

在 Kinetis 系列产品上使用低功耗模式

作者: Daniel Martinez 和 César Manzanarez
Guadalajara
México

1 简介

器件功耗以及低功耗设计的意义是目前常见的话题。

Freescale 提供节能产品解决方案，能助您满足应用的功耗要求。

本应用笔记将指导您实现 Kinetis 系列处理器可用的功耗模式。

需要注意的是，本应用笔记基于 Kinetis 50 MHz 器件，因此一些低功耗模式并不是在所有的 Kinetis 系列器件中都是可用的；相关章节对此会有所说明。

2 低功耗概述和最佳实践

有很多理由可以解释为什么嵌入式系统设计人员不断努力，以满足严格的功耗预算要求。

一个原因是成本。如果便携式设备的电池续航时间不理想，那么当面临能效更高的竞争产品时将会失去市场。

对于其他一些应用来说，成本并非最主要的因素——比如植入病患体内的医疗设备。这类设备需依靠电池工作多年，更换电池意味着对病人动手术，因而在该情形下高效功耗解决方案尤为重要。

低功耗设计应当在功耗和性能之间进行权衡，并尽量利用器件提供的每个可能特性来实现最佳结果。

内容

1	简介.....	1
2	低功耗概述和最佳实践.....	1
3	系统时钟生成.....	2
4	低功耗模式说明和使用方法.....	3
5	低功耗模式下的模块操作.....	11
6	支持的功耗模式转换.....	13
7	进入功耗模式的演示.....	14
8	结语.....	31
9	参考.....	31

软件和硬件设计人员都应参与能效设计，并在计划的早期阶段估算产品的功耗需求，避免随后的重新设计。

低功耗设计的提示与技巧：

- 将引脚设为已知状态：
 - 对于 Kinetis 系列，建议将未使用的引脚保持在浮空状态，并配置为禁用。
 - 选择并配置所需的时钟模式。记住，更高的内核频率意味着更高的功耗。
 - 较高的工作温度会增加功耗。
 - 禁用全部未使用模块的时钟。
 - 时钟监测器输出和调试使能后，会增加功耗，必须避免。

在以下各章中，我们将详细研究微控制器工作涉及的每一个模块，及其与 Kinetis 系列提供的低功耗模式的相互关系。

3 系统时钟生成

Kinetis 系列集成有多用途时钟发生器(MCG)模块，可通过外部（系统振荡器、RTC 振荡器）或内部时钟源生成系统时钟。在下图中，我们可以看出这些模块之间的依赖关系。

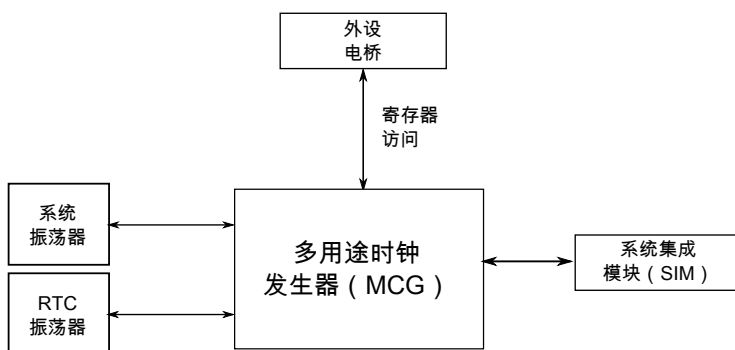


图 1. MCG 配置

系统振荡器和 RTC 振荡器是可以用来输入 MCG 的外部时钟源。

OSC 模块是一个晶体振荡器。该模块与外部晶体或谐振器一同生成 MCU 的参考时钟。

RTC 振荡器模块为 RTC 提供时钟源。RTC 振荡器模块与外部晶体一同生成 RTC 参考时钟。

此外，还有可以在如下两个内部时钟之间任选其一，作为 MCG 的输入时钟源：

- 慢速 IRC (32 kHz)和快速 IRC (4 MHz)

这些内部时钟属于 MCG 模块的一部分。

MCG 模块控制使用哪个时钟源来获得系统时钟。时钟生成逻辑将选定的时钟源分为不同的时钟域，包括用于系统总线主机、系统总线从机和闪存的时钟。

总而言之，共有 4 个输入时钟源可用作时钟发生器以供微控制器使用，应当根据您的应用需求来选择时钟源。

模块依赖关系：

- 所有模块在其一个或多个时钟上依赖 MCG。
- 振荡器模块在控制信号上依赖 MCG。
- 使用振荡器前，必须将其从 MCG 内部正确设置。
- 提升 MCGOUT 频率前必须设置系统时钟分频器，确保不超过最大时钟速率。
- MCG 在除超低漏电流停止模式外的所有低功耗模式下均通电。
- 退出复位后，晶振引脚默认为 XTAL 和 EXTAL。
- MCG 模块自身没有相关的时钟选通信号。

MCG 共有 9 种不同的操作模式：FEI、FEE、FBI、FBE、PBE、PEE、BLPI、BLPE 和 STOP 模式。

下一章介绍不同低功耗模式下系统时钟设置的重要详情。

有关各模块的详细信息，请参考器件参考手册。

3.1 低功耗模式下的系统时钟功能

- STOP - 正常停止模式
 - 可从任意 MCG 模式进入 STOP 模式
 - 可选择性地保持 PLL 使能，但是输出会被关闭 - C5[PLLSTEN]
 - 可选择性地保持内部参考时钟使能 - C1[IREFSTEN]
 - 可选择性地保持外部参考时钟使能 - OSC_CR[EREFSTEN]
 - 进入 STOP 后，将在相同的 MCG 模式下退出 STOP 模式。
 - 例外情况是在 PEE 模式下，PLLSTEN=0 时进入 STOP；这种情况下，退出 STOP 时，MCG 将处于 PBE 模式
- VLPR - 超低功耗运行模式
 - 选择快速 IRC 可从 BLPI 模式进入 VLPR。
 - 如果外部时钟最大值为 4 MHz，那么可从 BLPE 模式进入 VLPR。
 - 进入 VLPR 后 (BLPI 或 BLPE)，将在同样的 MCG 模式下退出 VLPR
- VLPW - 超低功耗等待模式
 - 选择快速 IRC 则可从 BLPI 模式进入 VLPW。
 - 如果外部时钟最大值为 4 MHz，那么可从 BLPE 模式进入 VLPW。
 - 进入 VLPW 后 (BLPI 或 BLPE)，将在同样的 MCG 模式下退出 VLPW
 - 可选择性地保持外部参考时钟使能 (最大 4 MHz)
- VLPS - 超低功耗停止
 - 可从任意 MCG 模式进入 VLPS 模式
 - 可选择性地保持外部参考时钟使能 (最大 4 MHz)
 - MCG 为静态，无时钟激活 (IREFSTEN 和 PLLSTEN 无影响)
 - 进入 VLPS 后，将在相同的 MCG 模式下退出 VLPS 模式。
 - 例外情况是在 PEE 模式下进入 VLPS；这种情况下，退出 VLPS 时，MCG 将处于 PBE 模式
- LLS - 低漏电流停止
 - 可从任意 MCG 模式进入 LLS 模式
 - 在低范围和低功耗振荡器模式 (32 kHz) 下，可选择性保持外部参考时钟使能
 - MCG 为静态，无时钟激活 (IREFSTEN 和 PLLSTEN 无影响)
 - 进入 LLS 后，将在相同的 MCG 模式下退出 LLS 模式。
 - 例外情况是在 PEE 模式下进入 LLS；这种情况下，退出 LLS 时，MCG 将处于 PBE 模式
- VLLS0/1/2/3 - 超低漏电流停止
 - 可从任意 MCG 模式进入 VLLSx 模式
 - MCG 关闭，无时钟激活，寄存器状态不保持
 - 将在 FEI 模式 (复位状态) 下退出
 - 在低范围和低功耗振荡器模式 (32 kHz) 下，可选择性保持外部参考时钟使能。清零 ACKISO 位前，必须重新配置振荡器。

4 低功耗模式说明和使用方法

本章介绍的工作模式适用于 Kinetis 系列器件。

提供下列工作模式：

- 运行
- 等待
- 停止
- VLPR (超低功耗运行)
- VLPW (超低功耗等待)

- VLPS (超低功耗停止)
- LLS (低漏电流停止)
- VLLS3 (超低漏电流停止 3)
- VLLS2 (超低漏电流停止 2)
- VLLS1 (超低漏电流停止 1)
- VLLS0 (超低漏电流停止 0) ¹
- BAT (仅备用电池)

本章将深入探讨本应用笔记相关的所有工作模式。也就是说，本章讨论除运行和 BAT 外的所有模式。

注

有关工作模式的完整说明，请参考器件参考手册，并搜索“功耗模式”一章。

4.1 运行模式

该模式是复位后默认的模式。

允许芯片实现最佳性能。

工作模式详情:

- 任意复位后选定
- 片内电压调节器开启，全功能
- 堆栈指针(SP)，程序计数器(PC)和链接寄存器均置位
- ARM 处理器退出复位并读取起始 SP
- ARM 处理器读取向量表中的起始 PC
- 清零 SCGCx 中的时钟门控位以降低功耗

4.2 等待模式

内核处于睡眠模式时允许外设运行以降低功耗。

NVIC 仍然检测中断；继续提供外设时钟。

工作模式详情:

- ARM 内核进入睡眠模式
- ARM 内核时钟关闭 (HCLK 关闭)
- NVIC 仍然对中断敏感 (FCLK 开启)
- 继续为外设提供时钟
- 清零 SCGCx 中的时钟门控位以降低功耗
- 中断时，ARM 内核退出睡眠模式：恢复处理

表 1. 如何进入等待模式

从	到	触发条件
运行	等待	执行 WAIT(); 这意味着 SLEEPDEEP 清零情况下会进入“立即睡眠”或退出时“睡眠模式”
等待	运行	中断或复位

1. 仅 50 MHz 器件提供此工作模式。该器件提供两种工作模式：上电复位检测电路使能或禁用。

4.3 停止模式

将芯片置于静态。是既能保持所有寄存器，同时还能维持 LVD 保护的最低功耗模式。NVIC 禁用；AWIC 用于从中断唤醒；外设时钟停止。

工作模式详情：

- ARM 内核进入深度睡眠模式
- ARM 内核时钟关闭 (HCLK 关闭)
- NVIC 禁用 (FCLK 关闭)
- WIC 用于从中断唤醒
- 平台和外设时钟停摆
- MCG 模块可配置用于使参考时钟保持运行状态
- 所有 SRAM 均处于工作状态 (保留内容, 保持 I/O 状态)

表 2. 如何进入停止模式

从	到	触发条件
运行	停止	执行 STOP(); 这意味着 SLEEPDEEP 置位情况下会进入立即睡眠或退出时睡眠模式
停止	运行	中断或复位 - 中断进入 ISR (无 LLWU)

4.4 VLPR (超低功耗运行)

片内电压调节器处于低功耗模式，仅提供芯片低频运行所需的电压。降频 Flash 访问模式 (1 MHz); LVD 关闭; 内部振荡器为内核、总线以及外设时钟提供低功耗 4 MHz 源。

工作模式详情：

- 片内电压调节器处于仅提供 MCU 低频运行所需电压的模式
- 内核和总线时钟限定为 4 Mhz
- Flash 频率限定为 1 Mhz
- 清零 SCGCx 中的时钟门控位以降低功耗
- 不允许 Flash 编程和擦除操作
- 不允许 FlexMemory (EEPROM)编程操作
- 时钟监视器
 - 如果 MCG 处于 BLPE 模式，那么在进入 VLPR 或 VLPW 前，CME0 和 CME1 也应设为逻辑 0。

表 3. 如何进入 VLPR 模式

从	到	触发条件
运行	VLPR	将系统总线和内核频率降至 2 MHz 或更低 Flash 访问频率限制为 1 MHz, AVLP = 1, 设置 RUNM = 10 注: 执行 VLPR 特定代码前轮询 VLPRS 位 (可以等待大约 5 μs, 而非等待 VLPRS)
VLPR	运行	LPWUI=1 时中断, 或复位

表 3. 如何进入 VLPR 模式

从	到	触发条件
		注：先轮询 REGONS 位再提高频率。

4.5 VLPW（超低功耗等待）

与 VLPR 相同，但内核处于睡眠模式下，以便进一步降低功耗；NVIC 仍然对中断敏感（FCLK 开启）。片内电压调节器处于低功耗模式，仅提供芯片低频运行所需的电压。

工作模式详情：

- ARM 内核进入睡眠模式
- ARM 内核时钟关闭（HCLK 关闭）
- NVIC 仍然对中断敏感（FCLK 开启）
- 片内电压调节器处于仅提供 MCU 低频运行所需电压的模式
- 系统和总线时钟限制为 4Mhz
- 清零 SCGCx 中的时钟门控位以降低功耗
- 时钟监视器：
 - 如果 MCG 处于 BLPE 模式，那么在进入 VLPR 或 VLPW 前，CME0 和 CME1 也应设为逻辑 0。

表 4. 如何进入 VLPW 模式

从	到	触发条件
VLPR	VLPW	执行 WAIT()；
VLPW	VLPR	LPWUI = 0 时中断
VLPW	RUN	LPWUI=1 时中断，或复位

4.6 VLPS（超低功耗停止）

将芯片置于静态，关闭 LVD 操作。ADC 和引脚中断可以工作的最低功耗模式。外设时钟停摆，但 LPTimer、RTC、CMP、TSI 可以使用。NVIC 禁用（FCLK 关闭），AWIC 用来从中断唤醒。

片内电压调节器处于低功耗模式，仅提供芯片低频运行所需的电压。所有 SRAM 均处于工作状态（保留内容，保持 I/O 状态）。

工作模式详情：

- ARM 内核进入深度睡眠模式
- ARM 内核时钟关闭（HCLK 关闭）
- NVIC 禁用（FCLK 关闭）
- WIC 用于从中断唤醒
- 平台和外设时钟停摆
- MCG 模块可配置用于使参考时钟保持运行状态

- 片内电压调节器处于仅提供 MCU 低频运行所需电压的模式
- 所有 SRAM 均处于工作状态（保留内容，保持 I/O 状态）

表 5. 如何进入 VLPS 模式

从	到	触发条件
VLPR	VLPS	STOPM = 000 或 010, 执行 STOP();
VLPS	VLPR	LPWUI = 0 时中断
RUN	VLPS	AVLP=1, STOPM =010, 执行 STOP();
VLPS	RUN	LPWUI=1 时中断, 或复位

4.7 LLS（低漏电流停止）

状态保留功耗模式。大多数外设处于状态保留模式（时钟停摆），但 LLWU、LPTimer、RTC、CMP、TSI 可以使用。NVIC 禁用；LLWU 用来唤醒。

注

LLWU 中断不可通过中断控制器屏蔽，以避免系统无法在 LLS 恢复时完全退出停止模式的现象。

所有 SRAM 均处于工作状态（保留内容，保持 I/O 状态）。

工作模式详情：

- ARM 内核进入深度睡眠模式
- ARM 内核时钟关闭（HCLK 关闭）
- NVIC 禁用（FCLK 关闭）
- LLWU 用于从中断唤醒
- 平台和外设时钟停摆
- MCG 模块可配置用于使参考时钟保持运行状态
- 片内电压调节器处于仅提供 MCU 低频运行所需电压的模式
- 所有 SRAM 均处于工作状态（保留内容，保持 I/O 状态）
- 大多数外设处于状态保留模式（无法工作）

表 6. 如何进入 LLS 模式

从	到	触发条件
运行	LLS	设置 PMPROT 中的 ALLS, PMCTRL_STOPM = 011, 执行 STOP();
LLS	运行	从已使能的 LLWU 引脚或模块源唤醒, 或者复位引脚。
VLPR	LLS	设置 PMPROT 中的 ALLS, PMCTRL_STOPM = 011, 执行 STOP();

4.8 VLLS3 (极低漏电流停止 3)

大多数外设 (时钟停摆) 禁用, 但 LLWU、LPTimer、RTC、CMP、TSI 可以使用。NVIC 禁用; LLWU 用来唤醒。SRAM_U 和 SRAM_L 保持上电状态 (保留内容, 保持 I/O 状态)。

工作模式详情:

- ARM 内核进入深度睡眠模式
- ARM 内核时钟关闭 (HCLK 关闭)
- NVIC 禁用 (FCLK 关闭)
- LLWU 应当由用户来配置, 以使能所需的唤醒源
- 平台和外设时钟停摆
- MCG 模块可配置用于使参考时钟保持运行状态
- 片内电压调节器处于仅提供 MCU 低频运行所需电压的模式
- 所有 SRAM 均处于工作状态 (保留内容, 保持 I/O 状态)
- 大多数模块禁用

表 7. 如何进入 VLLS3 模式

从	到	触发条件
运行	VLLS (3、2、1、0)	设置 PMPROT 中的 AVLLSx、PMCTRL_STOPM = 100 VLLSM = 011 (VLLS3)、010 (VLLS2)、001 (VLLS1)、000 (VLLS0) 注: 对于 VLLS0, 必须置位或清零 PORPO, 以便分别禁用或使能 POR 检测电路。执行 STOP();
VLLS (3、2、1、0)	运行	从已使能的 LLWU 输入源唤醒或复位。所有唤醒都会执行复位序列。检查 SRS 以确定唤醒源。检查 VLLSM 以确定工作模式。
VLPR	VLLS (3、2、1、0)	设置 PMPROT 中的 AVLLSx、PMCTRL_STOPM = 100 VLLSM = 011 (VLLS3)、010 (VLLS2)、001 (VLLS1)、000 (VLLS0) 注: 对于 VLLS0, 必须置位或清零 PORPO, 以便分别禁用或使能 POR 检测电路。执行 STOP();

4.9 VLLS2 (极低漏电流停止 2)

大多数外设 (时钟停摆) 禁用, 但 LLWU、LPTimer、RTC、CMP、TSI 可以使用。NVIC 禁用; LLWU 用来唤醒。SRAM_L 断电。SRAM_U 的一部分保持上电状态 (保留内容, 保持 I/O 状态)。

工作模式详情:

- ARM 内核进入深度睡眠模式
- ARM 内核时钟关闭 (HCLK 关闭)
- NVIC 禁用 (FCLK 关闭)
- LLWU 应当由用户来配置, 以使能所需的唤醒源
- 平台和外设时钟停摆
- MCG 模块可配置用于使参考时钟保持运行状态
- 片内电压调节器处于仅提供 MCU 低频运行所需电压的模式

- 仅一部分 SRAM 处于工作状态（保留内容，保持 I/O 状态）
- 大多数模块禁用

表 8. 如何进入 VLLS2 模式

运行	VLLS (3、2、1、0)	设置 PMPROT 中的 AVLLSx、 PMCTRL_STOPM = 100 VLLSM = 011 (VLLS3)、010 (VLLS2)、001 (VLLS1)、 000 (VLLS0) 注：对于 VLLS0，必须置位或清零 PORPO，以便分别禁用或使能 POR 检测电路。执行 STOP()；
VLLS (3、2、1、0)	运行	从已使能的 LLWU 输入源唤醒或复位。 所有唤醒都会执行复位序列。检查 SRS 以确定唤醒源。检查 VLLSM 以确定工作 模式。
VLPR	VLLS (3、2、1、0)	设置 PMPROT 中的 AVLLSx、 PMCTRL_STOPM = 100 VLLSM = 011 (VLLS3)、010 (VLLS2)、001 (VLLS1)、 000 (VLLS0) 注：对于 VLLS0，必须置位或清零 PORPO，以便分别禁用或使能 POR 检测电路。执行 STOP()；

4.10 VLLS1（极低漏电流停止 1）

大多数外设（时钟停摆）禁用，但 LLWU、LPTimer、RTC、CMP、TSI 可以使用。NVIC 禁用；LLWU 用来唤醒。

所有 SRAM_U 和 SRAM_L 均断电。32 字节系统寄存器文件和 32 字节 VBAT 寄存器文件保持上电，以便保存客户关键型数据。

工作模式详情：

- ARM 内核进入深度睡眠模式
- ARM 内核时钟关闭 (HCLK 关闭)
- NVIC 禁用 (FCLK 关闭)
- LLWU 应当由用户来配置，以使能所需的唤醒源
- 平台和外设时钟停摆
- MCG 模块可配置用于使参考时钟保持运行状态
- 片内电压调节器处于仅提供 MCU 低频运行所需电压的模式
- 所有 SRAM 均掉电，仅保留 32 字节寄存器内容，并保持 I/O 状态
- 大多数模块禁用

从	到	触发条件
运行	VLLS (3、2、1、0)	设置 PMPROT 中的 AVLLSx、 PMCTRL_STOPM = 100 VLLSM = 011 (VLLS3)、010 (VLLS2)、001 (VLLS1)、 000 (VLLS0) 注：对于 VLLS0，必须置位或清零 PORPO，以便分别禁用或使能 POR 检测电路。执行 STOP()；

下一页继续介绍此表...

从	到	触发条件
VLLS (3、2、1、0)	运行	从已使能的 LLWU 输入源唤醒或复位。所有唤醒都会执行复位序列。检查 SRS 以确定唤醒源。检查 VLLSM 以确定工作模式。
VLPR	VLLS (3、2、1、0)	设置 PMPROT 中的 AVLLSx、PMCTRL_STOPM = 100 VLLSM = 011 (VLLS3)、010 (VLLS2)、001 (VLLS1)、000 (VLLS0) 注：对于 VLLS0，必须置位或清零 PORPO，以便分别禁用或使能 POR 检测电路。执行 STOP();

4.11 VLLS0 (极低漏电流停止 0)

LLWU、RTC。所有 SRAM 均断电。

工作模式详情：

- ARM 内核进入深度睡眠模式
- ARM 内核时钟关闭 (HCLK 关闭)
- NVIC 禁用 (FCLK 关闭)
- LLWU 应当由用户来配置，以使能所需的唤醒源
- 平台和外设时钟停摆
- MCG 模块可配置用于使参考时钟保持运行状态
- 片内电压调节器处于仅提供 MCU 低频运行所需电压的模式
- 所有 SRAM 均断电。
- 除 LLWU 和 RTC 外的大多数模块都禁用

表 10. 如何进入 VLLS0 模式

从	到	触发条件
运行	VLLS (3、2、1、0)	设置 PMPROT 中的 AVLLSx、PMCTRL_STOPM = 100 VLLSM = 011 (VLLS3)、010 (VLLS2)、001 (VLLS1)、000 (VLLS0) 注：对于 VLLS0，必须置位或清零 PORPO，以便分别禁用或使能 POR 检测电路。执行 STOP();
VLLS (3、2、1、0)	运行	从已使能的 LLWU 输入源唤醒或复位。所有唤醒都会执行复位序列。检查 SRS 以确定唤醒源。检查 VLLSM 以确定工作模式。
VLPR	VLLS (3、2、1、0)	设置 PMPROT 中的 AVLLSx、PMCTRL_STOPM = 100 VLLSM = 011 (VLLS3)、010 (VLLS2)、001 (VLLS1)、000 (VLLS0) 注：对于 VLLS0，必须置位或清零 PORPO，以便分别禁用或使能 POR 检测电路。执行 STOP();

5 低功耗模式下的模块操作

本章中，我们将看到各种不同功耗模式下的微控制器模块状态。

为了选择最适合应用的低功耗模式，必须首先列出需要使能的模块。

下表很有用，根据需要使能的模块可确定应用应当进入何种低功耗模式。

表中所使用术语的含义：

- FF: 全部功能
- 静态: 保留模块寄存器状态和相关存储器。
- 通电: 存储器通电以保留内容。
- 低功耗: Flash 具有一个低功耗状态，能够保留配置寄存器以支持更加快速地唤醒。
- 关: 模块断电；唤醒后模块处于复位状态。
- 唤醒: 模块可用作芯片的唤醒源。

更多详情，请参考器件参考手册中的“低功耗模式下的模块操作”一章。

表 11. 低功耗模式下的模块操作

模块	停止	VLPR	VLPW	VLPS	LLS	VLLSx
内核模块						
NVIC	静态	FF	FF	静态	静态	关
系统模块						
模式控制器	FF	FF	FF	FF	FF	FF
LLWU ¹	静态	静态	静态	静态	FF	FF ²
调节器	开	低功耗	低功耗	低功耗	低功耗	VLLS2/3 时为低功耗，VLLS0/1 时为关
LVD	开	禁用	禁用	禁用	禁用	禁用
掉电检测	开	开	开	开	开	VLLS1/2/3 时为开，VLLS0 时为可选禁用 ³
DMA	静态	FF	FF	静态	静态	关
看门狗	FF	FF	FF	FF	静态	关
EWM	开	开	开	开	开	关
时钟						
1 kHz LPO	开	开	开	开	开	VLLS1/2/3 时为开，VLLS0 时为关
系统振荡器 (OSC)	OSCCERCLK 可选	OSCCERCLK (4 MHz 晶振最大值)	OSCCERCLK (4 MHz 晶振最大值)	OSCCERCLK (4 MHz 晶振最大值)	限定为低范围/低功耗	VLLS1/2/3 时限定为低范围/低功耗，VLLS0 时为关
MCG	静态 - MCGIRCLK 可选；PLL 选择性开，但选通	4 MHz IRC	4 MHz IRC	静态 - 无时钟输出	静态 - 无时钟输出	关
内核时钟	关	4 MHz (最大值)	关	关	关	关

下一页继续介绍此表...

表 11. 低功耗模式下的模块操作 (继续)

系统时钟	关	4 MHz (最大值)	4 MHz (最大值)	关	关	关
总线时钟	关	4 MHz (最大值)	4 MHz (最大值)	关	关	关
存储器 and 存储器接口						
Flash	通电	1 MHz (最大值) 访问 - 无编程	低功耗	低功耗	关	关
SRAM_U 部分 ⁴	低功耗	低功耗	低功耗	低功耗	低功耗	VLLS3,2 时为低功耗
剩余的 SRAM_U 和全部 SRAM_L	低功耗	低功耗	低功耗	低功耗	低功耗	VLLS3 中的低功耗
FlexMemory	低功耗	低功耗 ⁵	低功耗	低功耗	低功耗	关
寄存器文件 ⁶	通电	通电	通电	通电	通电	通电
EzPort	禁用	禁用	禁用	禁用	禁用	禁用
通信接口						
USB FS/LS	静态	静态	静态	静态	静态	关
USB DCD	静态	FF	FF	静态	静态	关
USB 电压调节器	可选	可选	可选	可选	可选	可选
UART	静态, 在边沿唤醒	125 kbps	125 kbps	静态, 在边沿唤醒	静态	关
SPI	静态	1 Mbps	1 Mbps	静态	静态	关
I ² C	静态, 地址匹配唤醒	100 kbps	100 kbps	静态, 地址匹配唤醒	静态	关
I ² S	带外部时钟的 FF ⁷	FF	FF	带外部时钟的 FF ⁷	静态	关
安全性						
CRC	静态	FF	FF	静态	静态	关
定时器						
FTM	静态	FF	FF	静态	静态	关
PIT	静态	FF	FF	静态	静态	关
PDB	静态	FF	FF	静态	静态	关
LPTMR	FF	FF	FF	FF	FF	FF ⁸
RTC- 32 Khz OSC ⁶	FF	FF	FF	FF	FF	FF
CMT	静态	FF	FF	静态	静态	关
模拟						
16 位 ADC	仅限 ADC 内部 时钟	FF	FF	仅限 ADC 内部 时钟	静态	关
CMP ⁹	HS 或 LS 比较	FF	FF	HS 或 LS 比较	LS 比较	VLLS1/2/3 时为 LS 比较, VLLS0 时为关
6 位 DAC	静态	FF	FF	静态	静态	静态
VREF	FF	FF	FF	FF	静态	关
人机接口						
GPIO	唤醒	FF	FF	唤醒	静态, 引脚锁存	关, 引脚锁存
TSI	唤醒	FF	FF	唤醒	唤醒 ¹⁰	唤醒 ^{10, 8}

1. 使用 LLWU 模块，则用于此芯片的外部引脚不需要使能相关的外设功能。它只需将控制该引脚（GPIO 或外设）的功能配置为输入，以便允许 LLWU 上发生一次转换。
2. 由于 LPO 时钟源禁用，VLLS0 期间将旁路滤波器。
3. SMC 模块中的 VLLSCTRL[PORPO]位控制该选项。
4. SRAM_U 模块的 8 KB 部分在低功耗模式 VLLS2 中保持通电状态。
5. 使能为 EEPROM 的 FlexRAM 在 VLPR 下不可写入，且会忽略写操作。允许对 VLPR 下使能为 EEPROM 的 FlexRAM 进行读访问 FlexRAM 配置为传统 RAM 后，对其无访问限制。
6. 这些元件在 BAT 功耗模式下仍然保持通电状态。
7. 使用外部生成的位时钟或外部生成的音频主时钟（包括 EXTAL）。
8. VLLS0 时不提供系统 OSC 和 LPO 时钟源。
9. 停止模式或 VLPS 时的 CMP 支持高速或低速外部引脚间或外部引脚与 DAC 之间的比较。LLS 或 VLLSx 时的 CMP 仅支持低速外部引脚间或外部引脚与 DAC 之间的比较。停止、VLPS、LLS、或 VLLSx 模式下不提供窗口、采样和滤波工作模式。
10. 从 LLS 和 VLLSx 模式进行 TSI 唤醒仅限于单个可选引脚。

6 支持的功耗模式转换

本章概述支持的功耗模式转换。重要的是需记住：在所有功耗模式之间切换时，存在一定限制。

下图旨在为您提供支持的功耗模式转换图形说明：

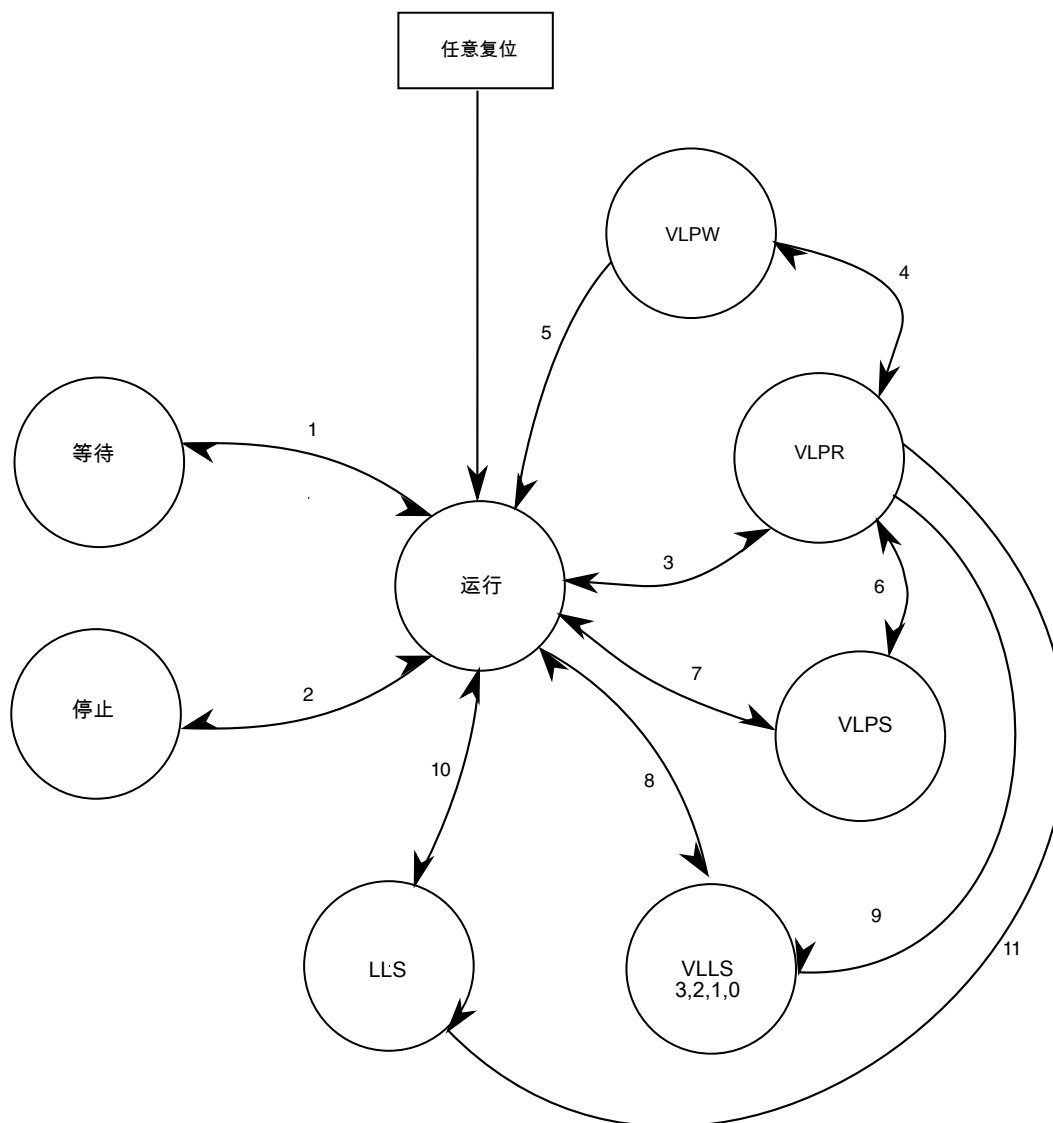


图 2. 功耗模式状态转换示意图

如上图所示，并非支持所有模式转换，比如当器件处于 VLLSx 模式并且想切换到停止模式时，那么必须首先使器件返回运行模式，然后才能切换到停止模式。

任何复位都会使芯片返回正常运行状态。

更多详情，请参考器件参考手册中的“功耗模式转换”一章。

7 进入功耗模式的演示

在本章中，我们将基于演示软件，回顾进入有效功耗模式的条件。

本演示的目的是熟悉 Kinetis 系列器件进入可用功耗模式所需的步骤。

需要说明的是，本演示中，退出任意模式时都只使用一种模式，但从任意特定模式退出时可以有很多方法，如前文所述。

本演示可在采用 K20DX128 器件的 TWR-K20D50M 上运行。

低功耗退出模式：

外部引脚 SW2 已使能，以便通过 LLWU 从任何 LLS 和 VLLSx 模式唤醒；来自这些模式的全部唤醒中断都必须通过 LLWU_ISR。

所有其他模式（非漏电流）可通过任意可用端口使能引脚发出的中断退出低功耗模式，本演示中为 SW2 或 SW3。

要求如下：

1. IAR Embedded Workbench for ARM v6.30 或 CodeWarrior MCU V10.1（安装 MCU 10.1 Kinetis 50 MHz 软件包）
2. TWR-K20D50M 塔式板。
3. P&E OSBDM OSJTAG Virtual Serial Toolkit。可前往下列网站下载：<http://www.pemicro.com/osbdm/index.cfm>
4. 演示项目：“low_power_demo”。

此工具包包含 P&E 终端实用程序，可通过 USB 串行将菜单选项可视化。

运行演示前，需将 IDE 配置为适当设置，具体取决于所用的编程/调试接口。

注

如果已安装首选调试接口，则可忽略以下内容，直接转到第 7.2 章“运行演示的步骤”。

7.1 Flash 编程和调试设置

本节说明利用 IAR 或 CodeWarrior IDE 以及下列任意接口：OSJTAG、P&E Multilink、Segger J-Link 在 Kinetis 板上编程并调试应用程序时所需的设置。

7.1.1 IAR 和 OSJTAG/P&E Multilink

这些连接必须使用 Flashloader FlashK20Xxxx。

解压演示项目 zip 文件内的 iar_FlashK20Xxxx_8k_ram.zip，并按下列步骤操作。

1. 将该 zip 中的所有文件（“readme.txt”除外）放到 IAR Systems\Embedded Workbench.6.0_x\arm\config\flashloader\Freescale 文件夹下

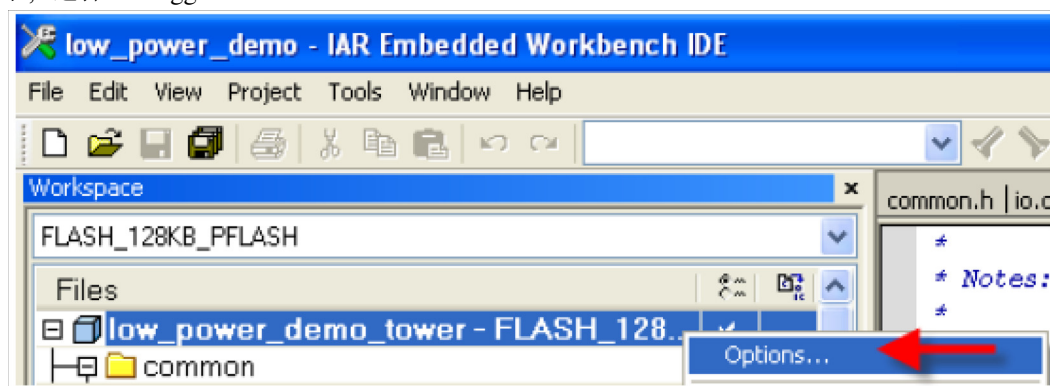
注

注意：此操作适用于当前 IAR 版本(IAR 6.30.4)，更新版本无需执行这些步骤。

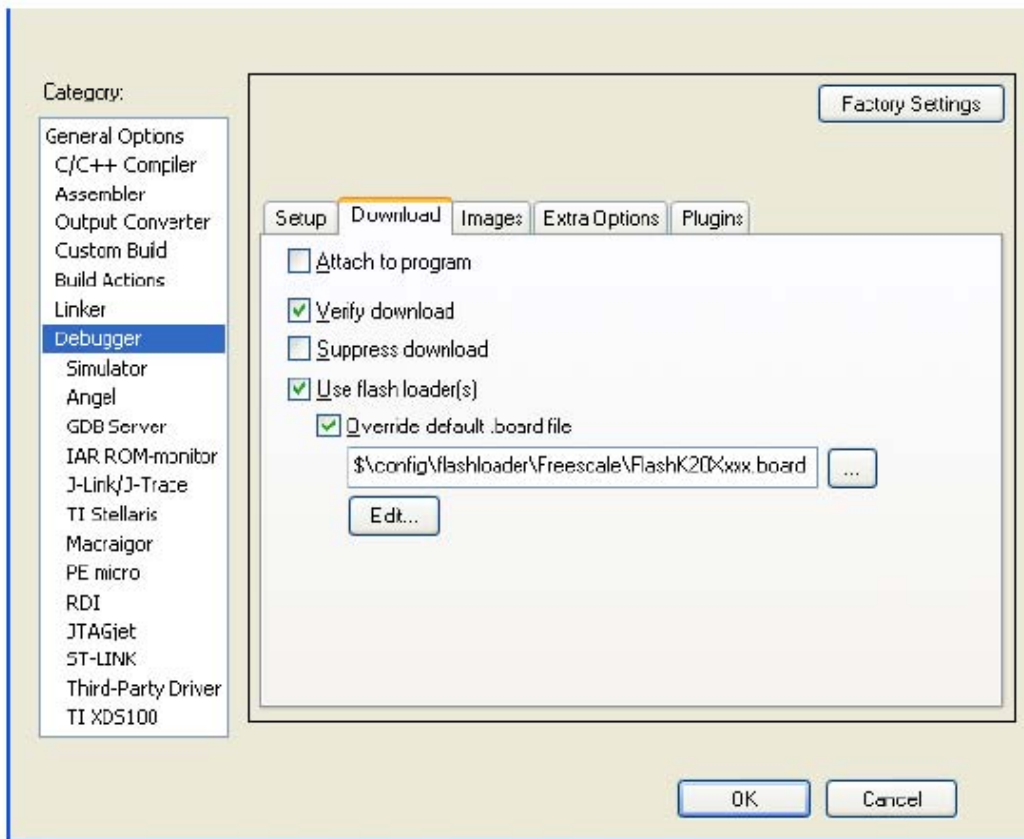
2. 打开 IAR 工作区文件，为项目选择 Flash 配置
 - a. 打开 IAR Embedded Workbench IDE。
 - b. 打开 low_power_demo 工作区。

要完成该操作，可拖放下列路径上的“low_power_demo.eww”文件：{演示安装路径}\buildIar\low_power_demo

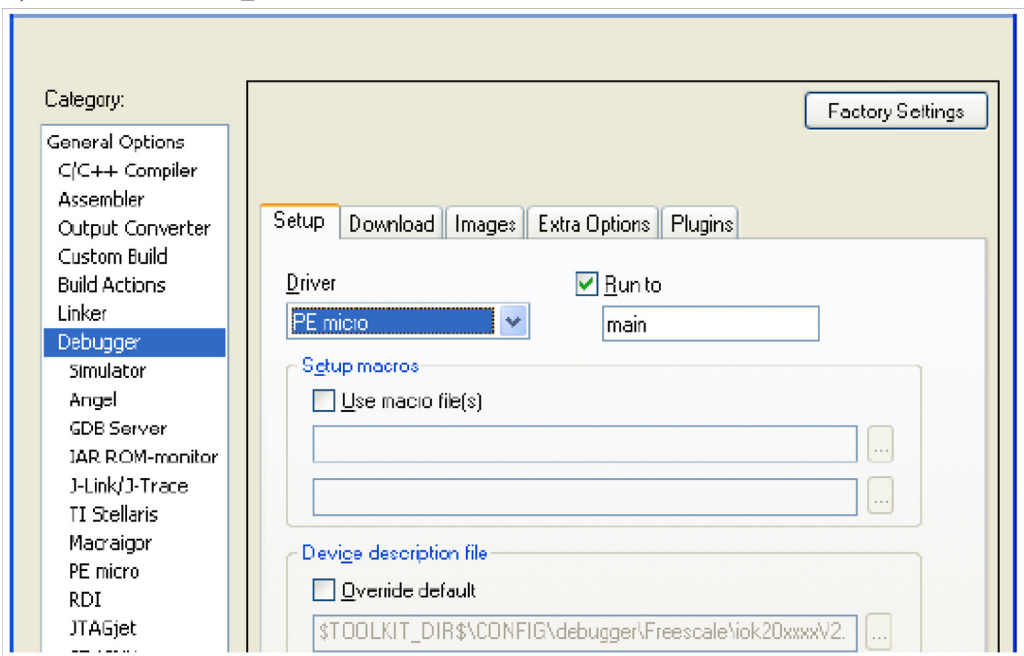
3. 在项目选项中，选择 Debugger->Download。



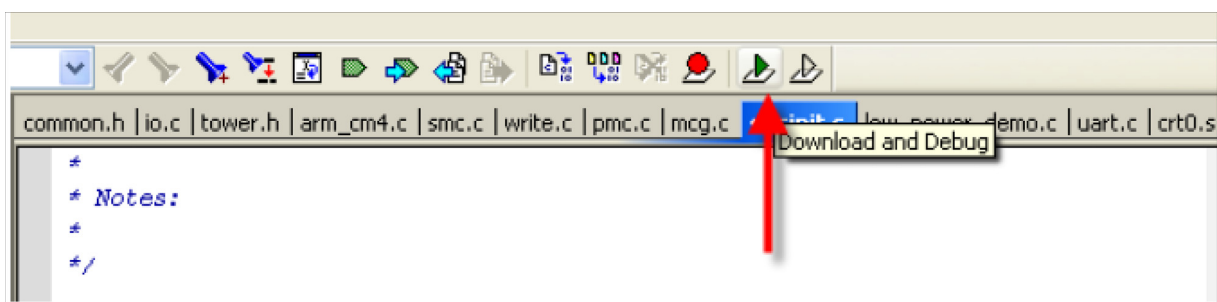
4. 勾选"Use Flash loader(s)"选择框。
5. 勾选"Override default .board file box"选择框。
6. 提供 FlashK20Xxxx.board 文件路径。该路径将表示如下: \$TOOLKIT_DIR\$\config\flashloader\Freescale\FlashK20Xxxx.board。



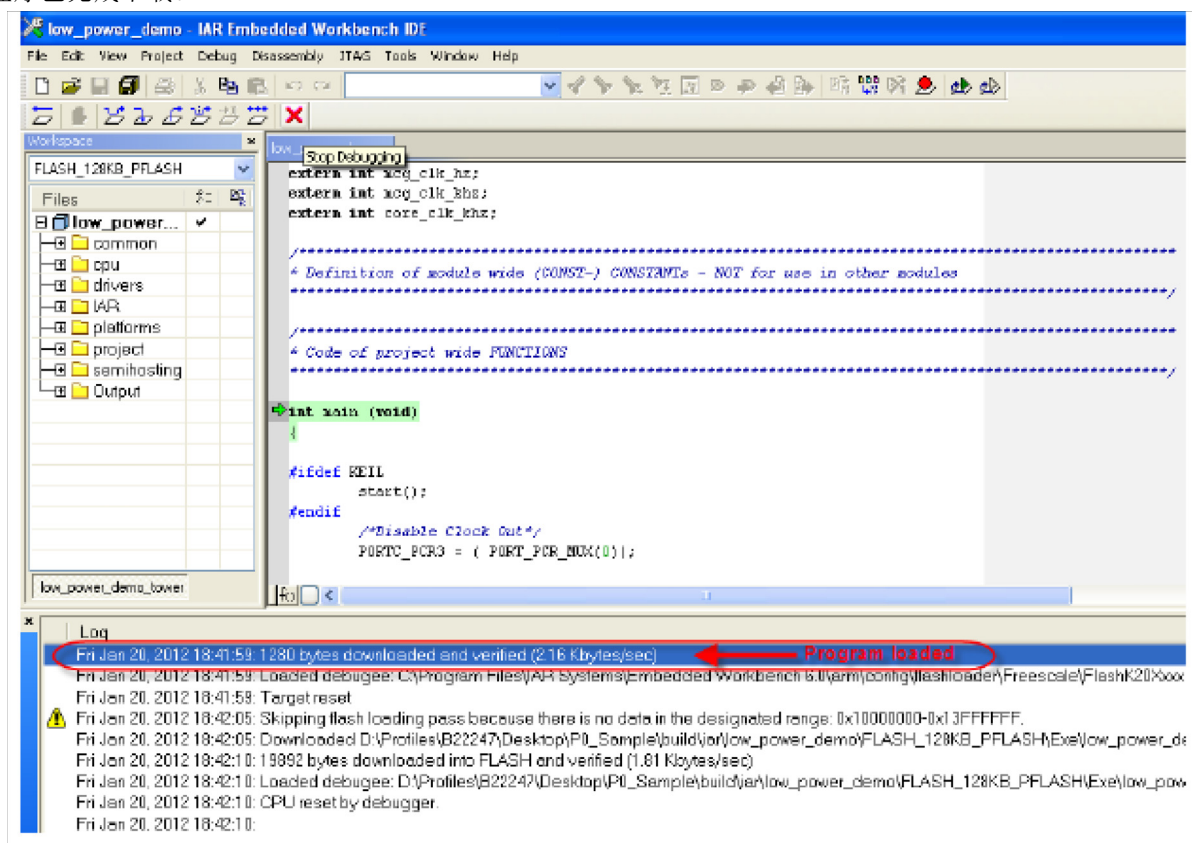
7. 在 Debugger 中, 选择驱动程序 PE_micro。



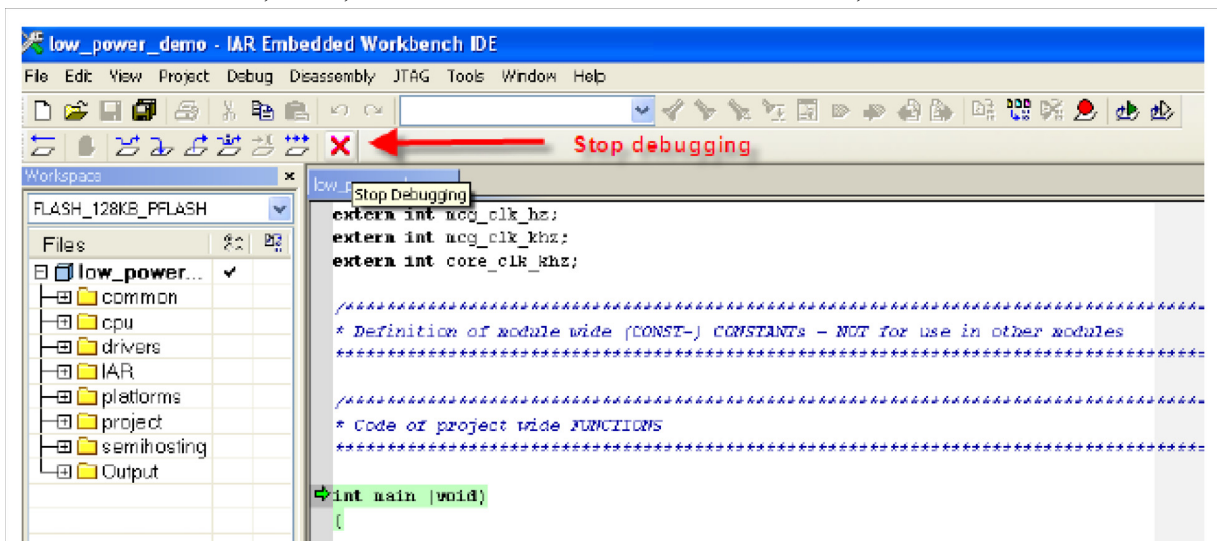
8. 要对应用程序进行编程和调试, 请单击调试图标。



9. 确认程序已完成下载。



10. 此时已完成应用程序的下载；现在，我们需要退出并关闭调试器：单击 X 标记，如下图所示：



注

停止调试会话是因为调试器激活时具有较高的功耗。

7.1.2 IAR 和 J-link

要正常工作，必须使用：**J-Link DLL V4.36i**

可前往下列网站下载：

http://www.segger.com/jlink-software.html?step=1&file=JLink_436kl

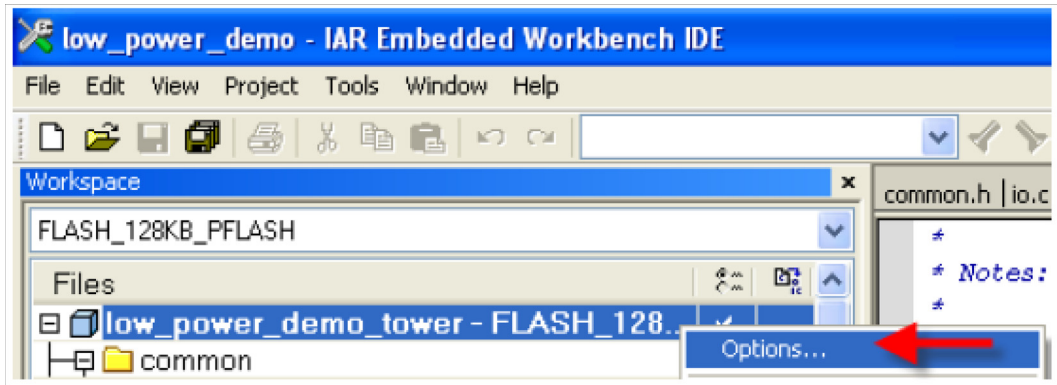
IAR 必须如下配置：

1. 打开 IAR 工作区文件，为项目选择 Flash 配置。
2. 打开 IAR Embedded Workbench IDE。
3. 打开 *low_power_demo* 工作区。

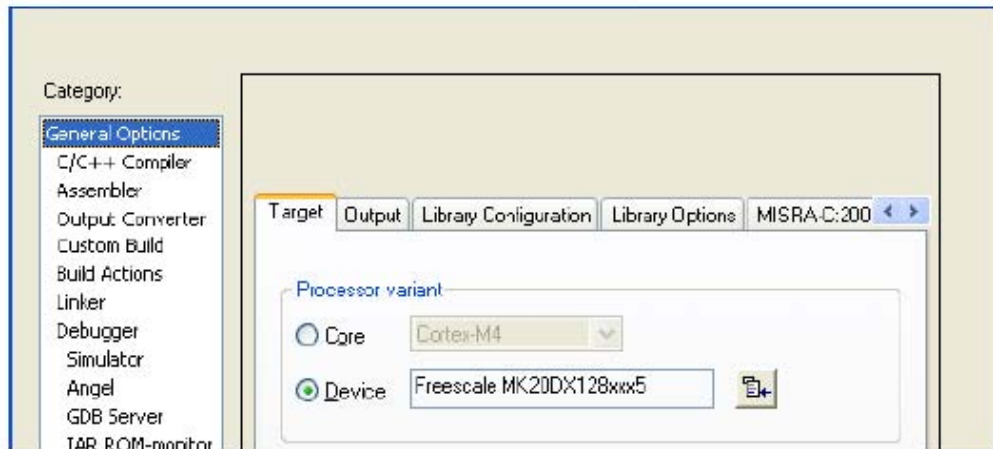
要完成该操作，可拖放下列路径上的“low_power_demo.eww”文件：

{演示安装路径}\build\iar\low_power_demo

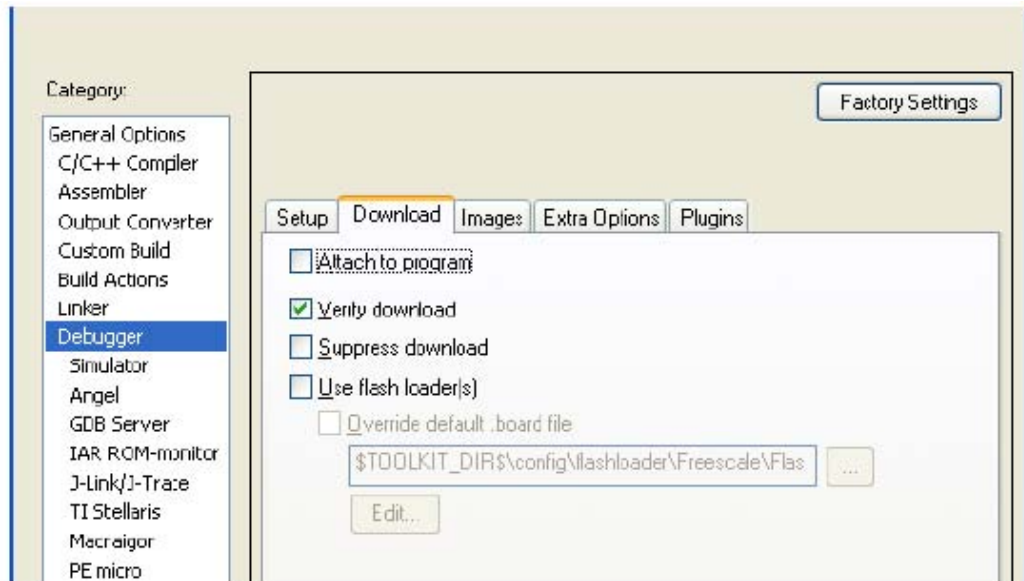
4. 在 project 选项中，选择 General Options。



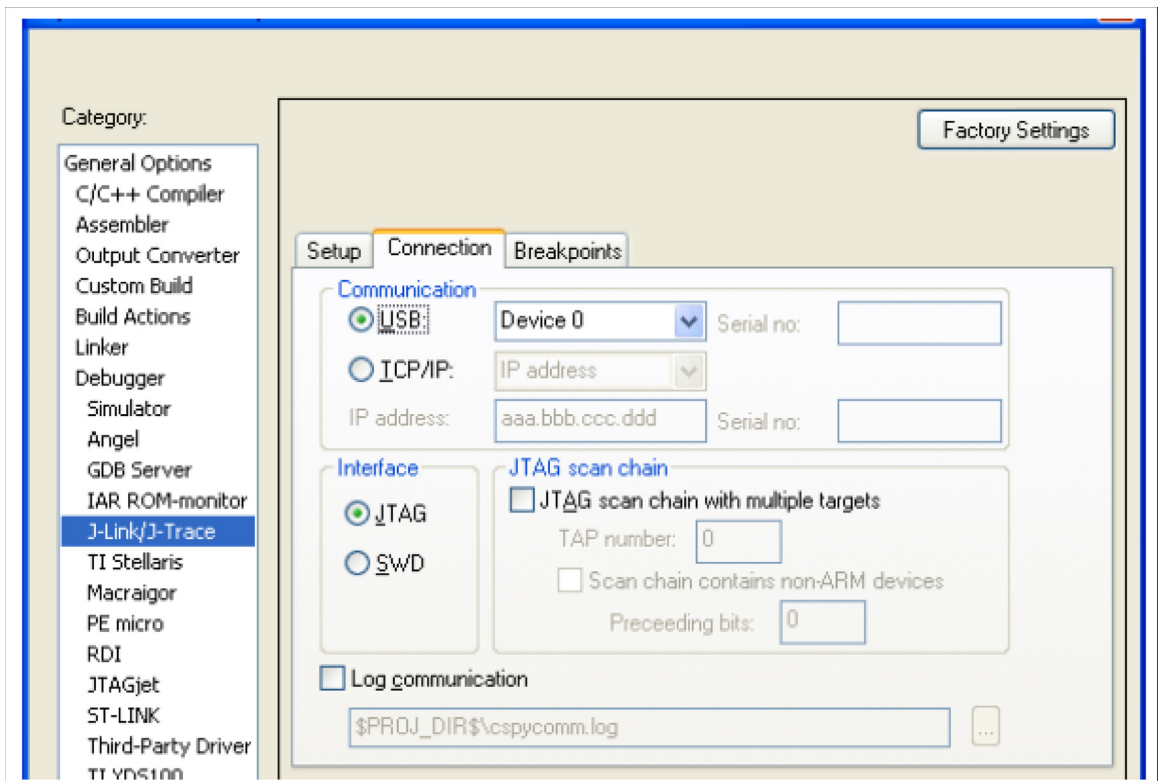
5. 选择 project 选项上的器件 MK20DX128xxx5，如下所示：



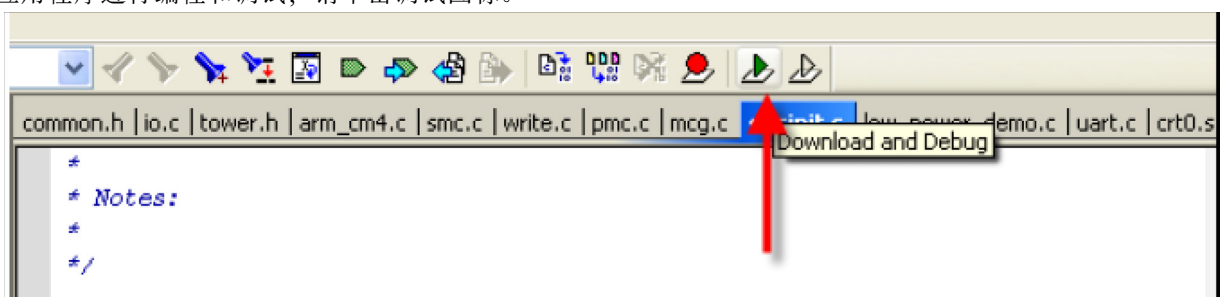
6. 在 Debugger 目录下：取消勾选“Use flash loaders”。



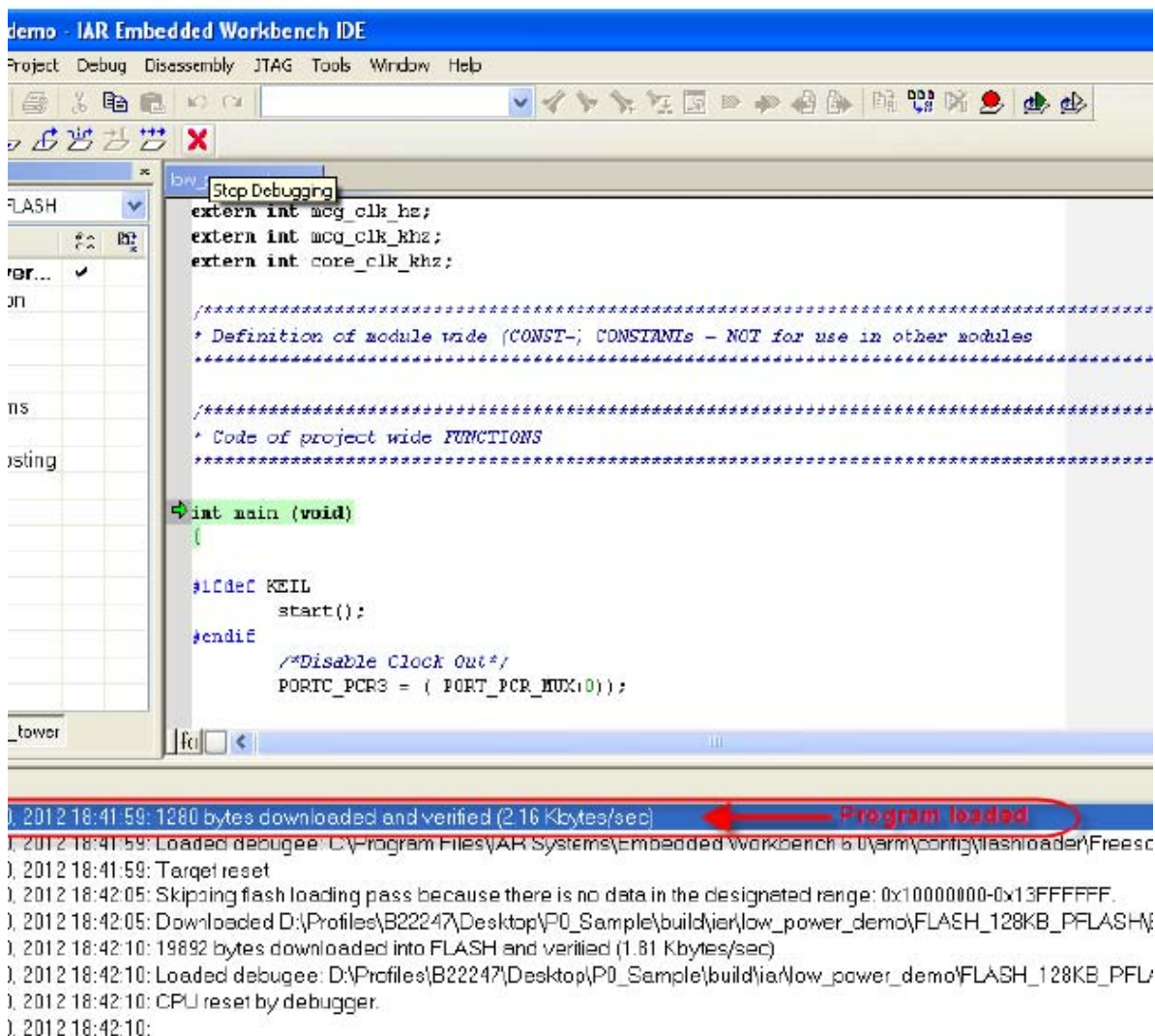
7. 在 J-link/J-trace 的 Connection 窗口中：选择 JTAG 接口。



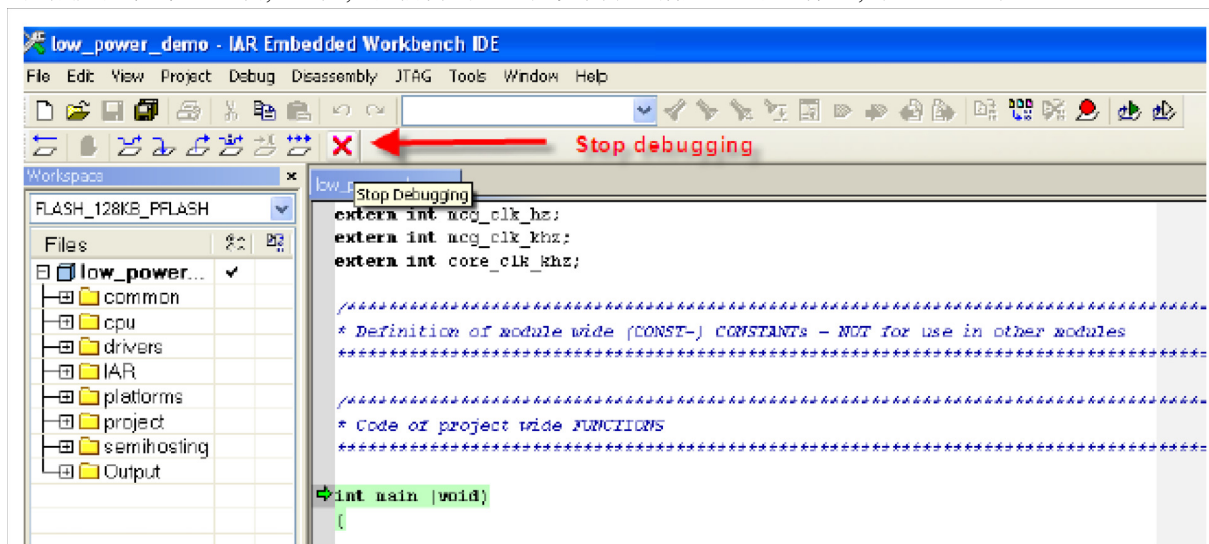
8. 要对应用程序进行编程和调试，请单击调试图标。



9. 确认程序已完成下载。



10. 此时已完成应用程序的下载；现在，我们需要退出并关闭调试器：单击 X 标记，如下图所示：



注

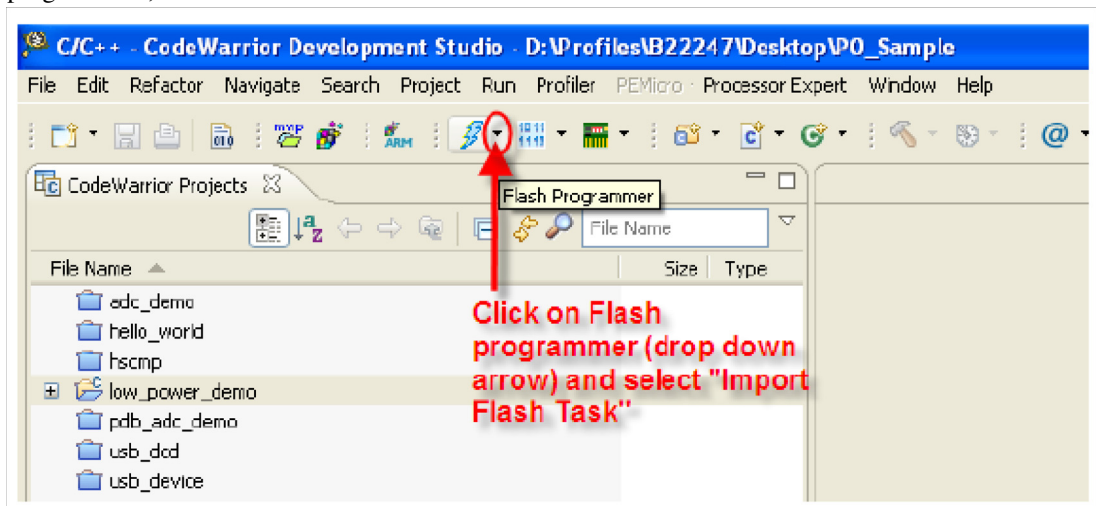
停止调试会话是因为调试器激活时具有较高的功耗。

7.1.3 CW MCU V10.1 和 OSJTAG/P&E Multilink

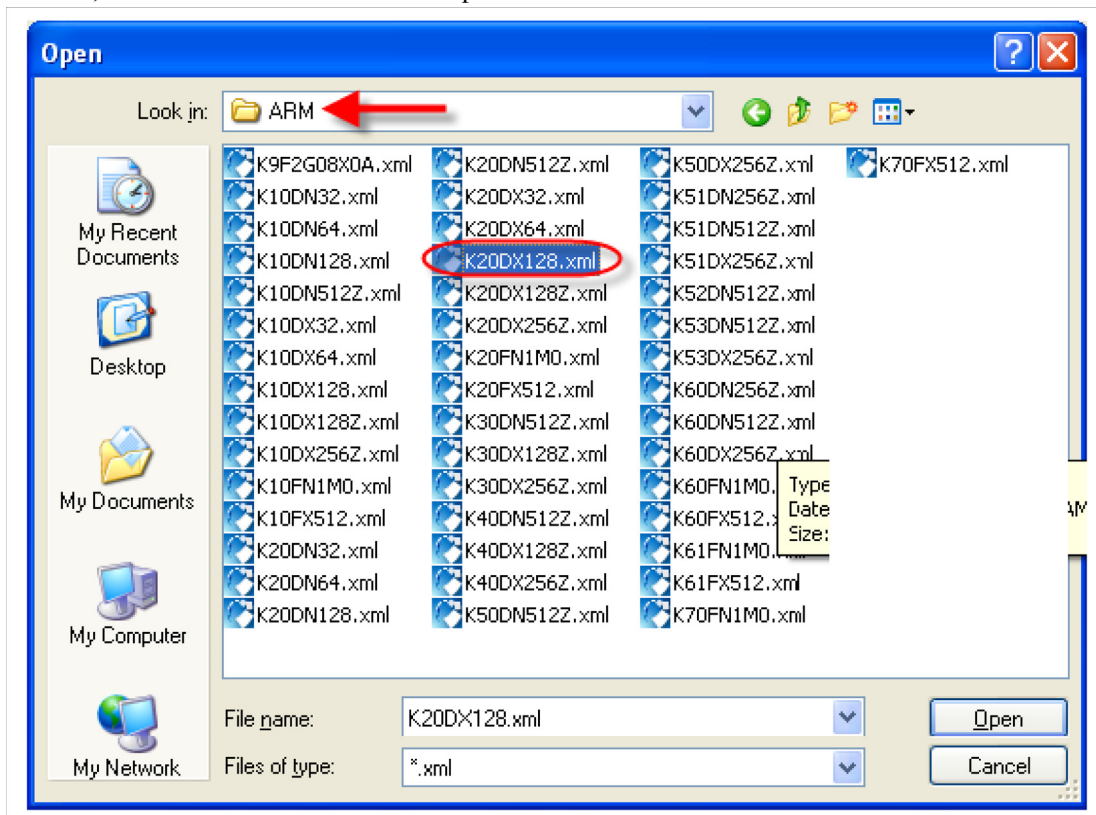
要使 CW10.1 正常工作，其配置必须如下：

更新 Flash 烧录器：

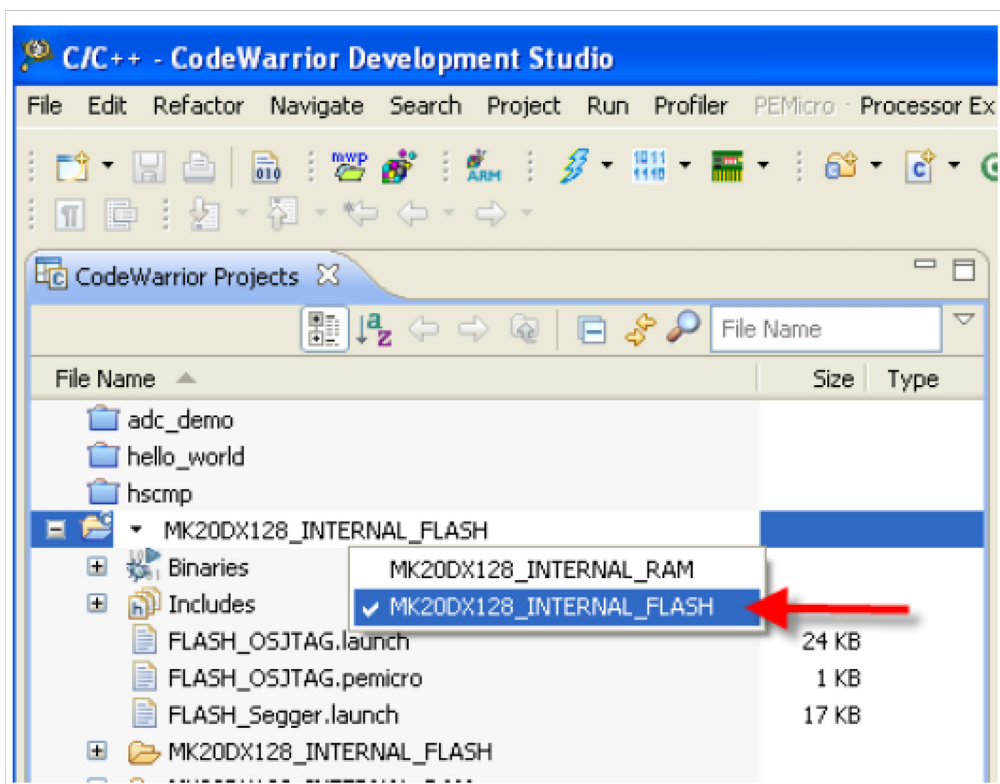
1. 单击 Flash programmer，然后导入“Flash Task”。



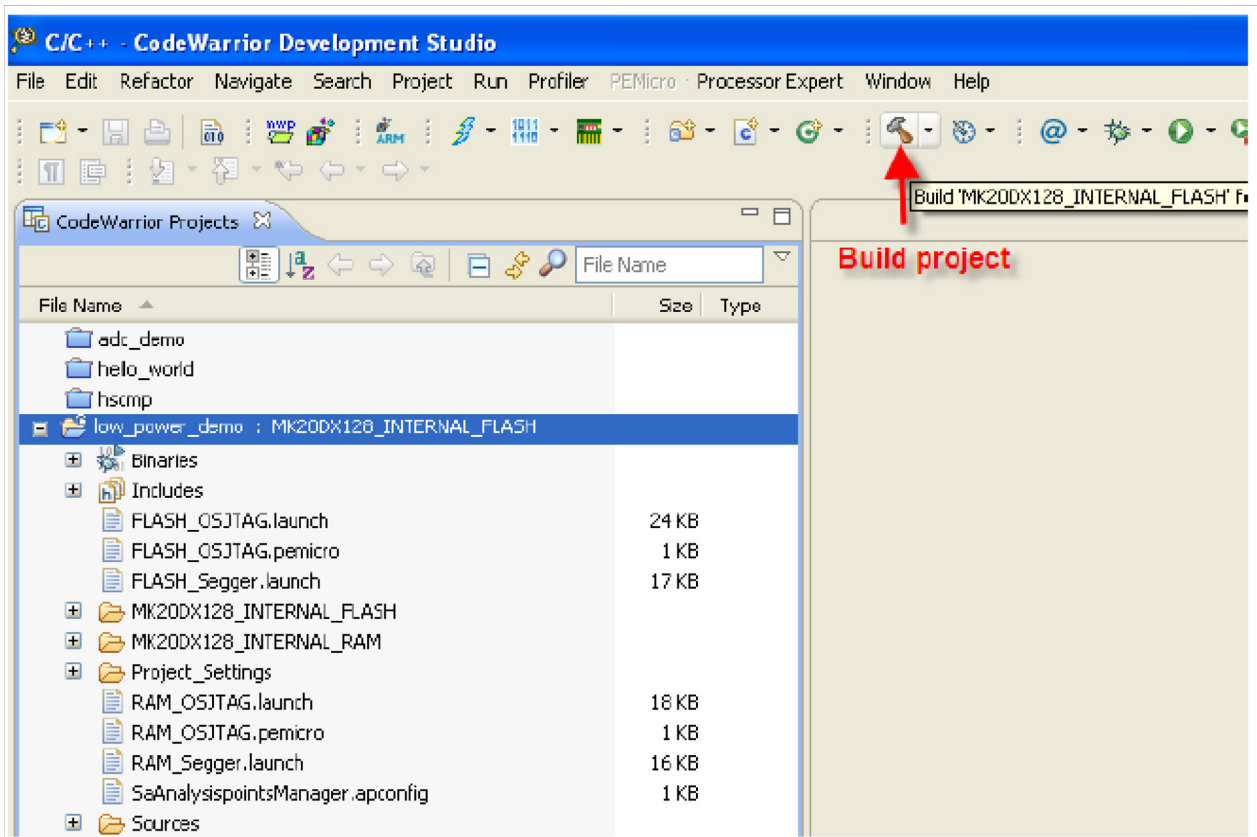
2. 打开 ARM 文件夹，选择 K20DX128.xml 并单击 Open。



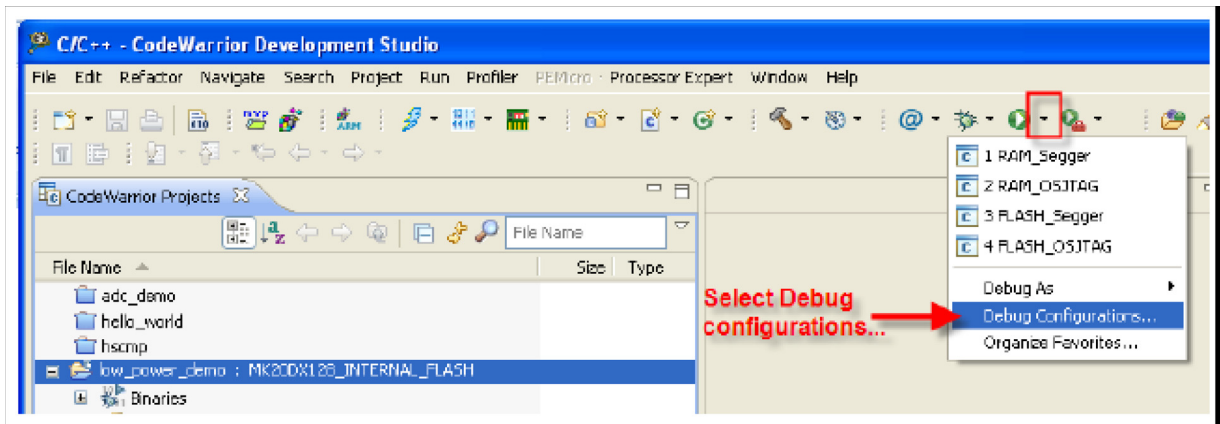
3. 选择 Flash 配置。



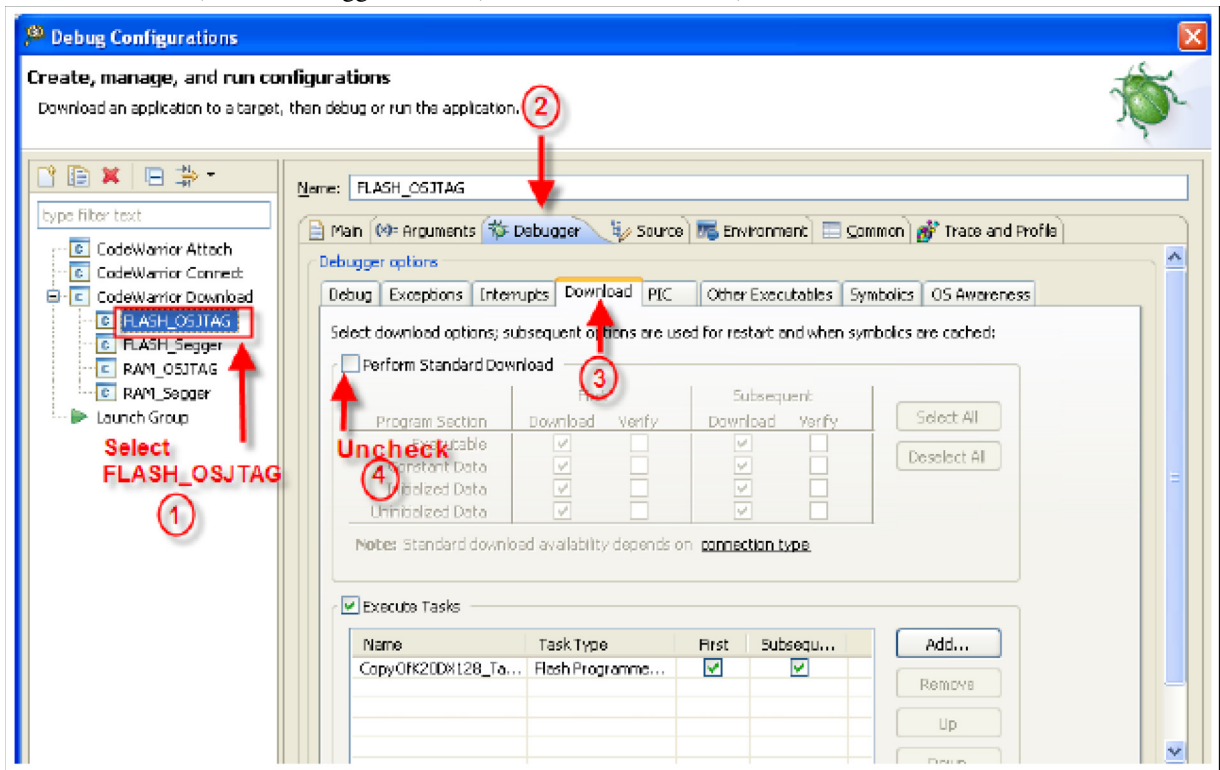
4. 如下所示，点击锤子图标，建立项目：



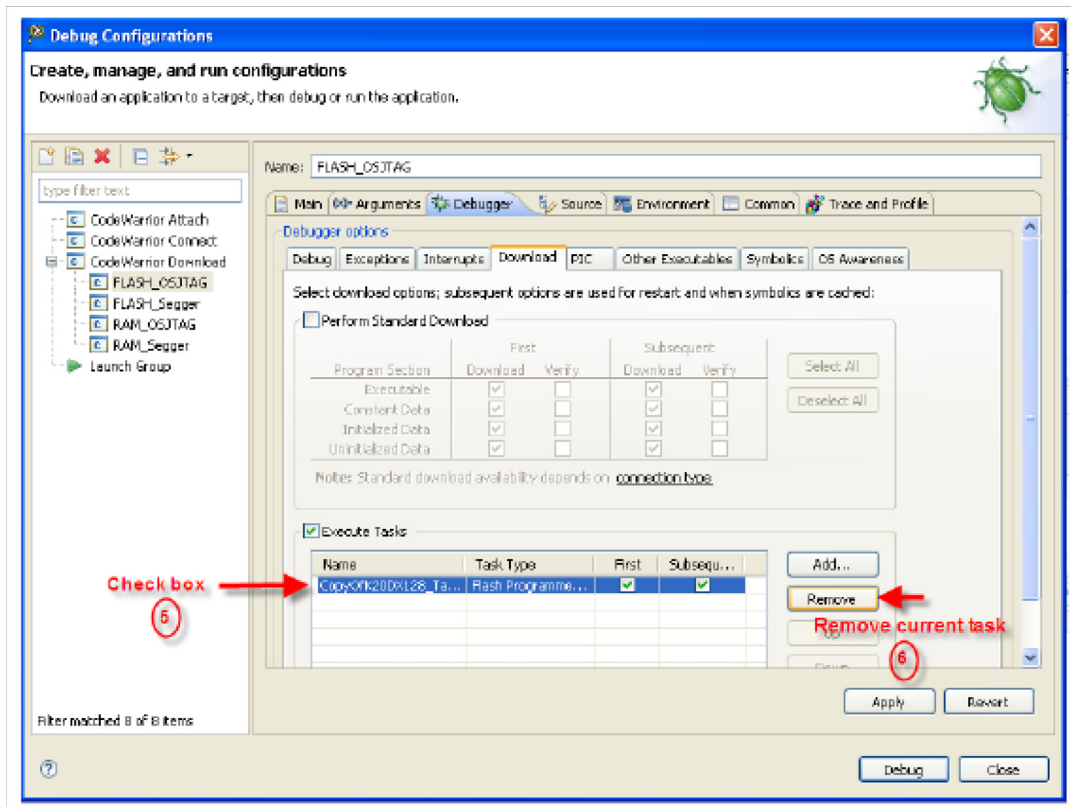
5. 点击下拉箭头，选择相应的调试配置。



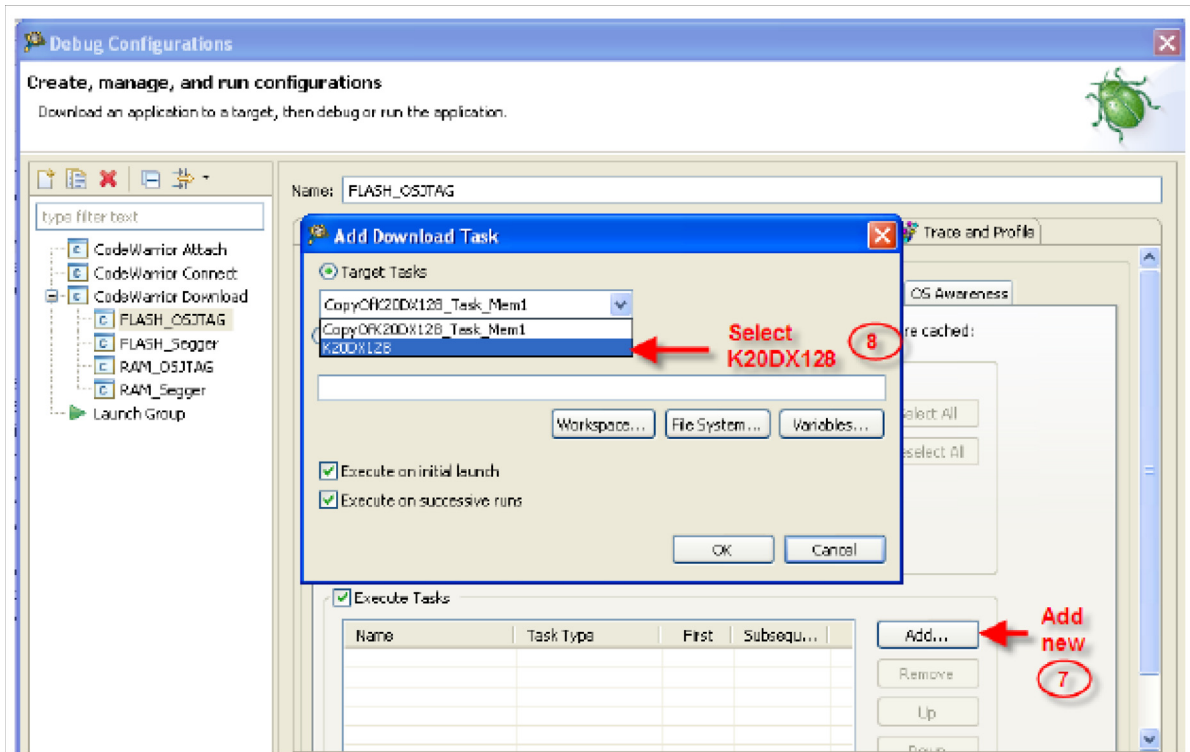
6. 选择 FLASH_OSJTAG，进入 Debugger 选项卡，打开 Download 部分，取消勾选“Perform Standard Download”。



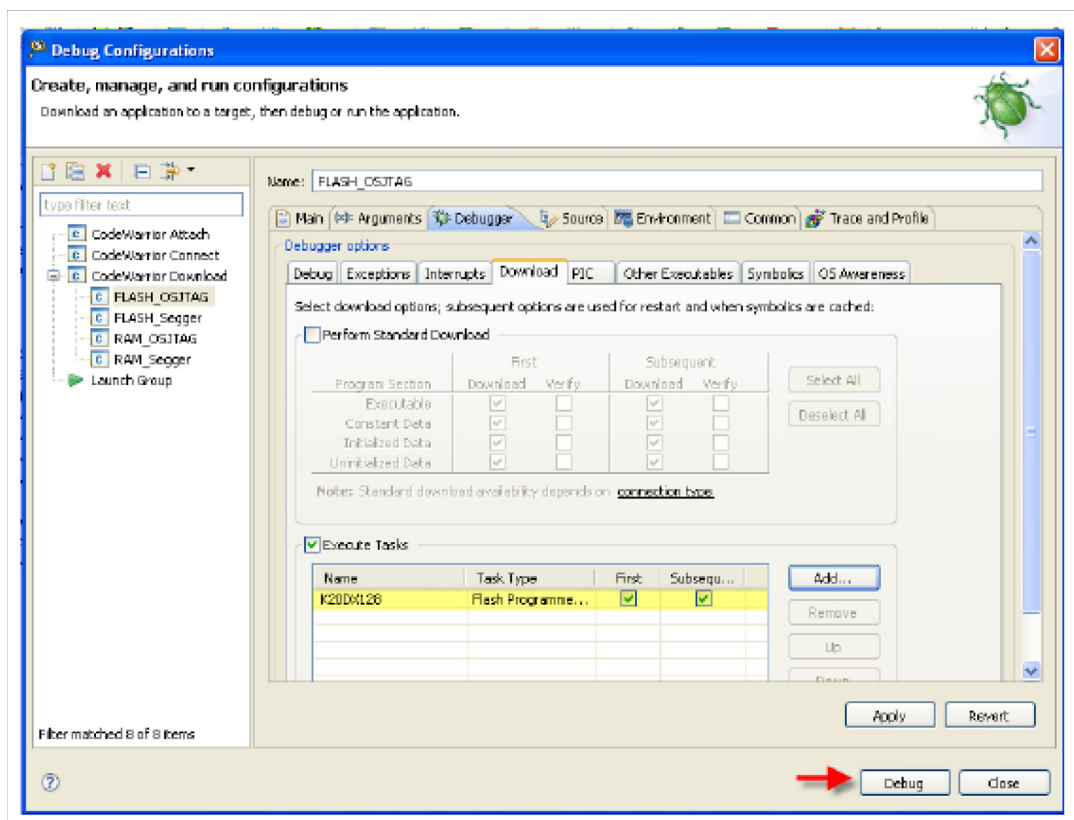
7. 勾选“Execute Tasks”选项框，选择当前任务并单击 Remove。



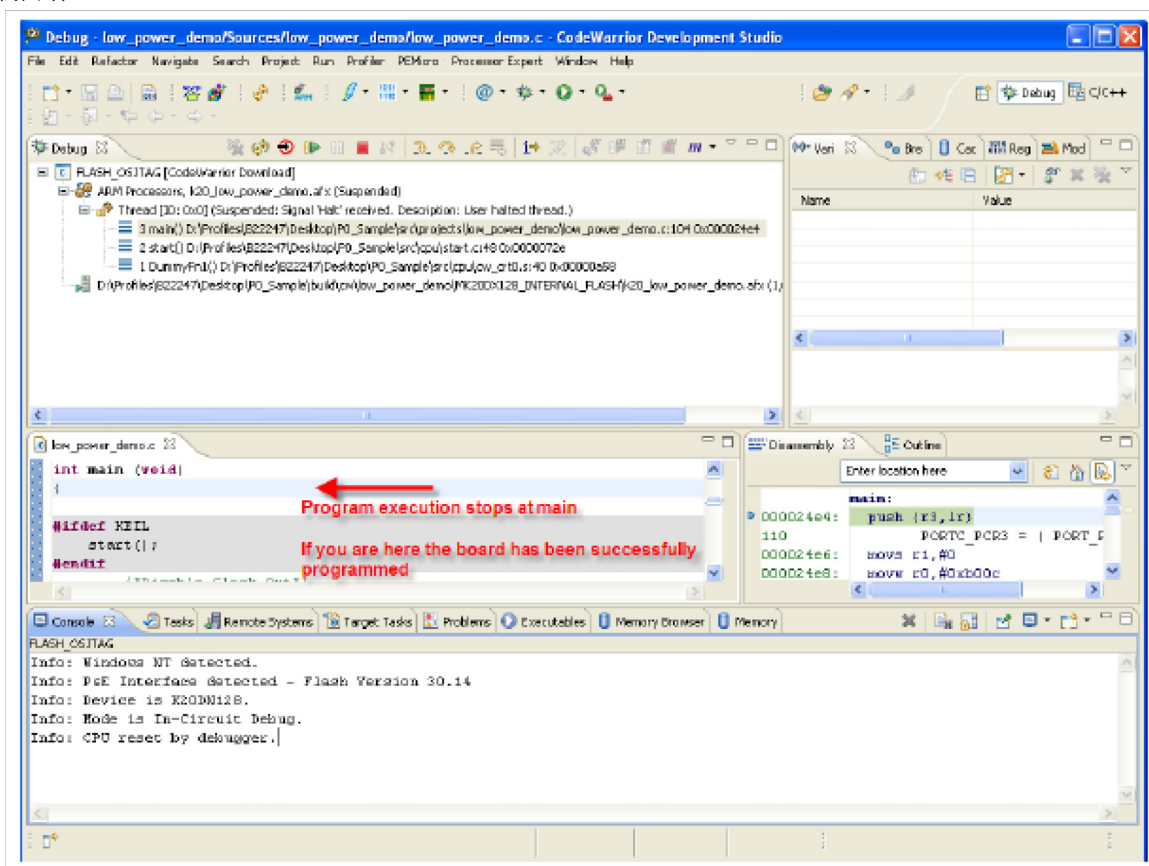
8. 单击 Add 按钮，选择 K20DX128。



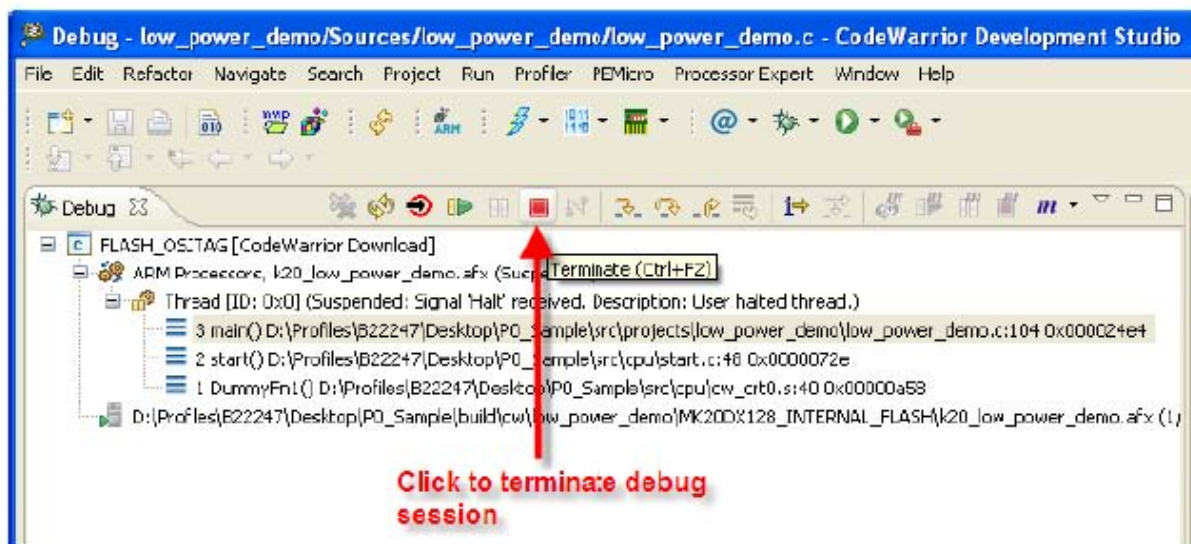
9. 单击 Apply 和 Debug。



10. 验证下载内容。



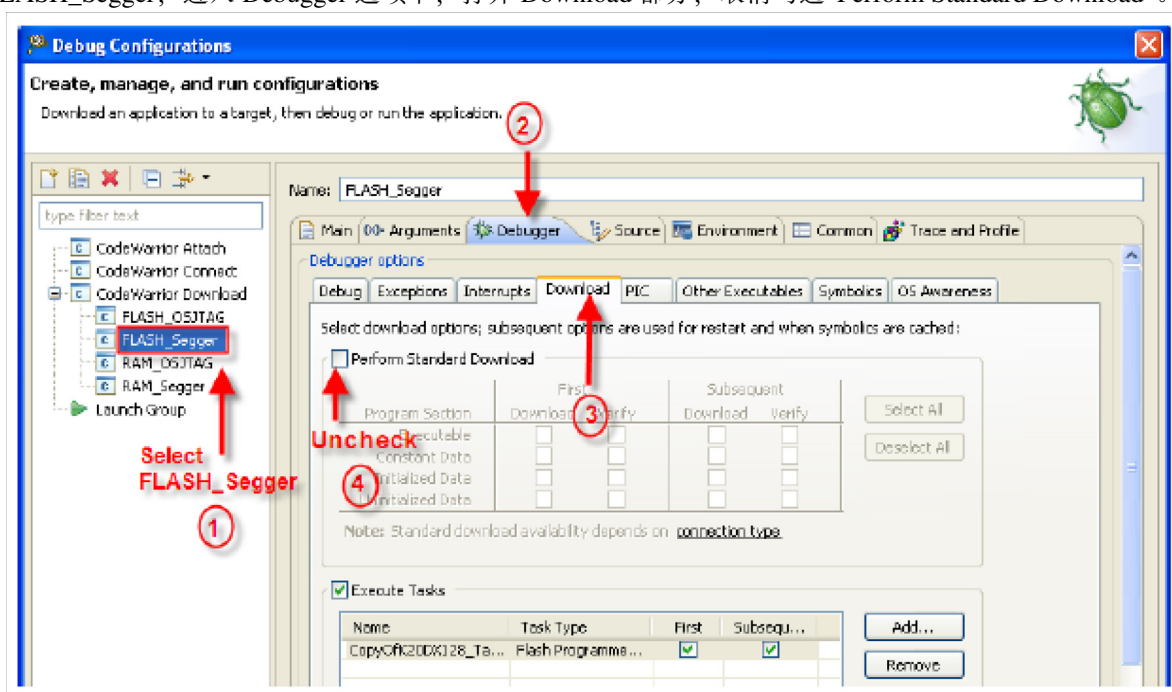
11. 要正确测量功耗，可单击红色方框按钮，终止调试会话。



CW MCU V10.1 和 JLink

重复上文步骤（针对 CW MCU V10.1 和 OSJTAG/P&E Multilink），但步骤 6 必须选择 **FLASH_Segger**。

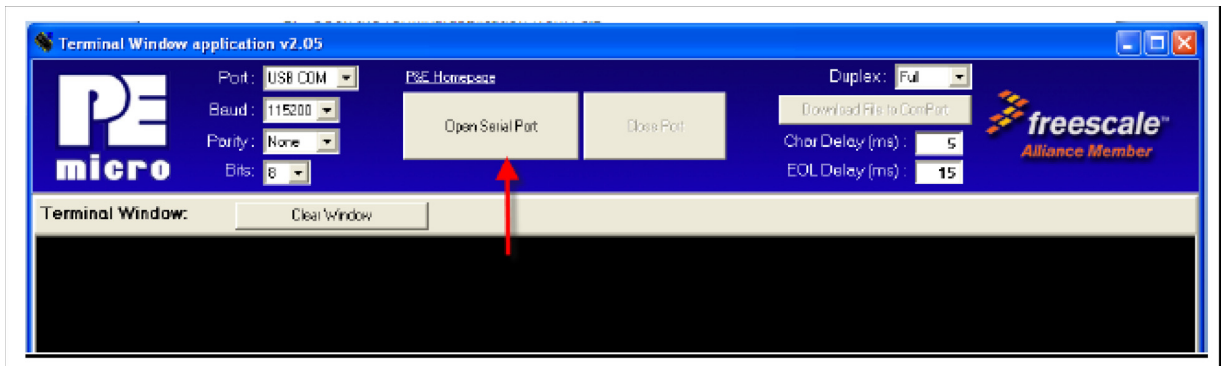
选择 FLASH_Segger，进入 Debugger 选项卡，打开 Download 部分，取消勾选“Perform Standard Download”。



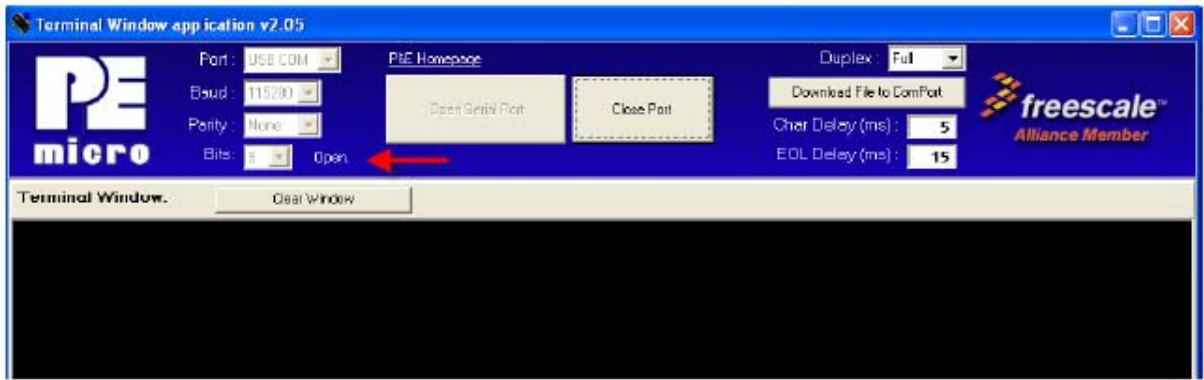
7.2 演示运行步骤

成功将应用程序载入器件并停止调试器后，可执行下列步骤：

1. 从 P&E 打开 Terminal 应用程序
2. 将端口设置为 Port: USB COM、Baud: 115200、Parity: None、Bits: 8；单击“Open Serial Port”。

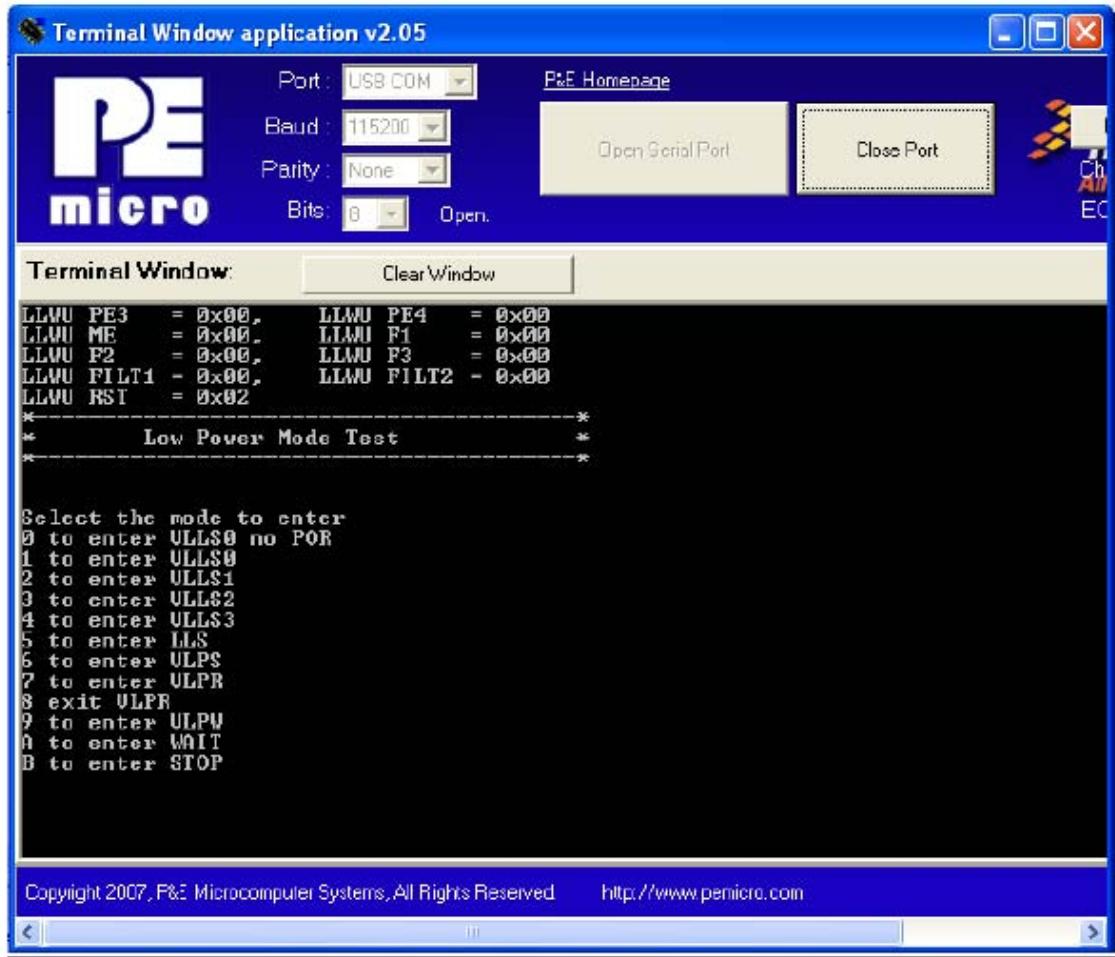


3. 确保端口顺利打开。



4. 然后，按一下板上的复位按钮。

5. 此时查看 Terminal Window，应当能看到所有可用选项。

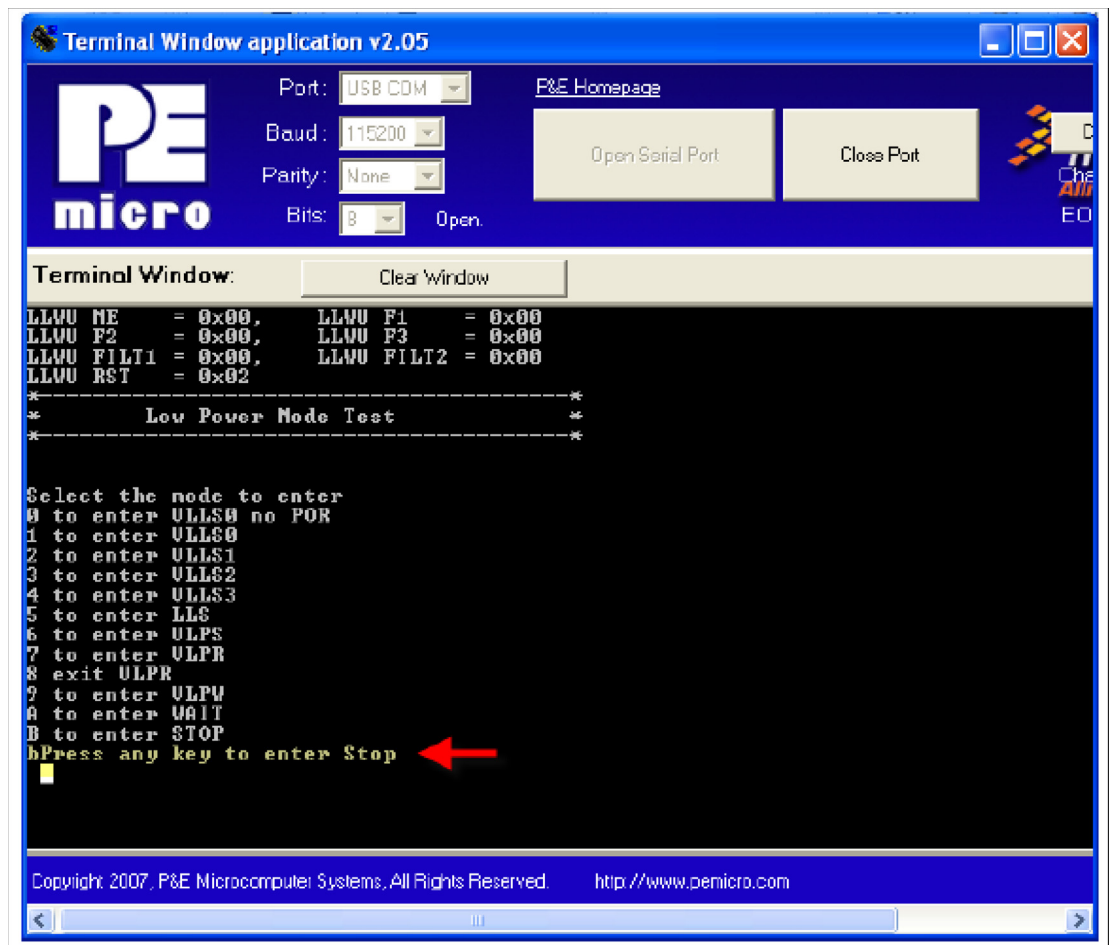


6. 按菜单上的指示操作。

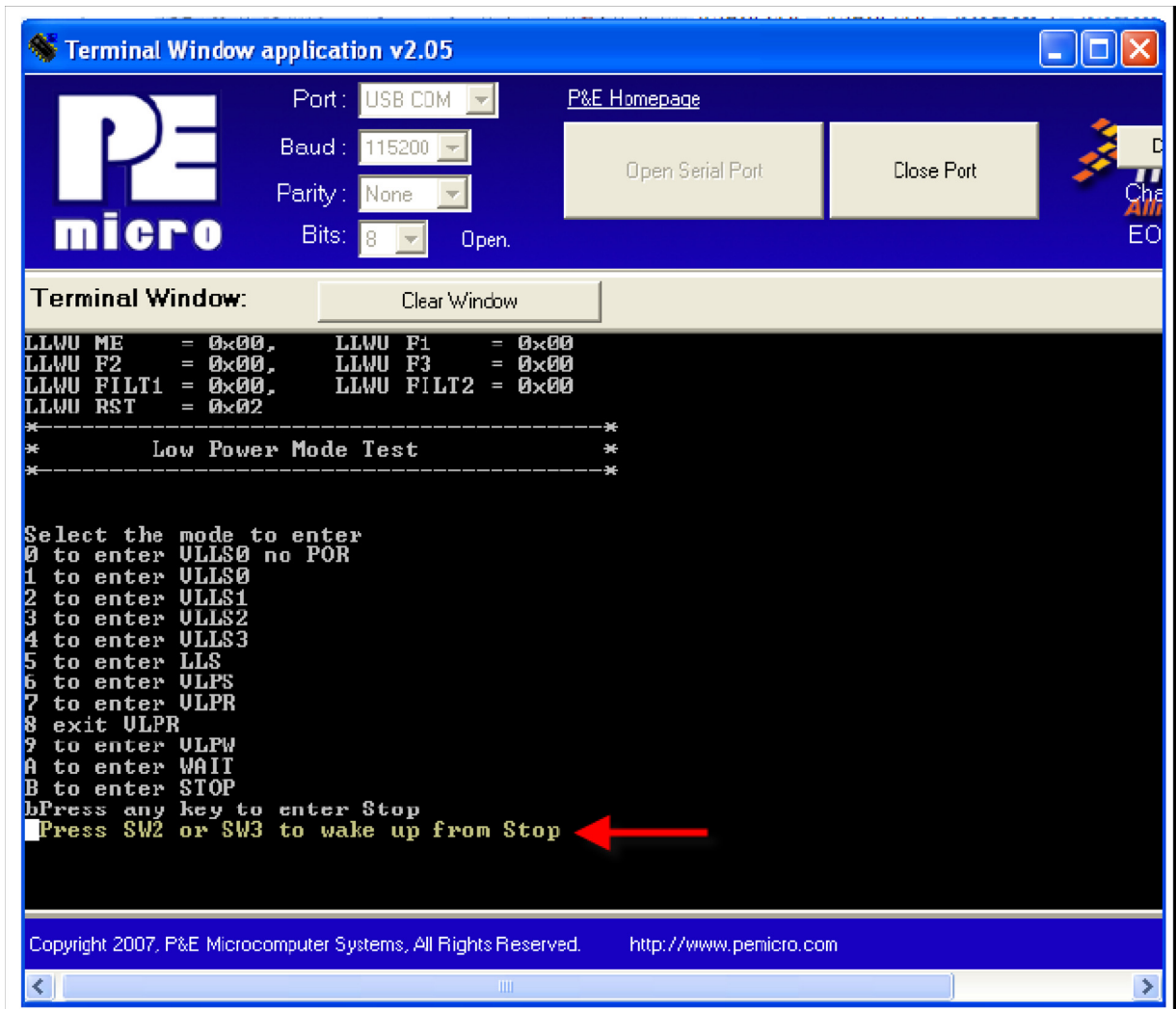
示例:

如需进入 STOP 模式:

- 在 Terminal Window 中输入字母 B。将显示下列消息:



- 按任意键。显示以下消息：



这表示目前处于停止模式。

- 要测量功耗（取决于您所用的开发板），应当拔下与 MCU 电源连接有关的跳线。

在 TWR-K20D50M（修订版 C）上，该跳线为 J25。

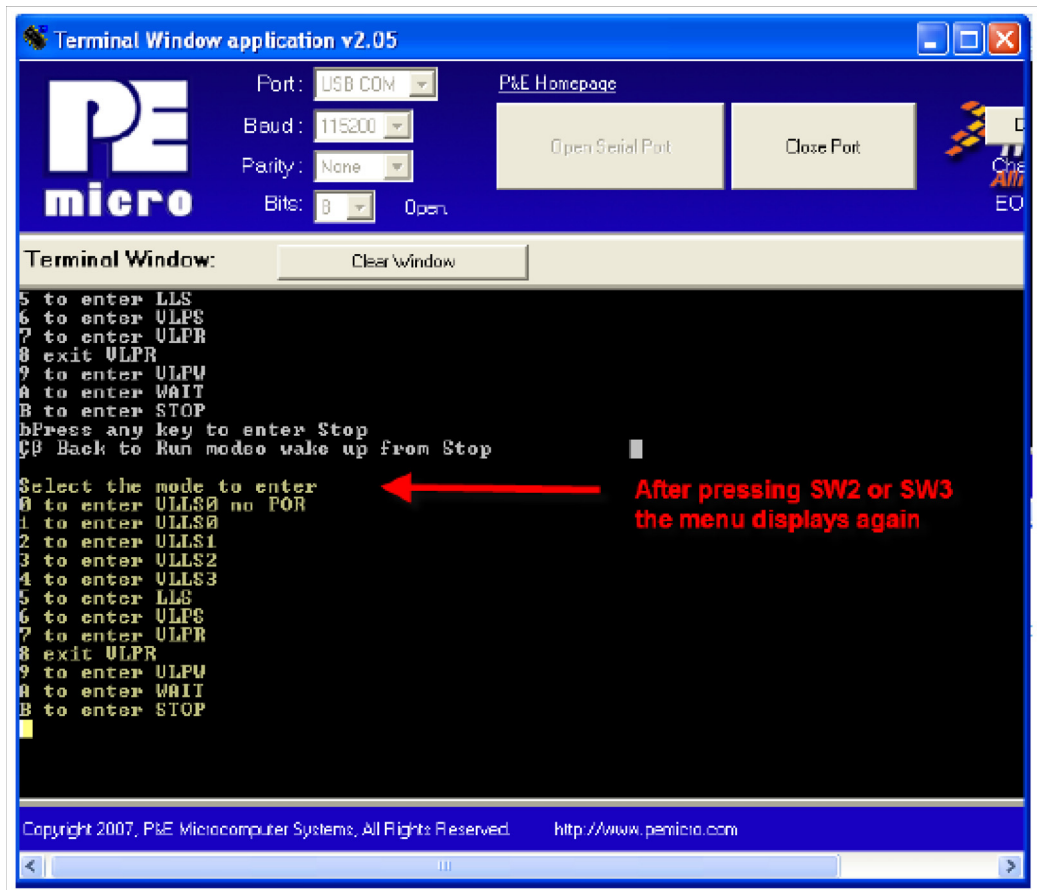
注

如果不确定是哪根跳线，请参考塔式板用户手册。

- 然后，测量这些端子之间的器件功耗：

测得的结果约为 320 μ A。

- 要退出此低功耗模式并进入另一种模式，请按下 SW2 或 SW3：



随后，可再次根据这些步骤进入另一功耗模式。

注

通过复位可退出任意功耗模式。

8 结语

进行低功耗设计时，必须考虑硬件和软件，因为二者对于成功设计高能效应用而言具有同等的重要性。

记住几点基本概念：将引脚设为已知状态，较高的内核频率和较高的温度都会增加功耗，禁用未使用模块的时钟。

9 参考

本应用笔记中的各章概括了每个主题的最重要细节。有关本文档各主题的更多详情，请参考下列内容：

功耗信息：

- 参考器件数据手册
搜索“功耗特性”

功耗模式的信息与详情：

- 参考器件参考手册
搜索“电源管理”一章



How to Reach Us:

Home Page:

freescale.com

Web Support:

freescale.com/support

本文档中的信息仅供系统和软件实施方使用 Freescale 产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale 保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性的或附带性的损害在内的所有责任。

Freescale 的数据表和/或规格中所提供的“典型”参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有运行参数，包括“经典值”在内，必须经由客户的技术专家对每个客户的应用程序进行验证。

Freescale 未转让与其专利权及其他权利相关的许可。Freescale 销售产品时遵循以下网址中包含的标准销售条款和条件：freescale.com/SalesTermsandConditions。

Freescale, the Freescale logo, and Kinetis, are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© 2012 Freescale Semiconductor, Inc.

© 2012 飞思卡尔半导体有限公司

Document Number AN4470
Revision 0, 01/2012

