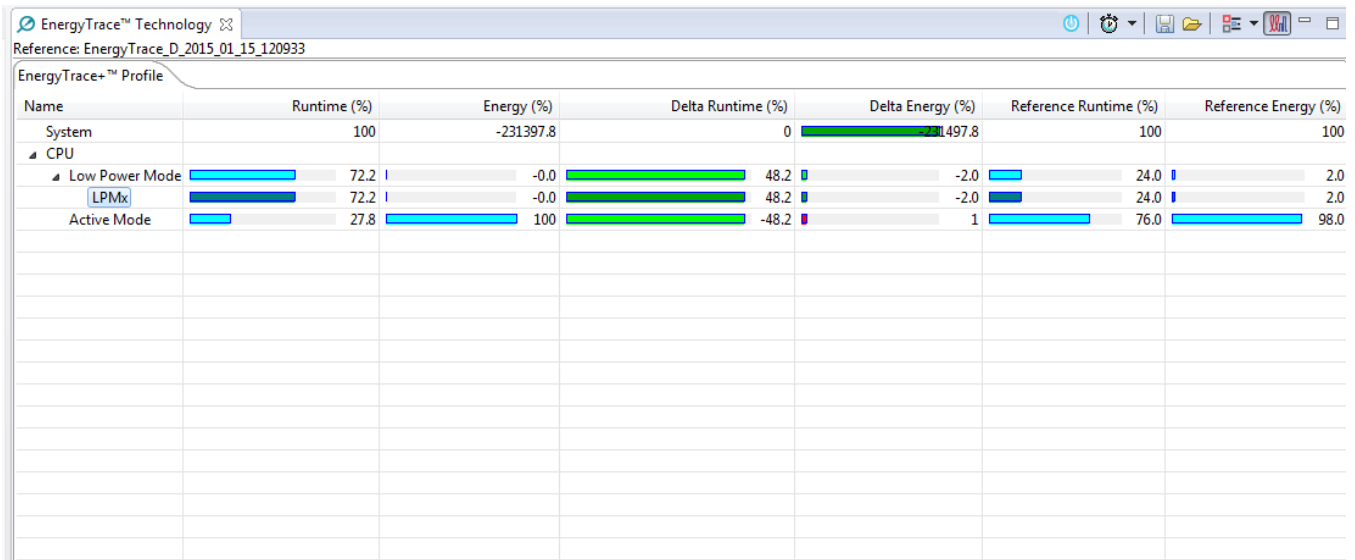


图 28. 在 EnergyTrace+ 模式下比较系统配置

在 **EnergyTrace** 模式下，不提供用于生成详尽报告的状态信息。但是，会将测量期间的总能耗与功率和电流的最小值、平均值和最大值进行比较。已经变好的参数用绿条显示，而变坏的参数用红条显示（请参见图 29）。

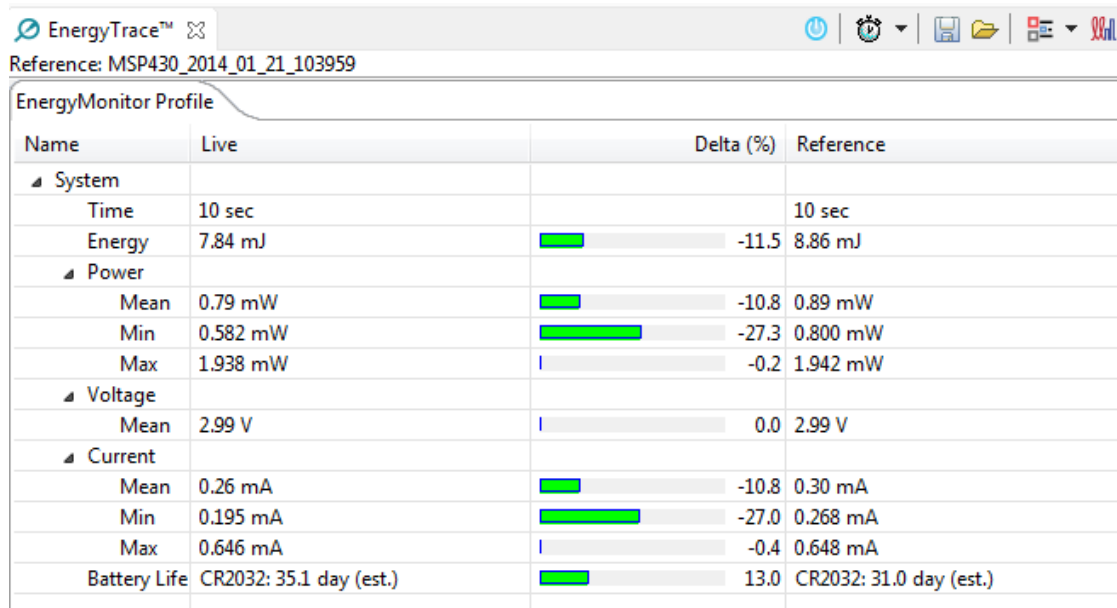


图 29. 在 EnergyTrace 模式下比较系统配置

增量条在 0% 至 50% 的区间呈线性变化。增量大于 50% 时不会产生更大的增量条。

7.8 EnergyTrace 技术常见问题解答

问: EnergyTrace+ 技术的采样频率为多少?

答: 采样频率取决于调试器和所选调试协议及其速度设置。其范围通常为 1kHz (例如, 使用设置为 SLOW 的 Spy-Bi-Wire 接口时) 到 3.2kHz (例如, 使用设置为 FAST 的 JTAG 接口时)。调试器会根据器件状态信息轮询 EnergyTrace+ 的状态信息。由于采样频率不同, 状态图中可能无法捕获具有短暂或快速占空比的活动外设状态。此外, 采样频率越高, 对 EnergyTrace 模式下的器件能耗影响越大。

问: EnergyTrace 技术的采样频率为多少?

答: 无论使用何种调试协议或速度, 测量能耗的采样频率均相同, 在自由运行模式下约为 4.2kHz。

Q: 我的功率曲线图似乎包含噪声。是我的电路板有缺陷吗?

答: 功率曲线图中显示的功率值取自 (也就是计算得出) 由测量系统计算出的累积电能。当目标消耗很少的电能时, 时间段内只有少量电能包提供给目标, 软件需要累积时间段内的 dc-dc 充电脉冲, 然后再计算新的电流值。对于低于 1μA 的电流, 可能需要最多 1 秒的时间, 而对于 mA 级电流, 可以每毫秒计算一次电流。不会应用其它滤波, 因此不会丢失详细信息。另外一个影响目标消耗的电能 (以及电流) 的因素是正常代码执行期间的周期性后台调试访问 (通过捕获状态信息, 或通过断点轮询)。尝试在自由运行模式中记录, 以查看一个更平滑的功率曲线图。

Q: 我有一个重复调用同样大小函数的代码。我认为函数系统配置显示平等分配运行时间。实际上, 我发现某些函数的运行时间比预期的时间略长, 而某些函数的运行时间稍短。

答: 程序计数器跟踪期间, 多种因素影响一个函数在运行时间内被分析器检测到的次数。此微控制器代码会得益于内部高速缓存, 从而以快于其它函数的速度执行某些函数。另外一个影响因素是内存等待状态和 CPU 流水线停顿, 这在代码执行中增加了时间变化。一个外部因素是调试器本身的采样频率, 此调试器的运行速度通常异步于微控制器代码执行速度, 但是在某些情况下显示出重叠运行方式, 这也导致了不均等的函数运行时间分布。

Q: 我的系统配置有时包含 <未确定> (<Undetermined>) 低功耗模式, 并且状态 (States) 图的功耗模式 (Power Mode) 部分中有缺口。<未确定> (<Undetermined>) 低功耗模式从何而来?

答: 在激活模式向低功耗模式转换期间, 内部器件时钟关闭, 状态信息偶尔不会完全更新。此状态在系统配置 (Profile) 窗口中显示为 <未确定> (<Undetermined>), 状态 (States) 图将在 <未确定> (<Undetermined>) 低功耗模式持续期间显示一个缺口。<未确定> (<Undetermined>) 状态表示应用已进入低功耗模式, 但是无法准确确定是哪个模式。相较于应用很少使用低功耗模式的情况, 如果应用频繁进入低功耗模式, 则 <未确定> (<Undetermined>) 状态将可能会出现得更加频繁。

Q: 在 EnergyTrace 模式下进行捕获时, 即使我的程序相同, 功率和电流的最小值与最大值也会显示偏差。我本来期望是完全一样的值。

答: 硬件上使用的电能测量方法计算运行时间内的直流-直流电荷脉冲。从运行时间内的电能计算出能量和功率。由于统计采样效应和输出电压缓冲电容器的充放电效应, 即使程序是完全一样的, 电流的最小和最大值也会发生几个百分比的变化。不过, 捕获到的电能应该大致相等 (在指定的精度范围内)。

Q: 影响电能测量精度的因素是什么?

答: 电能测量电路直接由 USB 总线电压供电, 因此容易受到 USB 总线电压变化的影响。校准期间, 定义了单个 dc-dc 充电脉冲的电能等效值, 该电能等效值取决于 USB 电压。为了保证良好的可重复性和精度, 应由激活的 USB 端口直接为调试器供电, 并且避免使用总线供电的集线器以及较长的 USB 电缆, 因为它们会导致压降 (特别是将其他耗电设备连接到 USB 集线器时)。而且, 用于生成基准电压的低压降稳压器 (LDO) 和电阻器, 以及那些校准电路中的器件在温度范围内具有特定的耐受值和 ppm 速率, 这些也会影响电能测量的精度。

问: 我试图在 EnergyTrace+ 或 EnergyTrace 模式下, 使用由外部供电的 MSP432 器件捕获数据, 但是系统配置 (Profile)、电能 (Energy)、功率 (Power) 和状态 (States) 窗口中并未显示数据。

答: EnergyTrace+ 模式和 EnergyTrace 模式均要求目标由调试器供电。如果目标微控制器由外部供电, 则无法捕捉任何数据。

问: 我在 EnergyTrace+ 模式下捕获数据时, 无法测量 LPM 电流。我期望有几 μA , 但测量结果超过 150 μA 。

答: 从目标微控制器读取数字数据会消耗微控制器的 JTAG 域中的电能。因此, 将电流表连接到器件电源引脚时, 测得的平均电流大约为 150 μA 。如果要通过调试通信消除能耗, 请切换到 EnergyTrace 模式, 并让目标微控制器在自由运行模式下执行。

Q: 我的 LPM 电流看起来是错的。我期望有几 μA , 但即使在自由运行模式下或让器件在没有独立电源供电的调试控制的情况下执行操作, 测量值也更高。

答: 产生此额外电流的最可能原因是 GPIO 端接错误, 因为悬空引脚可产生额外电流。此外再次检查 JTAG 引脚, 尤其是当调试器仍处于连接状态 (但空闲) 时, 因为调试器在空闲状态下的输出信号电平可能与应用代码配置的 JTAG 引脚情况不匹配。这也可能产生额外电流。

问: 在启动调试会话之前通过视图 (View) → 其他 (Other) → EnergyTrace 启动 EnergyTrace+ 窗口时, 有时无法启动数据捕获。

答: 通过窗口 (Window) → 首选项 (Preferences) → Code Composer Studio → 高级工具 (Advanced Tools) → EnergyTrace™ 技术 (EnergyTrace™ Technology) 启用 EnergyTrace。启动调试会话时, EnergyTrace+ 窗口自动打开, 数据捕获在器件执行时启动。如果在调试会话期间意外关闭所有 EnergyTrace+ 窗口, 可以通过视图 (View) → 其他 (Other) → EnergyTrace 重新打开这些窗口。

8 器件安全

在较新版本的 MSP432 器件型号中，可以为恢复出厂设置命令提供密码保护。如果您已经完成此操作，请跳过以下章节，然后继续“带密码的恢复出厂设置”

8.1 不带密码的恢复出厂设置

如果已禁用了器件上的 JTAG 访问或者正在需要解锁安全 IP 区域的应用上运行，则只能通过已调用循环重启的调试器擦除所有闪存（包括 USER 和 INFO 存储器）来解除锁定。要解锁器件，需要执行以下步骤：

- 选择与当前调试器类型匹配的目标配置
- 执行可触发重启擦除的脚本

下面会详细说明上述步骤。

转到查看 (View) → 目标配置 (Target Configurations) 以查看可用的调试器配置。

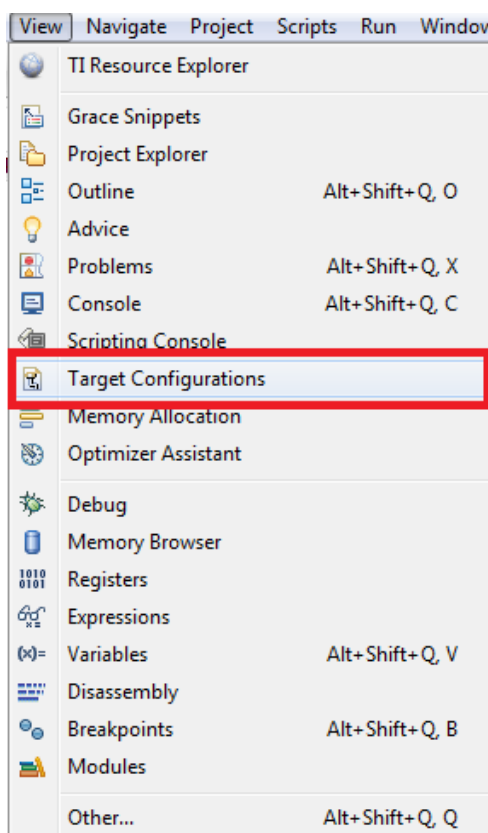


图 30. 显示目标配置视图

Code Composer Studio IDE 会打开一个视图，显示其可以在当前工作区中识别的目标配置。选择适用于该器件和调试器的目标配置。本示例使用了 XDS100v2 调试器的配置文件。

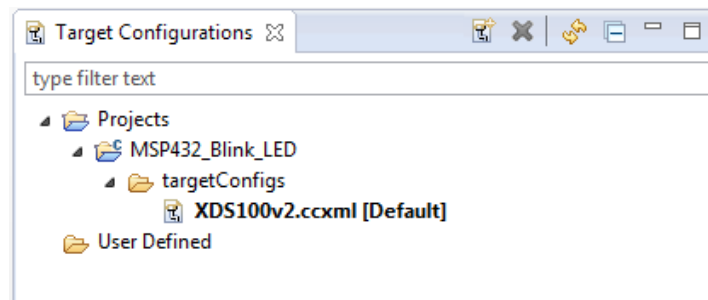


图 31. 目标配置列表

此时，右键单击目标配置，然后选择启动所选配置 (**Launch Selected Configuration**)。

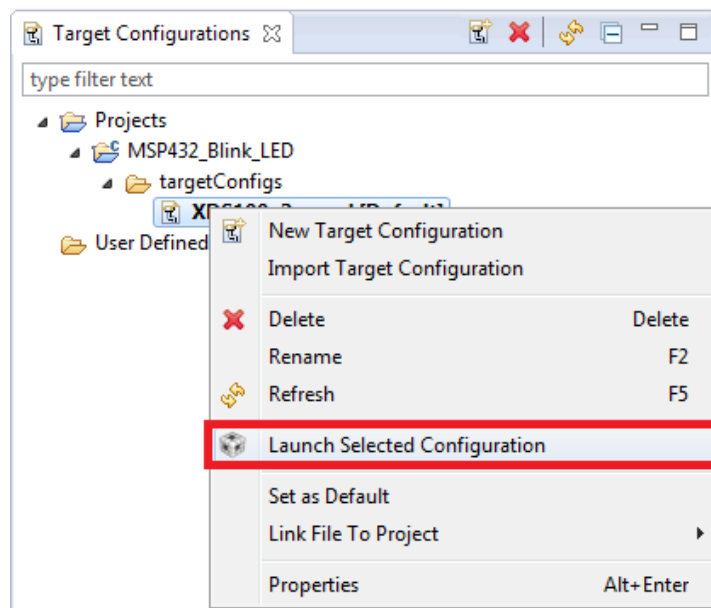


图 32. 启动所选目标配置

此时，调试器与器件相连（仍然有这种可能），但并不会试图暂停 CPU、写入寄存器或下载代码（这是不可能的）。调试 (Debug) 视图显示 CPU 内核，但将其标记为连接断开。

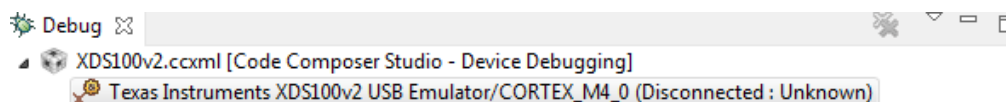


图 33. 启动目标配置后的调试视图

要访问调试器访问端口（即 DAP），请右键单击显示 CPU 内核的高亮行，然后从下拉菜单中选择显示所有内核 (**Show all cores**)。

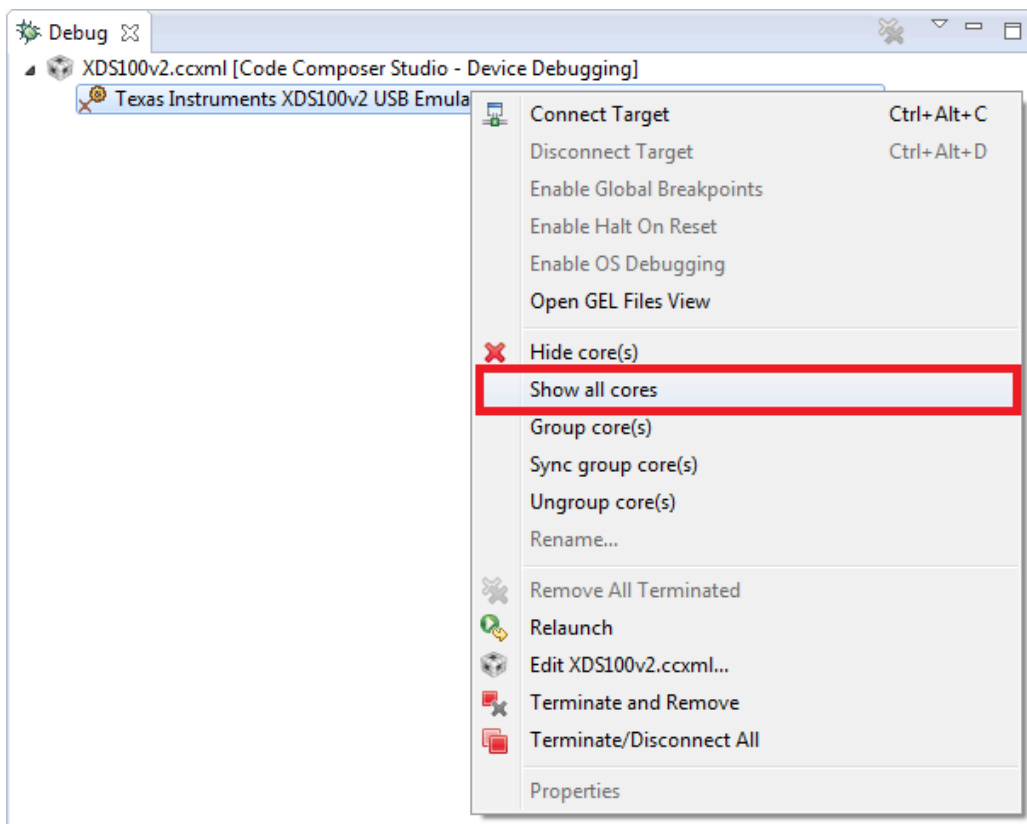


图 34. 显示所有内核 (Show all cores)

此时，MSP432 调试访问端口（即 DAP）列在非可调试器件 (Non Debuggable Devices) 下。

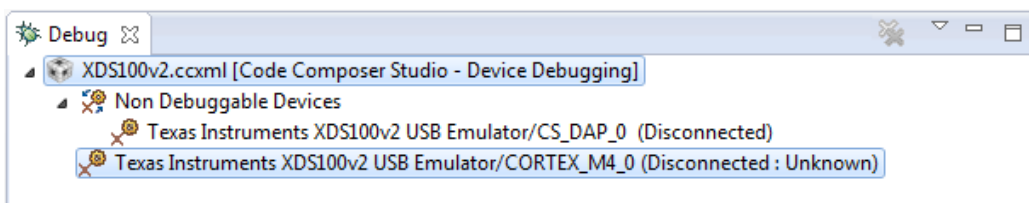


图 35. MSP432 中的所有内核列表

此时，右键单击 DAP，然后选择连接目标 (Connect Target)。

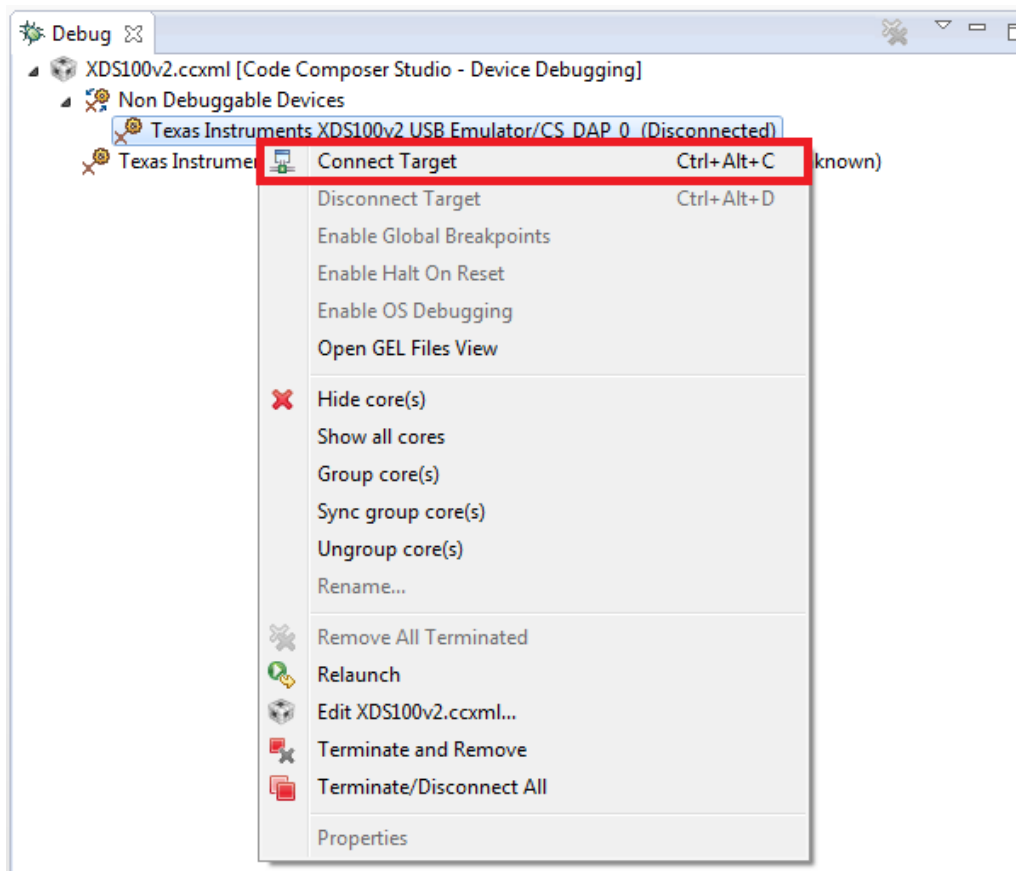


图 36. 手动连接到 DAP

调试器连接到 DAP 后，调试 (Debug) 视图窗口将发生变化，指示已成功连接 DAP。

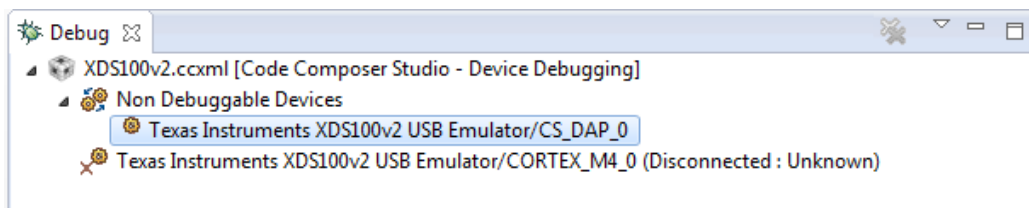


图 37. DAP 已连接

此时，需要执行 Code Composer Studio IDE 脚本，以通过 DAP 触发重启复位。单击脚本 (Scripts) → 默认 (default) → MSP432_Factory_Reset。

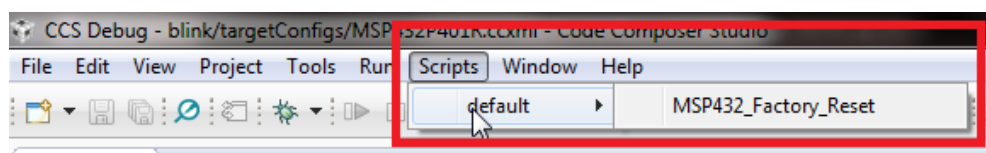


图 38. 执行恢复出厂设置脚本

执行脚本后，控制台 (Console) 窗口显示已执行批量擦除。此时，可以终止调试连接。在重新启动器件后，可以通过正常方式再次进行访问。

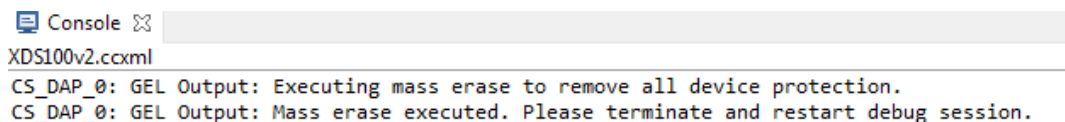


图 39. 批量擦除脚本控制台输出

8.2 带密码的恢复出厂设置

如果已启用了恢复出厂设置的密码保护，并禁用了器件上的 JTAG 访问或正在需要解锁安全 IP 区域的应用上运行，则只能通过已调用循环重启的调试器擦除所有闪存（包括 USER 和 INFO 存储器）来解除锁定。

前提条件：必须已使用闪存邮箱配置带密码的恢复出厂设置，可借助安全性和更新工具（请参见参考文献 [6]）或通过使用应用程序来控制闪存邮箱。

要解锁带密码保护的器件，需要执行以下步骤：

1. 打开 `ccs_base/emulation/gel/msp432_factory_reset_password.gel` 并在用户区域输入密码。
2. 启动 Code Composer Studio IDE 调试会话或目标配置，并连接到器件。
3. 打开工具 (Tools) → GEL 文件 (GEL Files)。

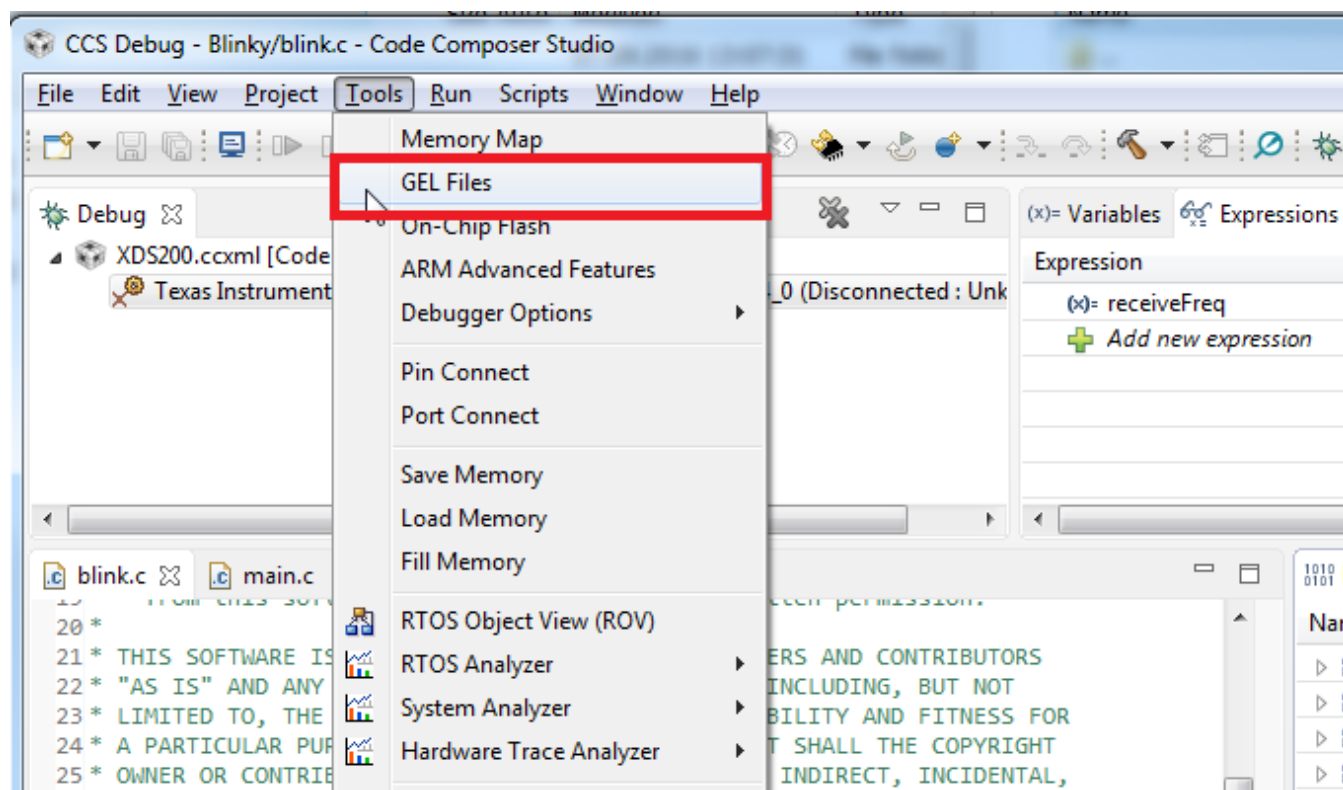


图 40. Code Composer Studio IDE 工具 (Tools) → GEL 文件 (GEL Files)

4. 在新窗口中单击右键，选择“加载 GEL...”(Load GEL...)。
5. 导航至 `msp432_factory_reset_password.gel`，然后单击打开 (Open)。
6. 要运行带密码的恢复出厂设置，请选择脚本 (Scripts) → 默认 (default) → `MSP432_FACTORY_RESET_PASSWORD`

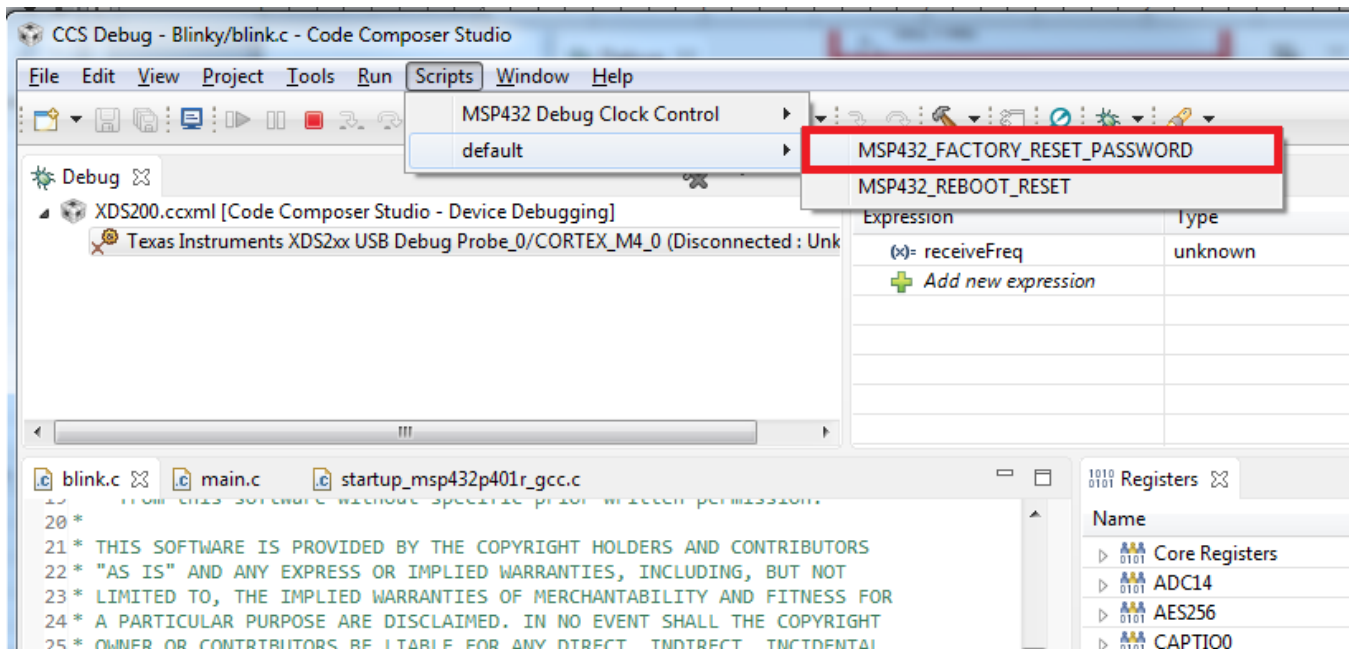


图 41. 带密码的恢复出厂设置 GEL 文件

执行脚本后，控制台 (Console) 窗口显示正在执行批量擦除。擦除完成后，会终止调试连接。重启器件以重新启用对其的访问权限。

注：恢复出厂设置命令的密码为一次性密码。正确发送密码后，将进行恢复出厂设置，此操作会重置“每个”安全设置，其中包括与带密码的恢复出厂设置相关的设置。

GEL 文件可用于 MSP432 器件支持文件 6.3.1.x 及更高版本。

9 低功耗调试

在正常调试控制下，MSP432 微控制器不会转换到比 LPM0 模式程度更深的低功耗模式，即 LPM0_VCORE0 或 LPM0_VCORE1 模式。这种行为由标准化 Cortex-M 调试架构来实现。因此，电流消耗和 IRQ 唤醒时序不同于自由运行的应用程序。

为了能在仍处于基本调试控制状态下时验证应用的电流消耗和时序，MSP432 提供了低功耗运行功能。启用后，MCU 将切换到应用指定的低功耗模式，同时禁用内部时钟并关闭电源管理模块的内部电源域。这意味着：

- 已达到之前设置的断点，但 IDE 不会自动指示器件已暂停。用户必须单击暂停图标启用 IDE，然后才能重新连接并显示程序暂停的位置。
- 加载完程序后自动运行功能失效：已达到 IDE 自动设置的断点 [例如，在 main() 处]，但 IDE 不会自动切换为暂停状态。用户在加载完程序后手动暂停时，程序计数器处于 main() 的起始位置。
- 转换到程度低于 AM0_SL 或 AM1_SL 模式的功耗模式后，SWO 跟踪功能失效

要启用 Code Composer Studio IDE 中的功能，请右键单击项目浏览器中的活动项目，然后单击属性 (Properties)。

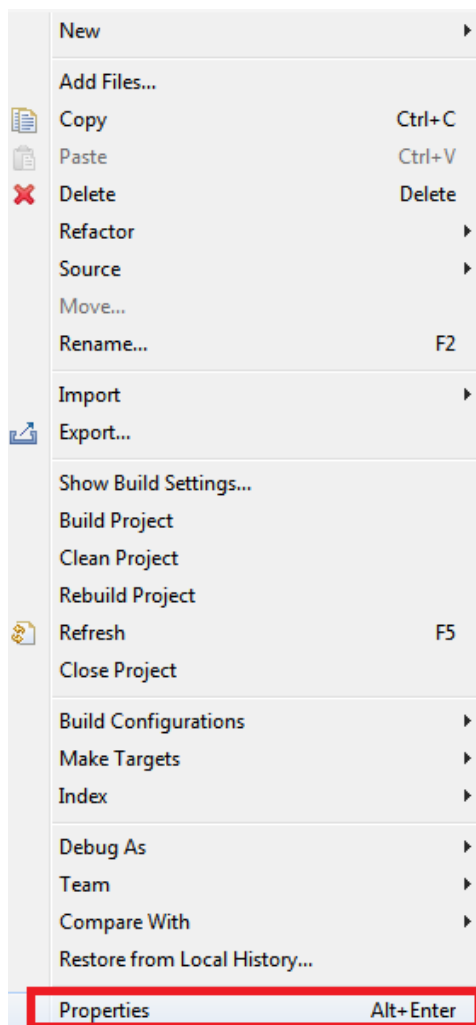


图 42. 属性 (Properties) 菜单

在属性 (Properties) 窗口中，选择调试 (Debug)，然后转到其他/其他选项 (Misc/Other Options)，并启用如果支持（低功耗运行），允许在运行时进行功耗模式转换 (Allow power transitions while running if supported (low power running)) 选项。

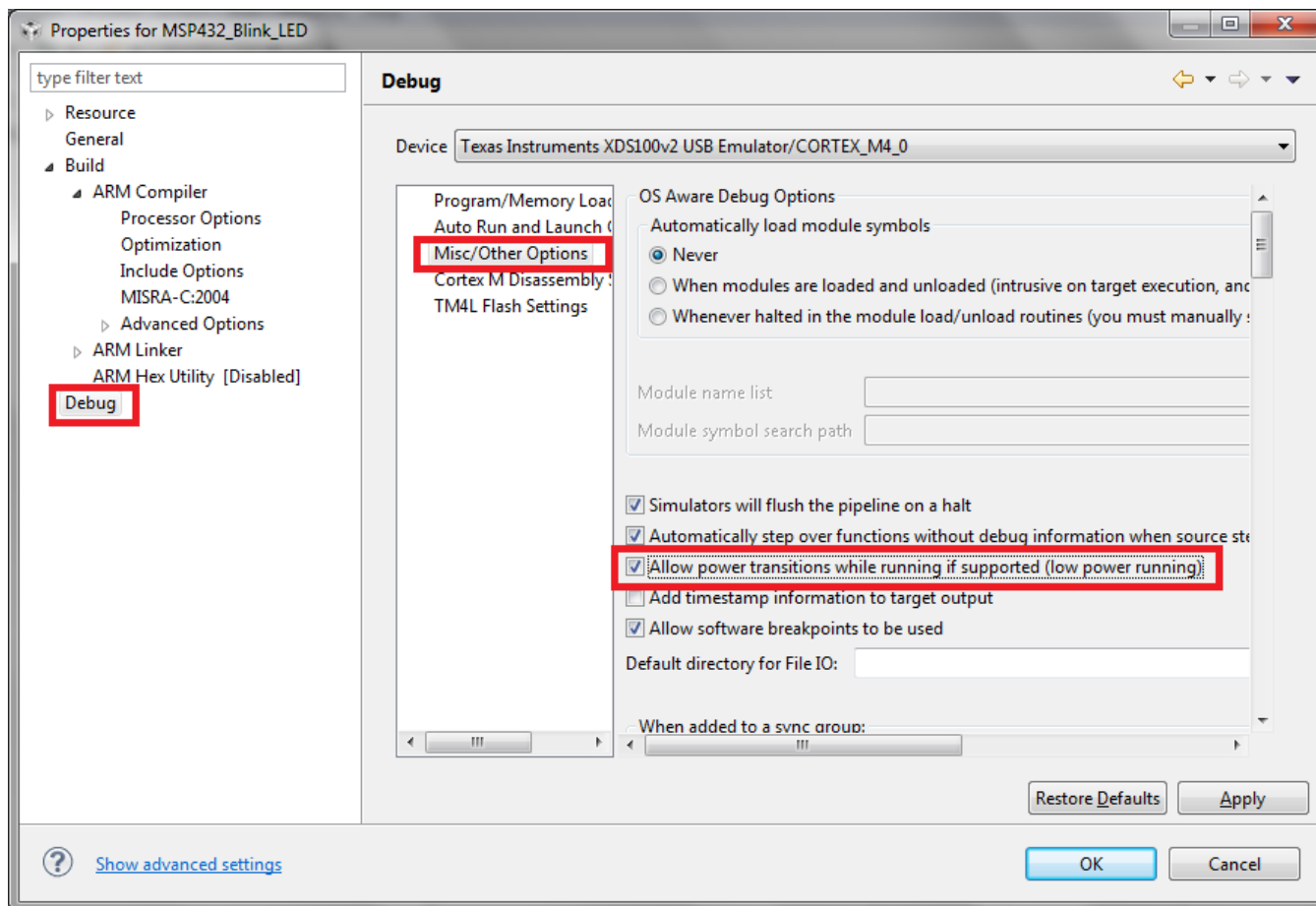


图 43. 启用低功耗运行

启用低功耗运行后，MCU 将进入指定的任意低功耗模式。可以在调试会话期间，通过观察 CPU 内核状态显示来验证效果。由于已禁用所有 MSP432 时钟（包括运行调试访问端口 (DAP) 的时钟），当 MCU 进入程度不低于 DSL 的低功耗模式时，调试器会报告连接丢失。

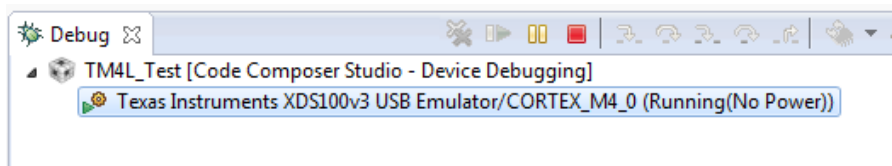


图 44. CPU 内核状态显示指示深度休眠模式

暂停器件后，调试器会重新连接，而且 IDE 会显示当前的程序计数器位置（此时应在使 MSP432 进入休眠状态的 WFI 汇编指令处）。

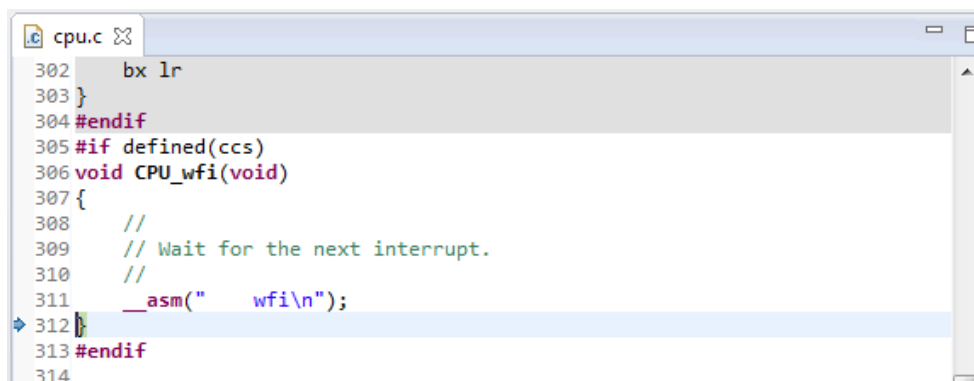


图 45. 程序计数器位于 WFI 指令处

10 常见问题解答

问：我无法对我的 *LaunchPad™* 套件进行编程；IDE 无法连接到目标。这是怎么回事？

答：请检查以下事项：

- JTAG 开关 (S101) 的方向是否正确？
切换到左侧时使用 XDS110-ET 板载调试器
切换到右侧时用于连接外部调试器
- 检查调试器设置：更改为 *SWD Mode (SWD 模式) – Aux COM port is target TDO pin (辅助 COM 端口为目标 TDO 引脚)*。更改端口 J 的设置 (PJSEL0 和 PJSEL1 位) 后，将完全禁止对这些引脚的 JTAG 访问。更改为使用 SWD 后，允许仅通过专用调试引脚进行访问。图 46 显示了如何通过修改 MSP432P401R.ccxml 文件将调试器配置为使用 SWD 而非 JTAG。

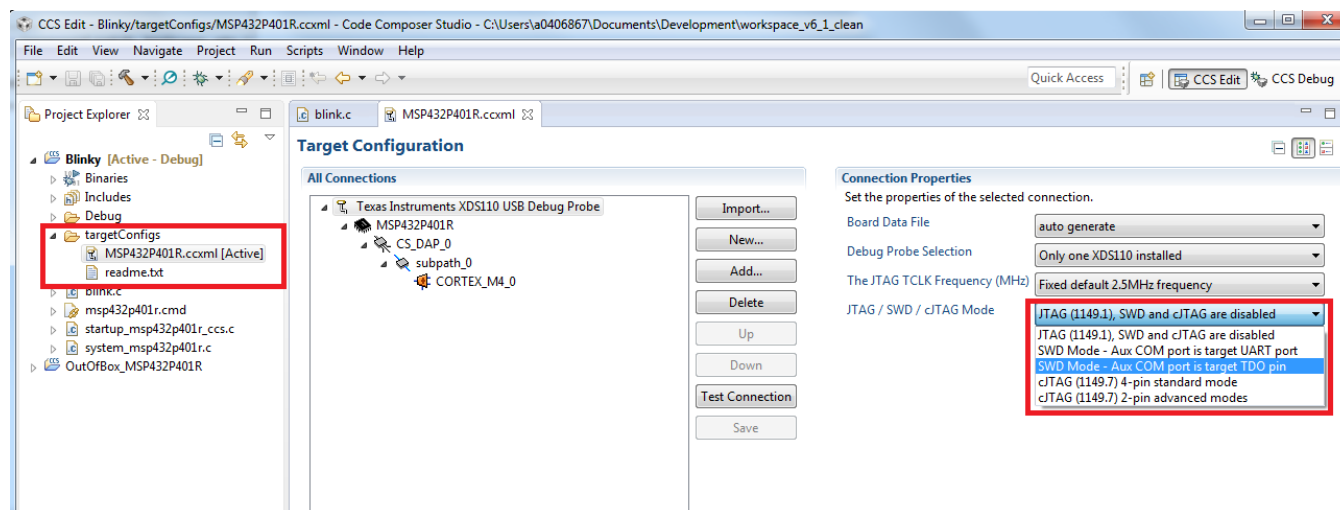


图 46. 将调试器设置更改为 SWD

- 如果仍无法连接，请将器件恢复出厂设置。有关如何在器件上执行恢复出厂设置的信息，请回顾 8 节的内容。

问：为什么 MSP432 LaunchPad 上的反向通道 UART 不能以 56000 波特以上的速度运行串行终端程序？

答：在特定波特率下，MSP432 LaunchPad 可能无法使用某些串行终端程序（如 HTerm 或 Code Composer Studio IDE 内置终端），从而导致软件无法打开虚拟 COM 端口或波特率配置不正确。我们已经发现了 LaunchPad 的仿真器固件方面的问题，会在下一个版本中进行修复。在有可用更新之前，请改用 Tera Term、ClearConnex 或 HyperTerminal，或将波特率降至 38400 波特或更低的速度。

问：将 MSP432 LaunchPad 插入 USB3.0 端口时出现问题

答：据观察，当 MSP432 LaunchPad 连接到 USB3.0 主机控制器硬件和相关设备驱动程序的特定组合提供的 USB3.0 端口时，IDE 无法与 LaunchPad 建立调试会话，同时 Code Composer Studio 会出现错误消息，如“CS_DAP_0: 连接到目标时出错: (Error -260 @ 0x0)，尝试连接到 XDS110 失败”。在这种情况下，Code Composer Studio IDE 提供的低级别命令行实用程序“xdsdfu”也无法与 LaunchPad 建立连接。

具体来说，在运行 Windows 7（设备管理器中显示“瑞萨电子 USB3.0 主机控制器”和相关“瑞萨电子 USB3.0 根集线器”）的 PC 上发现了此问题。将关联的 Windows USB 驱动程序更新为从硬件供应商处获取的较新版本后，问题就会得到解决。可能还有其他 USB3.0 硬件和设备驱动程序组合会导致同样问题。如果您认为可能会受到影响，请联系 PC 供应商，或者找到更新版本的 USB3.0 设备驱动程序并进行安装。或者，将 LaunchPad 连接到 PC 上的 USB2.0 端口（如果可用）。

问：我无法连接反向通道 UART。这是怎么回事？

答：请检查以下事项：

- 主机的终端应用和 eUSCI 设置中的波特率是否匹配？
- 隔离跳线块上的相关跳线是否到位？
- 探测 RXD 并从主机发送数据。如果您没有看到数据，则可能是主机端的问题。
- 探测 TXD，同时从 MSP432 发送数据。如果您没有看到数据，则可能是 eUSCI 模块的配置问题。
- 可考虑使用硬件流控制线（尤其是波特率较高时）。

问：如何快捷解锁 SYS_CTL 寄存器块？

答：对于 TI ARM 编译器，有一个宏定义可用于实现这一目的。请参见相应器件头文件中的 `#define UNLOCK_DEVICE`。

11 有关 Code Composer Studio IDE 的附加信息

有关 Code Composer Studio IDE 的更多信息，请参见以下链接：

- [Code Composer Studio 信息](#)
- [Code Composer Studio v7 培训](#)
- [Code Composer Studio v7 Wiki](#)

12 参考文献

1. [SimpleLink MSP432 SDK](#)
2. [J-Link 仿真器支持](#)
3. [MSP432™ 调试工具：使用 CCS 跟踪分析器实现的串行线输出](#)
4. [适用于 MSP432 微控制器的 MSP-FET](#)
5. [SimpleLink MSP432 安全性和更新工具](#)
6. [XDS110 JTAG 调试探针](#)
7. [适用于 MSP432 微控制器的调试器](#)
8. [《SimpleLink MSP432 SDK 迁移指南》](#)

修订历史记录

注：之前版本的页码可能与当前版本有所不同。

Changes from September 10, 2016 to March 6, 2017	Page
• 更改了本用户指南的标题	1
• 将整篇文档中的 Code Composer Studio 版本由 6.1 更改为 7.1	1
• 根据需要添加了“SimpleLink”品牌并更新了所引用文档的标题	1
• 在1 节，安装 Code Composer Studio IDE 中增加了 MSP432 对于 Code Composer Studio IDE 版本的要求以及 SimpleLink SDK 支持	4
• 删除了之前的图 2，可用更新	4
• 在3 节，创建基本的 SimpleLink MSP432 项目中增加了开头为“可以创建一个全新的项目...”的段落	5
• 更新了图 3，新建项目向导	6
• 更新了图 4，新建项目文件	6
• 替换了之前的第 4 节，其他 MSP432 示例和文档，替换为4 节，从 TI 资源浏览器导入 SimpleLink 示例	7
• 在5.2 节，调试 ROM 驱动程序库的开头增加了三段内容	12
• 更新了12 节，参考文献	39

有关 TI 设计信息和资源的重要通知

德州仪器 (TI) 公司提供的技术、应用或其他设计建议、服务或信息，包括但不限于与评估模块有关的参考设计和材料（总称“TI 资源”），旨在帮助设计人员开发整合了 TI 产品的应用；如果您（个人，或如果是代表贵公司，则为贵公司）以任何方式下载、访问或使用了任何特定的 TI 资源，即表示贵方同意仅为该等目标，按照本通知的条款进行使用。

TI 所提供的 TI 资源，并未扩大或以其他方式修改 TI 对 TI 产品的公开适用的质保及质保免责声明；也未导致 TI 承担任何额外的义务或责任。TI 有权对其 TI 资源进行纠正、增强、改进和其他修改。

您理解并同意，在设计应用时应自行实施独立的分析、评价和判断，且应全权负责并确保应用的安全性，以及您的应用（包括应用中使用的 TI 产品）应符合所有适用的法律法规及其他相关要求。您就您的应用声明，您具备制订和实施下列保障措施所需的一切必要专业知识，能够 (1) 预见故障的危险后果，(2) 监视故障及其后果，以及 (3) 降低可能导致危险的故障几率并采取适当措施。您同意，在使用或分发包含 TI 产品的任何应用前，您将彻底测试该等应用和该等应用所用 TI 产品的功能。除特定 TI 资源的公开文档中明确列出的测试外，TI 未进行任何其他测试。

您只有在为开发包含该等 TI 资源所列 TI 产品的应用时，才被授权使用、复制和修改任何相关单项 TI 资源。但并未依据禁止反言原则或其他法理授予您任何 TI 知识产权的任何其他明示或默示的许可，也未授予您 TI 或第三方的任何技术或知识产权的许可，该等产权包括但不限于任何专利权、版权、屏蔽作品权或与使用 TI 产品或服务的任何整合、机器制作、流程相关的其他知识产权。涉及或参考了第三方产品或服务的信息不构成使用此类产品或服务的许可或与其相关的保证或认可。使用 TI 资源可能需要您向第三方获得对该等第三方专利或其他知识产权的许可。

TI 资源系“按原样”提供。TI 兹免除对 TI 资源及其使用作出所有其他明确或默示的保证或陈述，包括但不限于对准确性或完整性、产权保证、无屡发故障保证，以及适销性、适合特定用途和不侵犯任何第三方知识产权的任何默认保证。

TI 不负责任何申索，包括但不限于因组合产品所致或与之有关的申索，也不为您辩护或赔偿，即使该等产品组合已列于 TI 资源或其他地方。对因 TI 资源或其使用引起或与之有关的任何实际的、直接的、特殊的、附带的、间接的、惩罚性的、偶发的、从属或惩戒性损害赔偿，不管 TI 是否获悉可能会产生上述损害赔偿，TI 概不负责。

您同意向 TI 及其代表全额赔偿因您不遵守本通知条款和条件而引起的任何损害、费用、损失和/或责任。

本通知适用于 TI 资源。另有其他条款适用于某些类型的材料、TI 产品和服务的使用和采购。这些条款包括但不限于适用于 TI 的半导体产品 (<http://www.ti.com/sc/docs/stdterms.htm>)、[评估模块](http://www.ti.com/sc/docs/stdterms.htm)和样品 (<http://www.ti.com/sc/docs/sampters.htm>) 的标准条款。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122
Copyright © 2017 德州仪器半导体技术（上海）有限公司