AN14416

基于FRDM-MCXW71低功耗蓝牙连接协议栈启用看门狗定时器模块

第1.0版—2024年9月10日 **应用笔记**

文档信息

信息	内容
关键词	AN14416、MCXW71、FRDM-MCXW71、看门狗定时器模块、连接协议栈集成演示、看门狗功能、看门狗配置
摘要	本文档介绍了使用连接协议栈集成（Connectivity Stack Integration）演示工程实现看门狗定时器模块功能的过程，该工程是与FRDM-MCXW71 SDK软件包中的示例一起提供的。

1 介绍

本文介绍了实现看门狗-连接协议栈集成演示工程的过程，该工程是与FRDM-MCXW71 SDK软件包中的示例一起提供的，并在MCUXpresso IDE平台上进行开发。

看门狗定时器（WDOG）模块是一个可供系统使用的独立定时器。它提供了一个安全功能，以确保软件按计划执行。它还确保CPU不会陷入无限循环或执行非预期代码。如果在一定时间内未对WDOG模块提供服务（刷新），它会复位MCU。要刷新看门狗计数器，软件必须在超时期限到期之前执行刷新写入序列。恩智浦微控制器具有稳健的刷新机制，这使得任何跑飞的代码都几乎不可能刷新看门狗。

2 功能说明

如果发生系统故障，WDOG模块会提供一种故障安全机制，以确保系统复位到已知的运行状态。系统故障的例子包括CPU时钟停止或软件代码中存在跑飞的情况。看门狗计数器使用可选择的时钟源连续运行，并期望定期得到服务（刷新）。如果看门狗没有定期刷新，就会生成一个复位触发事件。

2.1 看门狗刷新机制

要刷新看门狗计数器，软件必须在超时期限到期之前执行刷新写入序列。重要的是，在开始刷新序列之前，必须禁用全局中断。如果在刷新写入完成之前发生中断，中断实际上会使刷新序列失效，如图1所示。请记住，在刷新序列完成后，必须恢复全局中断控制状态。

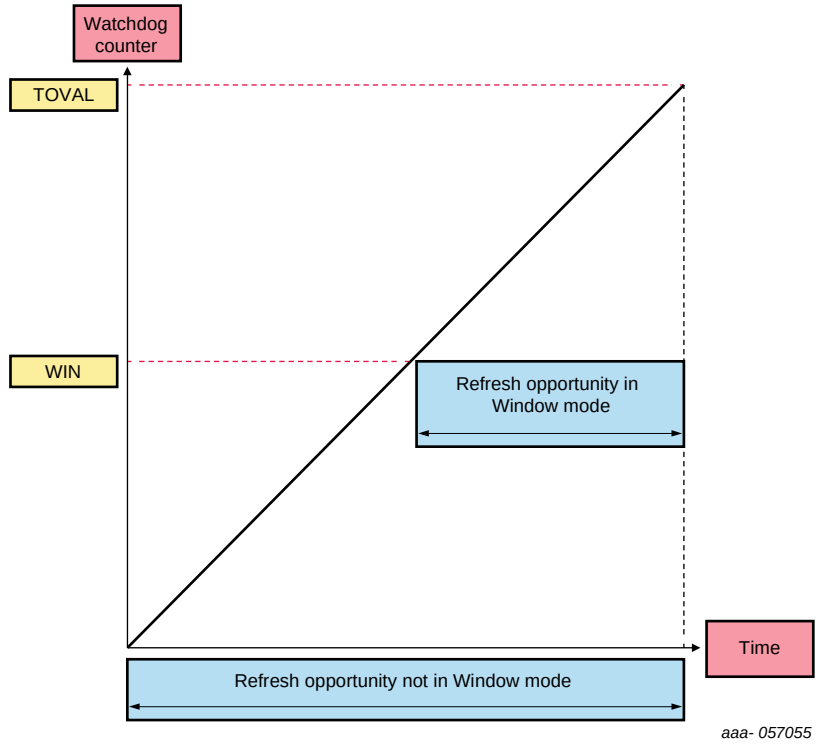

图1. 看门狗计数器的刷新机会

2.1.1 窗口模式下的看门狗刷新

如果软件比预期更快地完成主控制循环的运行，则通常表明出现了问题。如果使用窗口模式，软件必须在WIN寄存器中设置的时间值之后才能开始刷新序列。

2.2 看门狗的快速测试

看门狗具有一项功能，有助于最小化复位后应用程序代码的启动延迟。该功能允许看门狗将计数器拆分为字节级的各个阶段，从而加快了看门狗测试的速度。低字节和高字节独立运行，并且与超时值寄存器的相应字节进行超时测试。

使用此测试功能可将测试时间缩短至512个时钟周期。要进一步加快测试速度，请使用更快的时钟（如总线时钟）作为计数器的参考。

上电复位（POR）时，系统复位寄存器中的POR位被置位，这表明用户必须执行WDOG快速测试。

2.3 配置看门狗

2.3.1 一次性配置看门狗

所有看门狗控制位、超时值和窗口值在复位后都是“一次写入”。这意味着在写入发生后，除非发生复位，否则无法被更改。因此，必须首先配置窗口值和超时值，然后再配置其他控制位。此外，请确保CS[UPDATE]被设置为0。它提供了一个稳健的机制来配置看门狗，并确保在配置后，跑飞的情况不会错误地禁用或修改看门狗配置。

```
    config->enableWdog32         = true;
    config->clockSource          = kWDOG32_ClockSource1;
    config->prescaler            = kWDOG32_ClockPrescalerDivide256;
    config->workMode.enableWait  = false;
    config->workMode.enableStop  = false;
    config->workMode.enableDebug = false;
    config->testMode             = kWDOG32_UserModeEnabled;
    config->enableUpdate         = false;
    config->enableInterrupt      = false;
    config->enableWindowMode     = false;
    config->windowValue          = 0U;
    config->timeoutValue         = 1000;
```

2.3.2 重新配置看门狗

有时，需要在不先强制复位的情况下重新配置或禁用看门狗。

通过在复位后的看门狗初始配置中，将CS[UPDATE]设置为1，就可以通过执行解锁序列随时重新配置看门狗。

相反，如果CS[UPDATE]保持为0，则唯一重新配置看门狗的方法就是发起复位。

解锁序列与刷新序列类似，但使用不同的值。请参见下面以粗体标出的代码：

```
    config->enableWdog32         = true;
    config->clockSource          = kWDOG32_ClockSource1;
    config->prescaler            = kWDOG32_ClockPrescalerDivide256;
    config->workMode.enableWait  = false;
    config->workMode.enableStop  = false;
    config->workMode.enableDebug = false;
    config->testMode             = kWDOG32_UserModeEnabled;
    config->enableUpdate         = true;
    config->enableInterrupt      = false;
    config->enableWindowMode     = false;
    config->windowValue          = 0U;
    config->timeoutValue         = 1000;
```

2.4 在低功耗模式下配置看门狗

请记住，MCXW71中有以下不同的电源域：

- **主域**：CPU和大多数外设都由该电源域供电。
- **唤醒域**：具有从低功耗模式唤醒功能的特殊外设由该电源域供电。

每个电源域都有自己的低功耗模式。通常，这两个电源域都在相同的低功耗模式下工作。但有时，唤醒域必须处于不同的低功耗模式，以启用特定外设，达到MCU唤醒或RAM保持的目的。

为了从低功耗状态下启用看门狗唤醒，主域配置为深度睡眠模式，而唤醒域配置为睡眠模式。这是因为看门狗仅在睡眠模式下工作。

默认情况下，当器件处于等待、停止或调试模式时，看门狗是不工作的。要在这些模式中的任何一种下启用看门狗，必须按照以下章节所述更改配置。

2.4.1 调试模式

以下是一个配置调试模式下的看门狗的示例代码。

```
    config->enableWdog32         = true;
    config->clockSource          = kWDOG32_ClockSource1;
    config->prescaler            = kWDOG32_ClockPrescalerDivide256;
    config->workMode.enableWait  = false;
    config->workMode.enableStop  = false;
    config->workMode.enableDebug = true;
    config->testMode             = kWDOG32_UserModeEnabled;
    config->enableUpdate         = false;
    config->enableInterrupt      = false;
    config->enableWindowMode     = false;
    config->windowValue          = 0U;
    config->timeoutValue         = 1000;
```

2.4.2 等待模式

以下是一个配置等待模式下的看门狗的示例代码。

```
    config->enableWdog32 = true;
    config->clockSource          = kWDOG32_ClockSource1;
    config->prescaler            = kWDOG32_ClockPrescalerDivide256;
    config->workMode.enableWait  = true;
    config->workMode.enableStop  = false;
    config->workMode.enableDebug = false;
    config->testMode             = kWDOG32_UserModeEnabled;
    config->enableUpdate         = true;
    config->enableInterrupt      = false;
    config->enableWindowMode     = false;
    config->windowValue          = 0U;
    config->timeoutValue         = 1000;
```

2.4.3 停止模式

以下是一个配置停止模式下的看门狗的示例代码。

```
    config->enableWdog32         = true;
```

```
    config->clockSource           = kWDOG32_ClockSource1;
    config->prescaler             = kWDOG32_ClockPrescalerDivide256;
    config->workMode.enableWait   = true;
    config->workMode.enableStop   = true;
    config->workMode.enableDebug = false;
    config->testMode              = kWDOG32_UserModeEnabled;
    config->enableUpdate          = true;
    config->enableInterrupt       = false;
    config->enableWindowMode      = false;
    config->windowValue           = 0U;
    config->timeoutValue          = 1000;
```

3 时钟

看门狗计数器的时钟源选项可通过编程CS[CLK]来选择：

- SLOW_CLOCK
- 32K_CLK
- FRO-6M
- OSC-RF

这些时钟源选项允许软件为必须满足更严格安全要求的应用而选择独立于总线时钟的时钟源。使用总线时钟以外的时钟源可确保即使总线时钟因某种原因停止，看门狗计数器也能继续运行。适用于所有时钟源的一个可选固定预分频器允许更长的超时时间。当设置CS[PRES]时，时钟源会在为看门狗计数器提供时钟之前进行256预分频。

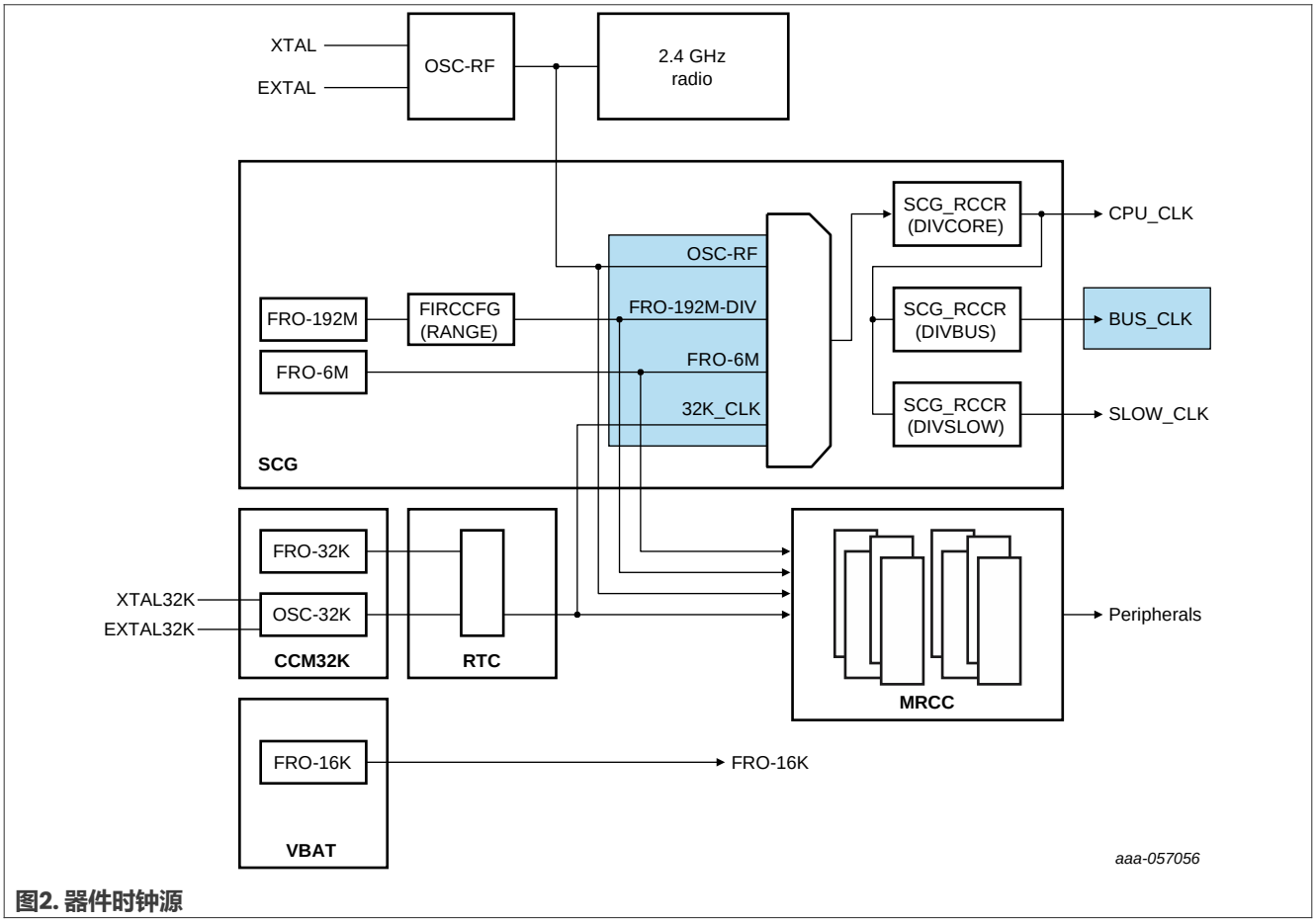

图2. 器件时钟源

4 复位和中断

除复位事件外，看门狗模块还可以生成一个系统中断事件，以便在复位事件发生之前进行快速诊断。

- 当启用中断时（CS[INT] = 1）：在触发复位的事件发生后，看门狗首先会生成一个中断请求。接着，看门狗会延迟128个总线时钟，然后再强制复位，以便中断服务例程（ISR）执行任务。
- 当禁用中断时（CS[INT] = 0）：看门狗在强制复位前不会延迟。

```
    config->enableWdog32         = true;
    config->clockSource          = kWDOG32_ClockSource1;
    config->prescaler            = kWDOG32_ClockPrescalerDivide256;
    config->workMode.enableWait  = false;
    config->workMode.enableStop  = false;
    config->workMode.enableDebug = false;
    config->testMode             = kWDOG32_UserModeEnabled;
    config->enableUpdate         = true;
    config->enableInterrupt      = true;
    config->enableWindowMode     = false;
    config->windowValue          = 0U;
    config->timeoutValue         = 1000;
```

5 在低功耗模式下启用连接协议栈中的WDOG

本节介绍了完整的实施看门狗-连接协议栈集成演示工程的步骤。

5.1 前提条件

本文档随附一个在低功耗模式下使用看门狗的功能演示。该示例基于低功耗参考设计工程。该工程包含在MCXW71 SDK软件包中，并在MCUXpresso IDE平台上开发。

要完成看门狗-连接协议栈集成演示工程的实施，需要以下前提条件：

- MCUXpresso SDK Builder v11.10.0或更高版本
- FRDM-MCXW71 SDK v2.16.00
- 低功耗参考设计演示包
- FRDM-MCXW71开发板

5.2 下载和安装软件开发工具包

本节提供了为使用FRDM-MCXW71下载SDK所需的步骤，作为起点。

1. 访问MCUXpresso网站。
2. 点击**Select Development Board**。
3. 使用已注册的账户登录。
4. 在**Search for Hardware**字段中搜索“FRDM-MCXW71”。
5. 选择建议的开发板，并点击**Build SDK**。

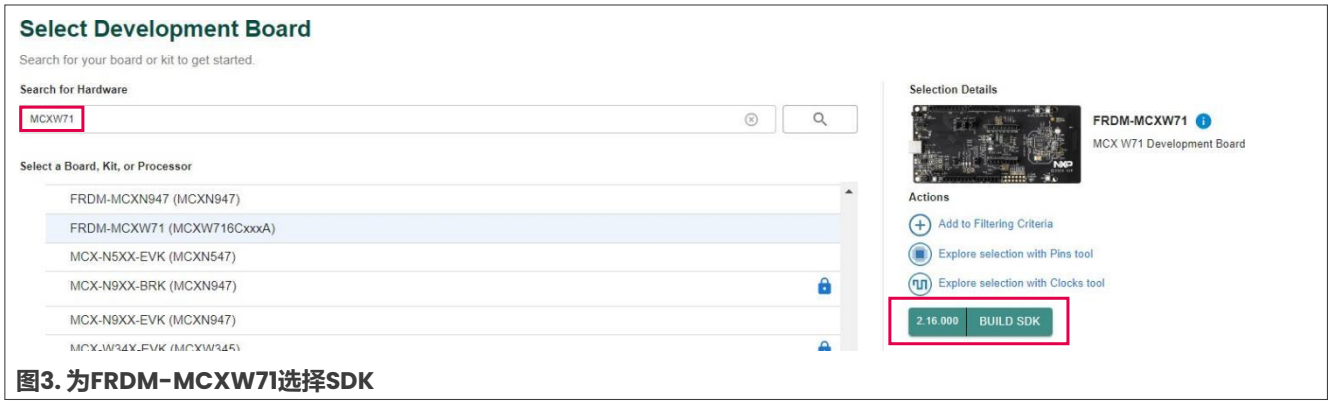

图3. 为FRDM-MCXW71选择SDK

6. 在Toolchain/IDE组合框中选择“MCUXpresso IDE”。选择支持的操作系统。点击**Build SDK**，系统需要几分钟时间将软件包发送到您在MCUXpresso网页上的账户中。阅读并接受许可协议。SDK会自动开始在您的电脑上下载。

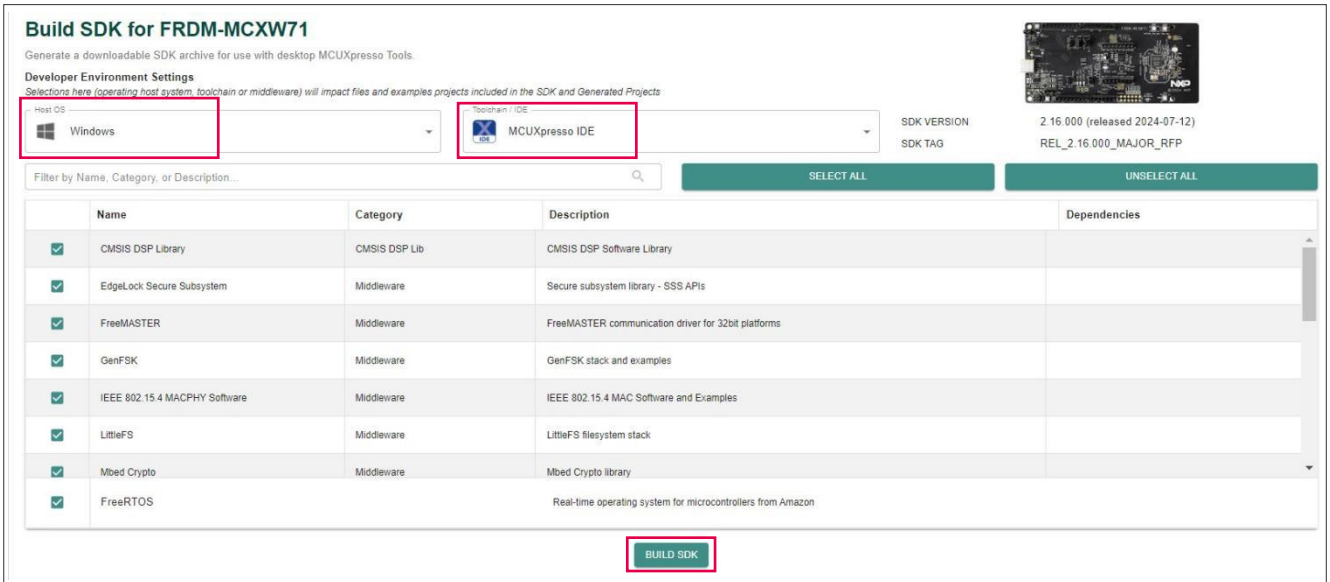

图4. 为FRDM-MCXW71编译SDK

7. 在MCUXpresso SDK控制面板上，点击所需SDK构建的**Download**。SDK会开始在您的电脑上下载。

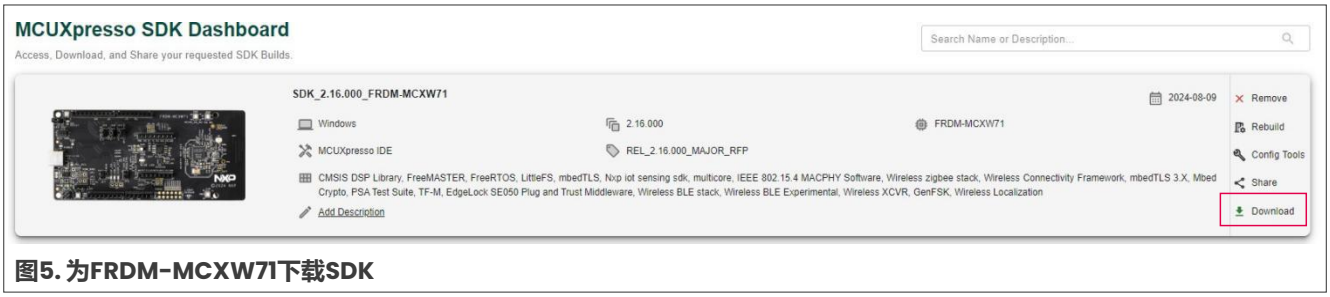

图5. 为FRDM-MCXW71下载SDK

8. 将FRDM-MCXW71的SDK压缩包文件夹拖放到**Installed SDKs**列表中。

图6. FRDM-MCXW71的SDK压缩包含在Installed SDKs列表中

现在，已经下载并安装了FRDM-MCXW71开发板的SDK包。

5.3 源文件中的主要修改

本节介绍了需要在源文件中进行的修改，以便将看门狗驱动程序文件包含在工程中。

注：*所使用的演示工程默认不包含看门狗驱动程序文件，必须将这些文件添加到工程中。*

在自定义工程中包含看门狗驱动程序文件后，下一步就是添加正确的配置，以便在低功耗模式下启用看门狗。下文将解释需要注意的主要方面。

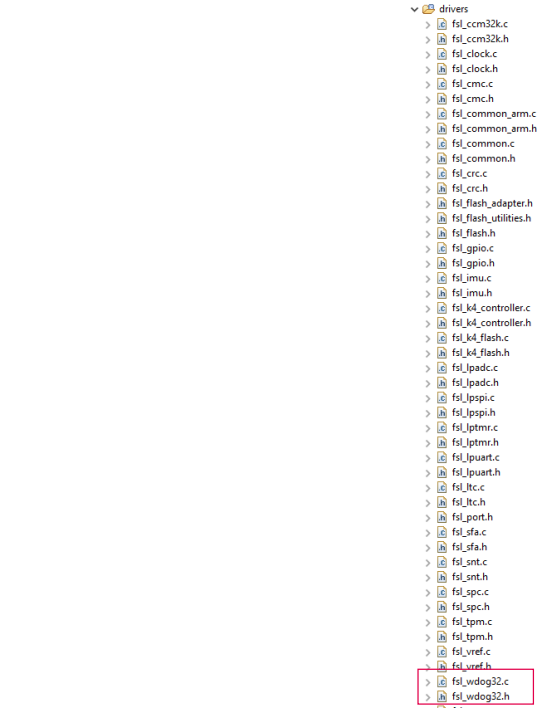

图7. 在自定义工程中包含看门狗驱动器文件

5.3.1 app_preinclude.h文件

要配置看门狗定时器，请执行以下步骤：

1. 打开位于`sources`文件夹中的`app_preinclude.h`文件，并查找启用或控制所需低功耗模式的宏。

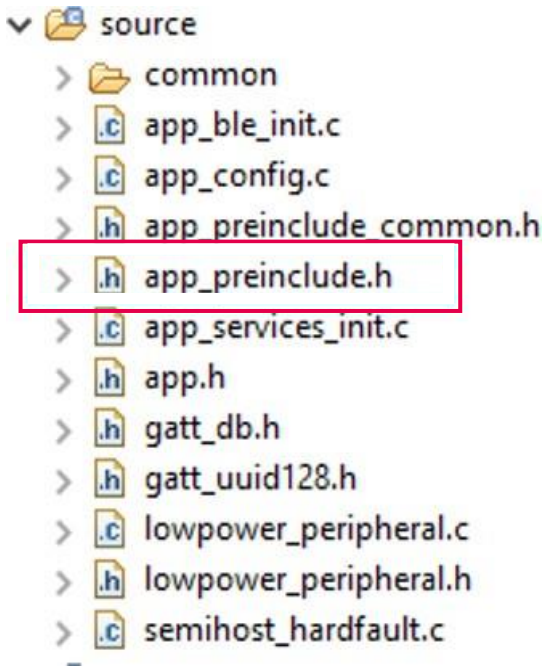

图8. app_preinclude.h文件

2. 要进入深度睡眠模式，请将gAppLowPowerConstraintInNoBleActivity_c变量更新为2。
3. 为各种低功耗蓝牙状态编辑低功耗约束设置。

```
#define gAppLowPowerConstraintInAdvertising_c       2
/* Scanning not supported on peripheral */
//#define gAppLowPowerConstraintInScanning_c        2
#define gAppLowPowerConstraintInConnected_c         2
#define gAppLowPowerConstraintInNoBleActivity_c     2
#endif
```

图9. app_preinclude.h文件

5.3.2 main.c文件

要配置WDOG，请执行以下步骤：

1. 在main.c文件中添加声明和变量，如图11中的代码高亮部分所示。

```
#include "fsl_wdog32.h"

/*******************************************************************************
 * Definitions WDOG32
 ******************************************************************************/
#define APP_SKIP_LOW_BYTE_TEST 0

#define RESET_CHECK_FLAG       (*((uint32_t *)0x20014000))
#define RESET_CHECK_INIT_VALUE 0x0D0D
#define EXAMPLE_WDOG_BASE      WDOG0
#define WDOG_IRQHandler        WDOG0_IRQHandler
#ifndef APP_WDOG_RESET_FLAG_SET
#if defined(FSL_FEATURE_SOC_RCM_COUNT) && (FSL_FEATURE_SOC_RCM_COUNT)
#include "fsl_rcm.h"
#elif defined(FSL_FEATURE_SOC_SMC_COUNT) && (FSL_FEATURE_SOC_SMC_COUNT > 1) /* MSMC */
#include "fsl_msmc.h"
#elif defined(FSL_FEATURE_SOC_ASMC_COUNT) && (FSL_FEATURE_SOC_ASMC_COUNT) /* ASMC */
#include "fsl_asmc.h"
#elif defined(FSL_FEATURE_SOC_CMC_COUNT) && (FSL_FEATURE_SOC_CMC_COUNT) /* CMC */
#include "fsl_cmc.h"
#endif
#endif


static WDOG_Type *wdog32_base = EXAMPLE_WDOG_BASE;
#if defined(FSL_FEATURE_SOC_RCM_COUNT) && (FSL_FEATURE_SOC_RCM_COUNT)
static RCM_Type *rcm_base = RCM;
#endif
AT_QUICKACCESS_SECTION_DATA(static wdog32_config_t config);
```

图10. 在main.c文件中声明WDOG变量

2. 如果应用程序需要调用看门狗，则创建并声明配置WDOG和中断的函数。

```
#if !(defined(FSL_FEATURE_SOC_ASMC_COUNT) && (FSL_FEATURE_SOC_ASMC_COUNT))
/*!
 * @brief WDOG0 IRQ handler.
 *
 */
void WDOG_IRQHandler(void)
{
    WDOG32_ClearStatusFlags(wdog32_base, kWDOG32_InterruptFlag);

    RESET_CHECK_FLAG++;
    SDK_ISR_EXIT_BARRIER;
}
#endif /* FSL_FEATURE_SOC_ASMC_COUNT */


void INIT_wd32(void)
{

    WDOG32_GetDefaultConfig(&config);

    config.testMode = kWDOG32_UserModeEnabled;

    config.clockSource  = kWDOG32_ClockSource1;
    config.prescaler    = kWDOG32_ClockPrescalerDivide256;
    config.windowValue  = 0U;
    config.timeoutValue = 200U;
    config.enableInterrupt = false;

    /* Refresh test in none-window mode */
    config.enableWindowMode = false;
    config.enableWdog32     = true;

    WDOG32_Init(wdog32_base, &config);
}
```

图11. 配置WDOG和中断

3. WDOG的使用方式必须确保计数在广播开始时启动。必须能够监控软件流程，并防止无限循环或其他类型的故障。因此，必须在低功耗功能开始之前在主程序中调用此函数。

```
#if defined(gAppLowpowerEnabled_d) && (gAppLowpowerEnabled_d > 0)

            INIT_wd32();
            (void)PWR_EnterLowPower(0U);
#else
            __WFI();
#endif
        }

        OSA_EnableIRQGlobal();
    }
```

图12. 在低功耗功能之前在主程序中调用看门狗函数

5.3.3 主域和唤醒域

看门狗模块与唤醒域相连。为了使看门狗能够从低功耗状态唤醒器件，必须将唤醒域置于睡眠模式。主域可进一步降至深度睡眠模式，以节省更多电量。

要配置这种场景，请编辑`fsl_pm_board`文件并修改每个域的睡眠模式，如下所示：

```
.main_domain = kCMC_DeepSleepMode,
.wake_domain = kMC_SleepMode,
```

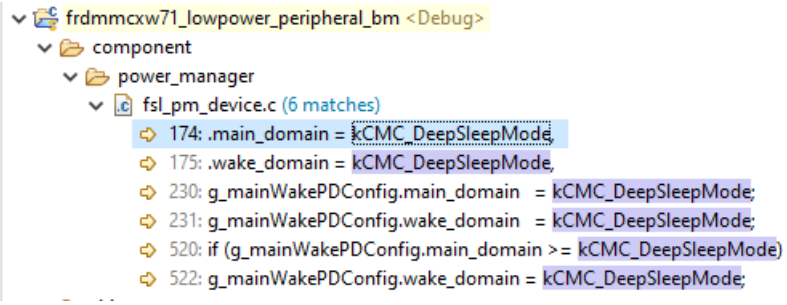

图13. 启用唤醒域

6 WDOG功能

图14所示为广播功能。观察广播期间的快速峰值。在WDOG复位MCU后，第一个峰值显示复位事件，然后在广播再次开始之前进行重新配置。

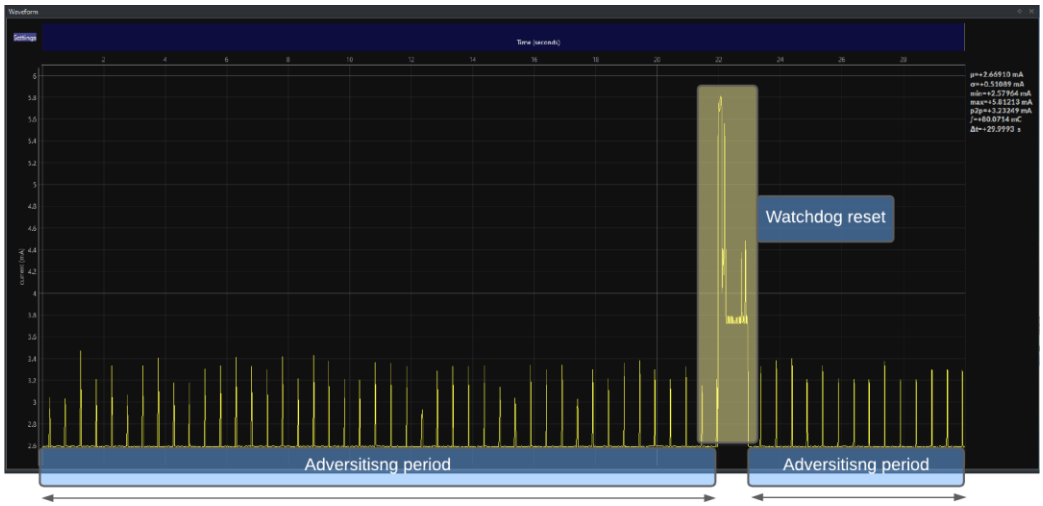

图14. 由WDOG执行的复位

7 参考资料

表1列出了一些其他文档和资源，可供参考以获取更多信息。其中有些文档只有在签订了保密协议（NDA）的情况下才能访问。要获取此类文件，请联系当地的恩智浦现场应用工程师（FAE）或销售代表。

表1. 参考资料

文档	链接/如何获取
MCXW71参考手册	请联系恩智浦FAE/销售代表
MCXW71产品系列数据手册	
FRDM-MCXW71开发板用户手册	

8 修订历史

表2汇总了本文的修订情况。

表2. 修订历史

文档ID	发布日期	说明
AN14416 v.1.0	2024年9月10日	首次公开发布

9 关于本文中源代码的说明

本文中所示的示例代码具有以下版权和BSD-3-Clause许可：

2024年恩智浦版权所有；在满足以下条件的情况下，可以源代码和二进制文件的形式重新分发和使用本源代码（无论是否经过修改）：

1. 重新分发源代码必须保留上述版权声明、这些条件和以下免责声明。
2. 以二进制文件形式重新分发时，必须在文档和/或随分发提供的其他材料中复制上述版权声明、这些条件和以下免责声明。
3. 未经事先书面许可，不得使用版权所有者的姓名或参与者的姓名为本软件的衍生产品进行背书或推广。

本软件由版权所有者和参与者“按原样”提供，不承担任何明示或暗示的担保责任，包括但不限于对适销性和特定用途适用性的暗示保证。在任何情况下，无论因何种原因或根据何种法律条例，版权所有者或参与者均不对因使用本软件而导致的任何直接、间接、偶然、特殊、惩戒性或后果性损害（包括但不限于采购替代商品或服务；使用损失、数据损失或利润损失或业务中断）承担责任，无论是因合同、严格责任还是侵权行为（包括疏忽或其他原因）造成的，即使事先被告知有此类损害的可能性也不例外。

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com.cn/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

目录