AN14120

在i.MX 8M、i.MX 8ULP和i.MX 93上使用VS Code调试Cortex-M

第2.0版—2024年3月13日 **应用笔记**

文档信息

信息	内容
关键词	AN14120、i.MX 8M、i.MX 93、Cortex-M、i.MX 8MN、i.MX 8MP、i.MX 8MM、VS Code、MCUXSDK、MCUXpresso SDK、J-Link、SEGGER、Cortex-M调试
摘要	本文介绍了如何使用微软Visual Studio Code，在i.MX 8M系列、i.MX 8ULP和i.MX 93 Cortex-M处理器上对应用进行交叉编译、部署和调试。

1 介绍

本文介绍了如何使用微软Visual Studio Code，在i.MX 8M系列、i.MX 8ULP和i.MX 93 Cortex-M处理器上对应用进行交叉编译、部署和调试。

1.1 软件环境

该解决方案可以在Linux和Windows两种操作系统的主机上进行安装和运行。本文假定使用Windows PC，但这并非强制性要求。

本文使用Linux BSP版本6.1.22_2.0.0，并使用了以下预构建镜像：

- **i.MX 8M Mini**：`imx-image-full-imx8mmevk.wic`
- **i.MX 8M Nano**：`imx-image-full-imx8mnevk.wic`
- **i.MX 8M Plus**：`imx-image-full-imx8mpevk.wic`
- **i.MX 8ULP**：`imx-image-full-imx8ulpevk.wic`
- **i.MX 93**：`imx-image-full-imx93evk.wic`

有关如何构建这些镜像的详细步骤，请参阅《i.MX Linux用户指南》（文档IMXLUG）和《i.MX Yocto工程用户指南》（文档IMXLXYOCTOUG）。

如果使用Windows PC，请使用Win32 Disk Imager（https://win32diskimager.org/）或Balena Etcher（https://etcher.balena.io/）将预构建镜像写入SD卡。

如果使用的是Ubuntu PC，则使用下面的命令将预构建镜像写入SD卡：

```
$  sudo dd if=<image_name>.wic of=/dev/sd<x> bs=1M status=progress conv=fsync
```

***注：**请检查读卡器分区，并将`sd<x>`替换为相应的分区。*

1.2 硬件设置和设备

- 开发套件：
  - 恩智浦i.MX 8MM EVK LPDDR4
  - 恩智浦i.MX 8MN EVK LPDDR4
  - 恩智浦i.MX 8MP EVK LPDDR4
  - 恩智浦i.MX 93 EVK for 11x11 mm LPDDR4
  - 恩智浦i.MX 8ULP EVK LPDDR4
- Micro SD卡：SanDisk Ultra 32-GB Micro SDHC I Class 10用于当前的实验。
- 用于调试端口的Micro USB（i.MX 8M）或Type-C（i.MX 93）线缆。
- SEGGER J-Link硬件调试器。

2 前提条件

在开始调试之前，必须满足几个前提条件才能获得正确配置的调试环境。

2.1 主机 – i.MX板调试连接

要建立硬件调试连接，请执行以下步骤：

1. 使用USB线缆通过DEBUG USB-UART和PC USB连接器将i.MX电路板连接到主机。Windows操作系统会自动查找串行设备。
2. 在“设备管理器”中，在“端口”（COM和LPT）下找到两个或4个连接的USB串行端口（COM<port_number>）。其中一个端口用于Cortex-A核生成的调试消息，另一个端口用于Cortex-M核。

   在确定所需的正确端口之前，需谨记以下几点：
   - [**i.MX 8MP、i.MX 8ULP、i.MX93**]：“设备管理器”中有4个可用端口。最后一个端口用于Cortex-M调试，倒数第二个端口用于Cortex-A调试，调试端口按升序计数。
   - [**i.MX 8MM、i.MX 8MN**]：“设备管理器”中有两个可用端口。第一个端口用于Cortex-M调试，第二个端口用于Cortex-A调试，调试端口按升序计数。
3. 使用您喜欢的串行终端仿真器（例如PuTTY），通过设置以下参数打开正确的调试端口：
   - Speed to 115200 bps
   - 8 data bits
   - 1 stop bit (115200, 8N1)
   - No parity
4. 将SEGGER硬件调试器的USB连接到主机，然后将SEGGER的JTAG连接器连接到i.MX电路板的JTAG接口。

如果i.MX电路板的JTAG接口没有引导连接器，则将红色线与脚1对齐来确定方向，如图1所示。

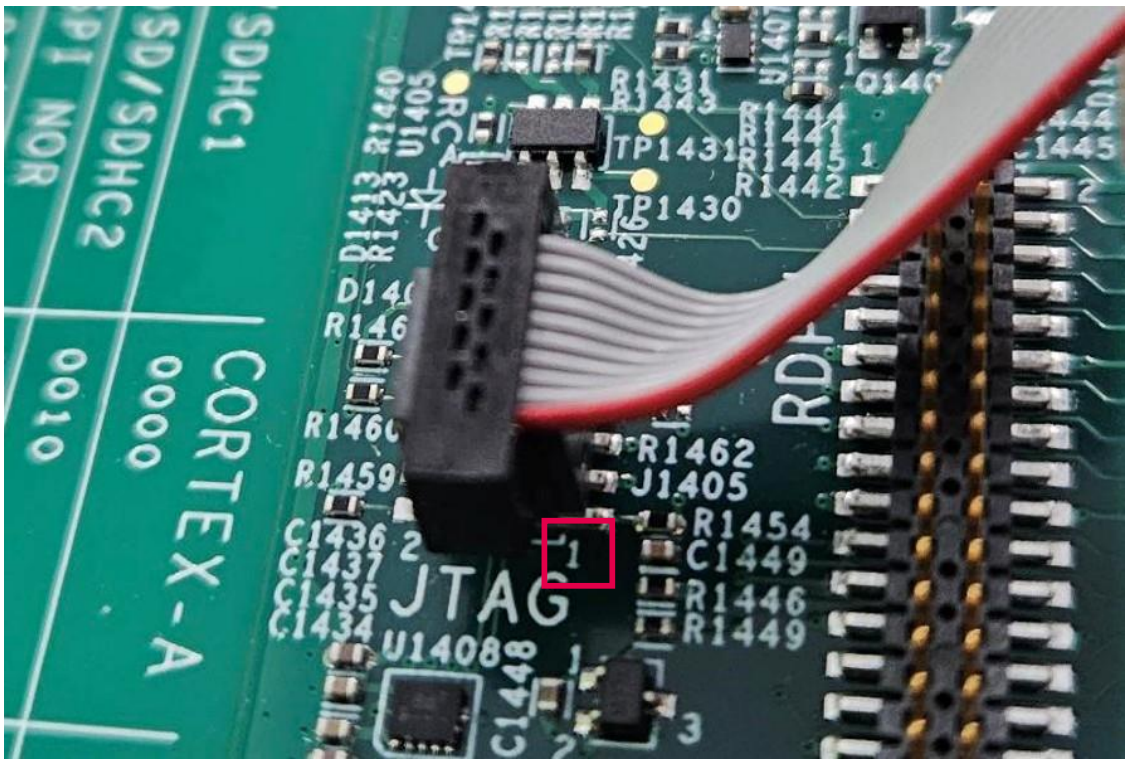

图1. JTAG连接

2.2 VS Code配置

要下载和配置VS Code，请执行以下步骤：

1. 从官网下载并安装最新版本的微软Visual Studio Code。如果使用Windows作为主机操作系统，请在Visual Studio Code主页选择“Download for Windows”（下载Windows版）按钮。

图2. 下载微软Visual Studio Code

2. 安装Visual Studio Code后，打开它并选择“扩展”选项卡，或按Ctrl+Shift+X组合键。

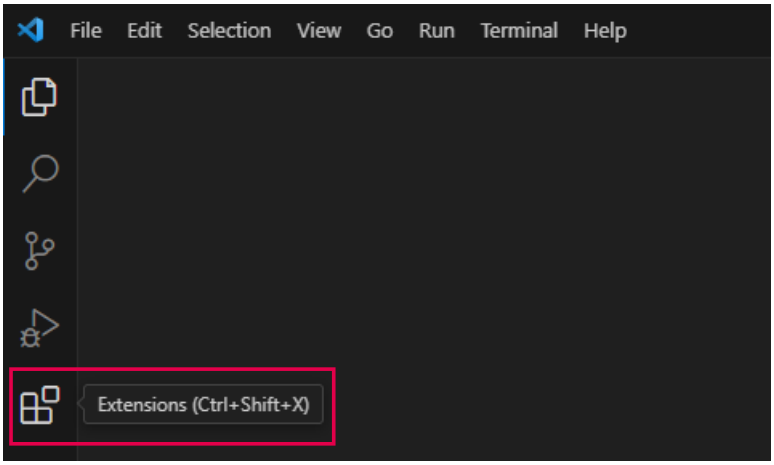

图3. 微软Visual Studio Code扩展选项卡

3. 在专用搜索栏中，输入“*MCUXpresso for VS Code*”，然后安装扩展。VS Code窗口的左侧会出现一个新选项卡。

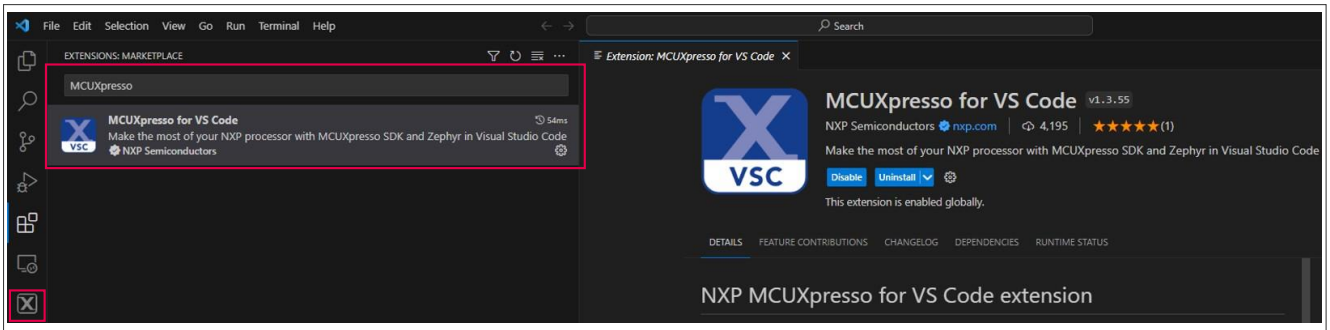

图4. 安装MCUXpresso for VS Code扩展

2.3 MCUXpresso扩展配置

要配置MCUXpresso扩展，请执行以下步骤：

1. 单击左侧栏中的MCUXpresso扩展专用选项卡。在“快速启动面板”中，单击“打开MCUXpresso安装程序”，并授予安装程序下载权限。
2. 很快会出现安装程序窗口。单击“MCUXpresso SDK Developer”，在SEGGER J-Link上单击“安装”按钮。安装程序安装存档文件、工具链、Python支持、Git和硬件调试器所需的软件。

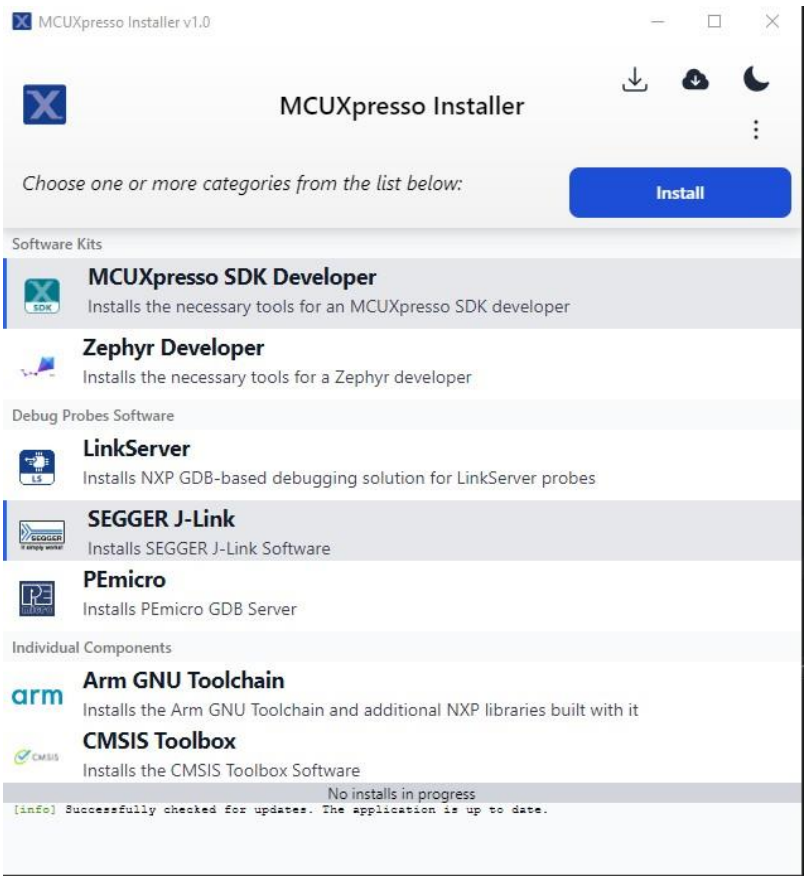

图5. 安装SEGGER J-Link软件

安装完所有软件包后，确定J-Link调试器已连接到主机。然后，在MCUXpresso扩展中查看“硬件调试器”视图下是否也能看到这个调试器，如图6所示。

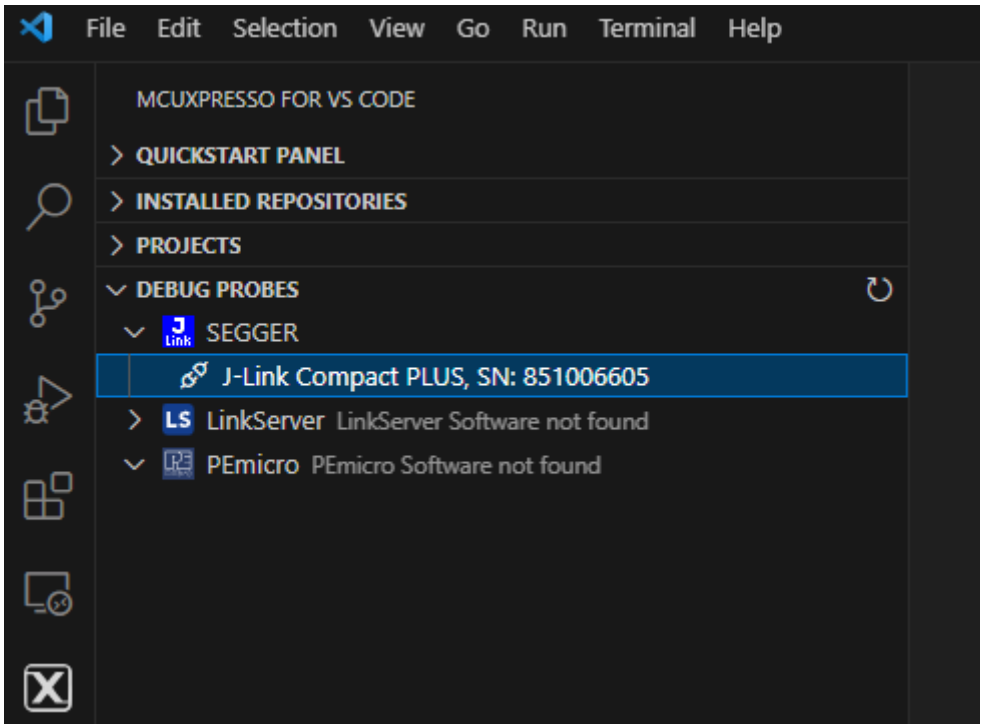

图6. “硬件调试器”视图下能看到J-Link调试器

2.4 导入MCUXpresso SDK

根据正在运行的电路板，从恩智浦官网构建并下载特定的SDK。对于本文，以下SDK已经过了测试：

- SDK_2_15_000_EVK-MIMX8MM
- SDK_2_15_000_EVK-MIMX8MN
- SDK_2_15_000_EVK-MIMX8MP
- SDK_2.14.0_EVK-MIMX8ULP
- SDK_2_15_000_MCIMX93-EVK

要构建i.MX 93 EVK的示例，请参见图7：

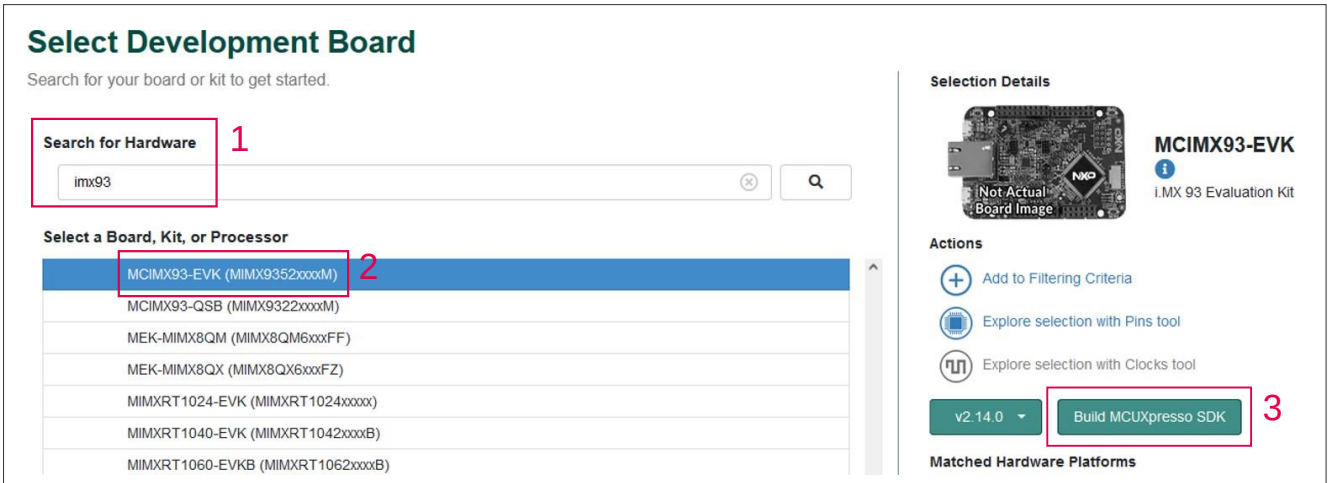

图7. 下载MCUXpresso SDK

要将MCUXpresso SDK存储库导入VS Code，请执行以下步骤：

1. 下载了SDK后，打开Visual Studio Code。单击左侧的MCUXpresso选项卡，然后展开“已安装存储库”和“工程”视图。
2. 单击“导入存储库”，然后选择“本地存档”。单击“存档”字段对应的“浏览”，然后选择新下载的SDK存档文件。
3. 选择解压缩存档文件的路径，并填入“位置”字段。
4. “名称”字段可以保留默认的名称，也可以选择自定义名称。
5. 根据需要勾选或取消勾选“创建Git存储库”，然后单击“导入”。

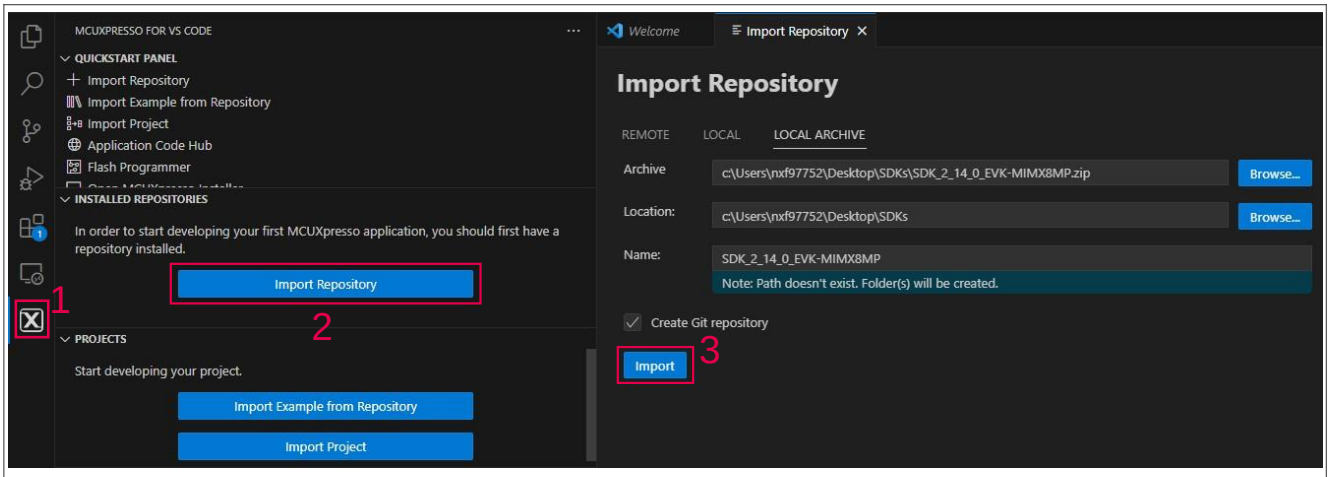

图8. 导入MCUXpresso SDK存储库

2.5 导入示例应用

导入SDK时，它会在“已安装存储库”视图中显示。

要从SDK存储库导入示例应用，请执行以下步骤：

1. 单击“工程”视图中的“从存储库导入示例”按钮。
2. 从下拉列表中选择一个存储库。
3. 从下拉列表中选择工具链。
4. 选择目标电路板。

5. 从“选择模板”列表中选择demo_apps/hello_world示例。
6. 为工程选择一个名称（可以使用默认名称），并将路径设置为工程的“位置”。
7. 单击“创建”。
8. 对于旧版本的SDK（v5.15.000以下），无法正确解析设备名称。仅对i.MX 8M系列执行以下步骤。在“工程”视图下，展开导入的工程。进入“设置”部分，然后单击mcuxpresso-tools.json文件。
   a. 在“debug”>“segger”下添加“interface”：“JTAG”
   b. 对于i.MX 8MM，在“debug”>“segger”下添加以下配置：“device”：“MIMX8MM6_M4”
   c. 对于i.MX 8MN，在“debug”>“segger”下添加以下配置：“device”：“MIMX8MN6_M7”
   d. 对于i.MX 8MP，在“debug”>“segger”下添加以下配置：“device”：“MIMX8ML8_M7”

以下代码所示为对mcuxpresso-tools.json进行上述更改后i.MX8MP“调试”部分的示例：

```
"debug": {
    "linkserver": {},
    "pemicro": {},
    "segger": {
      "device": "MIMX8ML8_M7",
      "interface": "JTAG"
      },
  },
```

成功导入该示例应用后，在“工程”视图中能够看到它。此外，在“资源管理器”（Ctrl+Shift+E）选项卡中能够看到工程源文件。

3 构建应用

要构建应用，请按下左侧的“*Build Selected*”（构建已选）图标，如图9所示。

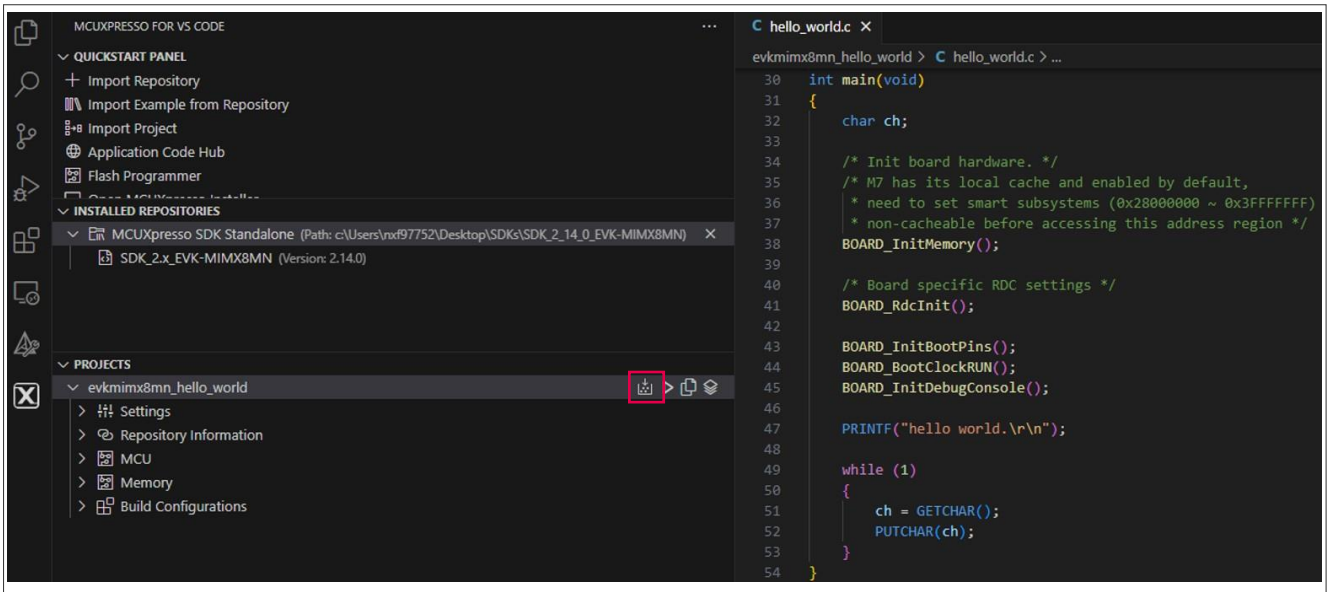

图9. 构建应用

4 为调试器准备电路板

要使用JTAG调试Cortex-M应用，根据不同的平台，需要满足几个前提条件：

1. **对于i.MX 93**
   要支持i.MX 93，必须安装SEGGER J-Link的补丁：SDK_MX93_3RDPARTY_PATCH.zip。
   注：*即使之前安装过，也必须安装该补丁。*
   下载完成后，解压缩存档文件并将*Devices*目录和`JLinkDevices.xml`文件复制到`C:\Program Files\SEGGER\JLink`。如果使用Linux PC，则目标路径为`/opt/SEGGER/JLink`。

   - **在仅运行Cortex-M33的情况下调试Cortex-M33**
     在该模式下，需要将启动模式开关**SW1301[3:0]**设置为**[1010]**。然后可以使用"调试"按钮直接加载和调试M33镜像。有关更多详细信息，请参阅第5节。

   如果需要在Cortex-A55上运行Linux的同时调试Cortex-M33，可以通过两种方式进行调试：

   - **当Cortex-A55处于U-Boot时调试Cortex-M33**
     首先，将第3节中生成的`sdk20-app.bin`文件（位于`armgcc/debug`目录中）复制到SD卡的启动分区中。启动电路板并将其停止在U-Boot中。当启动开关配置为启动Cortex-A时，启动序列不会启动Cortex-M。必须使用以下命令手动启动。如果Cortex-M没有启动，JLink就无法连接到内核。

     ```
     u-boot=> fatload mmc 1:1 80000000 sdk20-app.bin
     u-boot=> cp.b 0x80000000 0x201e0000 0x10000
     u-boot=> bootaux 0x1ffe0000 0
     ```

     注：*如果系统无法正常调试，请尝试右键单击MCUXpresso for VS Code中的工程，然后选择"Attach to debug the project"（连接，调试工程）。*

   - **当Cortex-A55在Linux中时调试Cortex-M33**
     必须更改内核DTS以禁用UART5，UART5使用与JTAG接口相同的引脚。如果使用Windows PC，最简单的方法是安装WSL+Ubuntu 22.04 LTS，然后交叉编译DTS。
     安装完WSL+Ubuntu 22.04 LTS后，打开在WSL上运行的Ubuntu设备，安装所需的软件包：

     ```
     $ sudo apt update
     $ sudo apt install build-essential flex bison gcc-aarch64-linux-gnu git
     ```

     现在，可以下载内核源代码：

     ```
     $ git clone https://github.com/nxp-imx/linux-imx
     $ cd linux-imx
     $ git checkout lf-6.1.22-2.0.0
     ```

     要禁用UART5外设，请在`linux-imx/arch/arm64/boot/dts/freescale/imx93-11x11-evk.dts`文件中搜索`lpuart5`节点，并将`okay`状态替换为`disabled`：

     ```
     &lpuart5 {
            /* BT */
            pinctrl-names = "default";
            pinctrl-assert-gpios = <&pcal6524 19 GPIO_ACTIVE_HIGH>;
            pinctrl-0 = <&pinctrl_uart5>;
            status = "disabled";

            bluetooth {
                    compatible = "nxp,88w8987-bt";
            };
     ```

```
};
```

重新编译DTS：

```
$ ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- make freescale/imx93-11x11-
evk.dtb
```

将新创建的`linux-imx/arch/arm64/boot/dts/freescale/imx93-11x11-evk.dtb`文件复制到SD卡的启动分区。

将第3节生成的`hello_world.elf`文件（位于`armgcc/debug`目录中）复制到SD卡的启动分区中。

在Linux中启动电路板。由于Cortex-A启动时，启动ROM不会启动Cortex-M，必须手动启动Cortex-M。

```
root@imx8mm-lpddr4-evk:/lib/firmware# cp /run/media/mmcblk2p1/
hello_world.elf /lib/firmware
root@imx93evk:~# echo hello_world.elf > /sys/class/remoteproc/remoteproc0/
firmware
root@imx93evk:~# echo start > /sys/class/remoteproc/remoteproc0/state
```

***注：** `hello_world.elf`文件必须放在`/lib/firmware`目录中。*

2. **对于i.MX 8M**

   要支持i.MX 8M Plus，必须安装SEGGER J-Link的补丁：

   `iar_segger_support_patch_imx8mp.zip`。

   下载完成后，解压缩归档文件，并将*Devices*目录和`JLinkDevices.xml`文件从*JLink*目录复制到`C:\Program Files\SEGGER\JLink`。如果使用Linux PC，则目标路径为`/opt/SEGGER/JLink`。

   - **当Cortex-A处于U-Boot时调试Cortex-M**

     在这种情况下，无需进行任何特别的操作。在U-Boot中启动电路板，然后跳到第5节。

   - **当Cortex-A运行Linux时调试Cortex-M**

     要在Cortex-A运行Linux的同时运行和调试Cortex-M应用，必须为Cortex-M分配并保留特定的时钟。这些操作在U-Boot中完成。

     在U-Boot中停止电路板并运行以下命令：

     ```
     u-boot=> run prepare_mcore
     u-boot=> boot
     ```

3. **对于i.MX 8ULP**

   要支持i.MX 8ULP，必须安装SEGGER J-Link的补丁：

   `SDK_MX8ULP_3RDPARTY_PATCH.zip`。

   ***注：**即使之前安装过，也必须安装该补丁。*

   下载完成后，解压缩存档文件并将*Devices*目录和`JLinkDevices.xml`文件复制到`C:\Program Files\SEGGER\JLink`。如果使用Linux PC，则目标路径为`/opt/SEGGER/JLink`。

   对于i.MX 8ULP，由于Uppower单元的存在，需要先在“VSCode”存储库中使用`m33_image`构建`flash.bin`。M33镜像可在`{CURRENT REPO}\armgcc\debug\sdk20-app.bin`中找到。关于如何构建`flash.bin`镜像，请参阅`SDK_2_xx_x_EVK-MIMX8ULP/docs`中《面向EVK-MIMX8ULP和EVK9-MIMX8ULP的MCUXpresso SDK快速入门》的第6节。

   ***注：**使用活动VSCode存储库中M33镜像。否则，程序无法正确连接。*

   右键单击并选择“连接”。

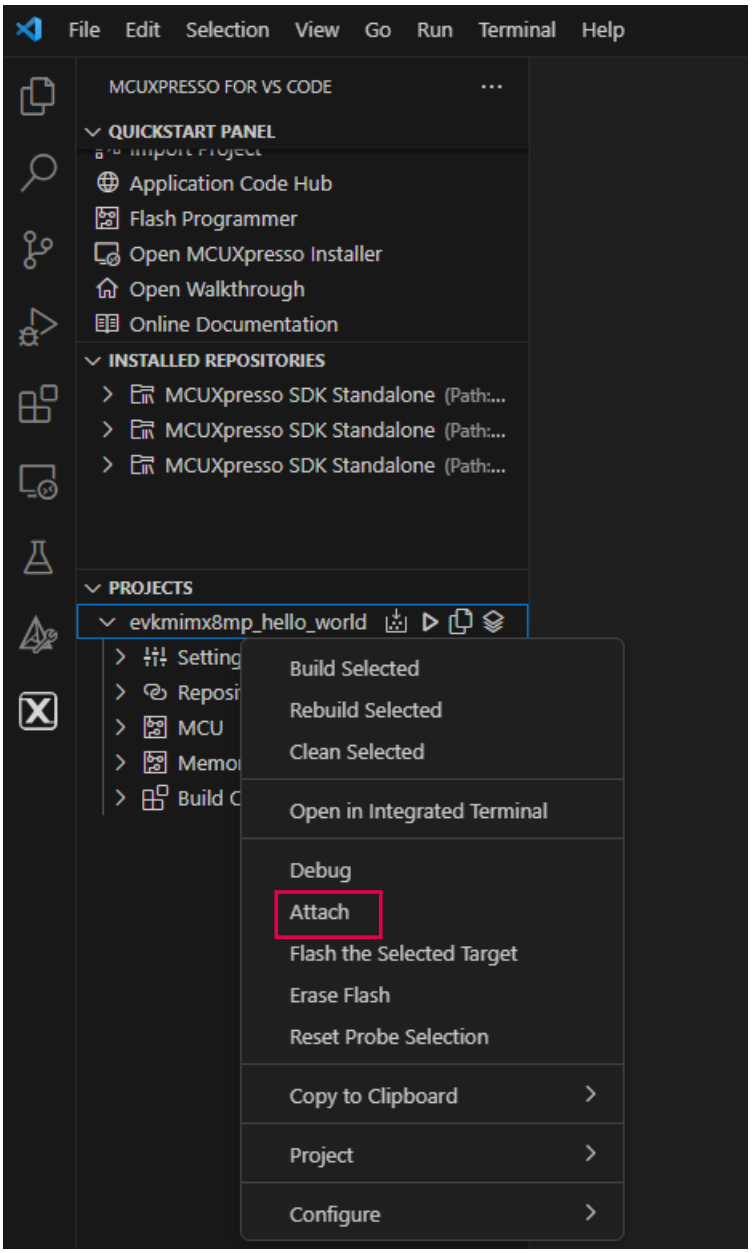

图10. 为i.MX 8ULP连接J-Link调试器

5 运行与调试

按下调试按钮后，选择“调试工程配置”，调试会话开始。

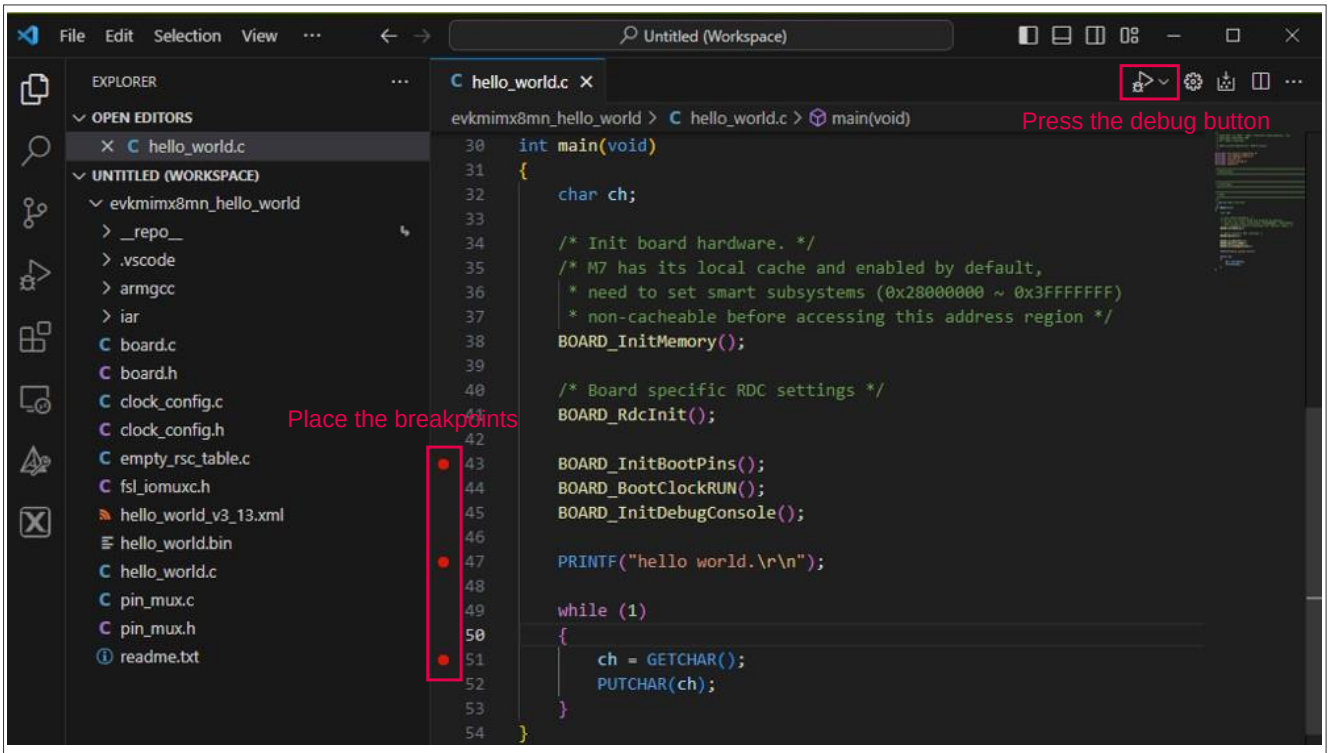

图11. 启动调试会话

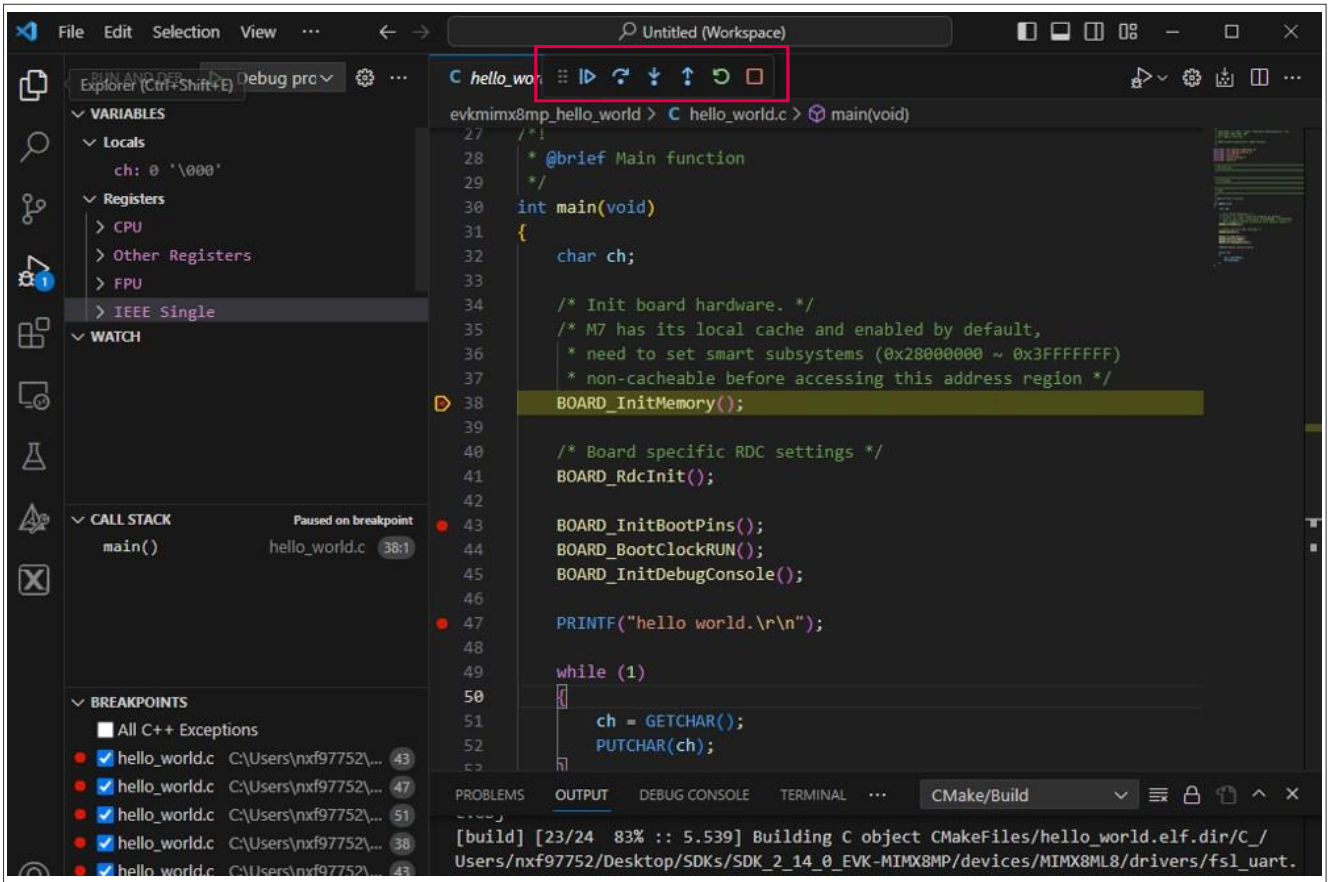

图12. 调试菜单

调试会话开始时，会显示一个专用菜单。该调试菜单中有一些按钮，用于启动执行直到断点触发、暂停执行、单步执行、进入函数、跳出函数、重新启动和停止调试。

此外，我们可以在左侧的导航器中查看局部变量、寄存器值、观察某些表达式，以及检查调用协议栈和断点。这些功能区域位于“运行与调试”选项卡上，而不是MCUXpresso for VS Code中。

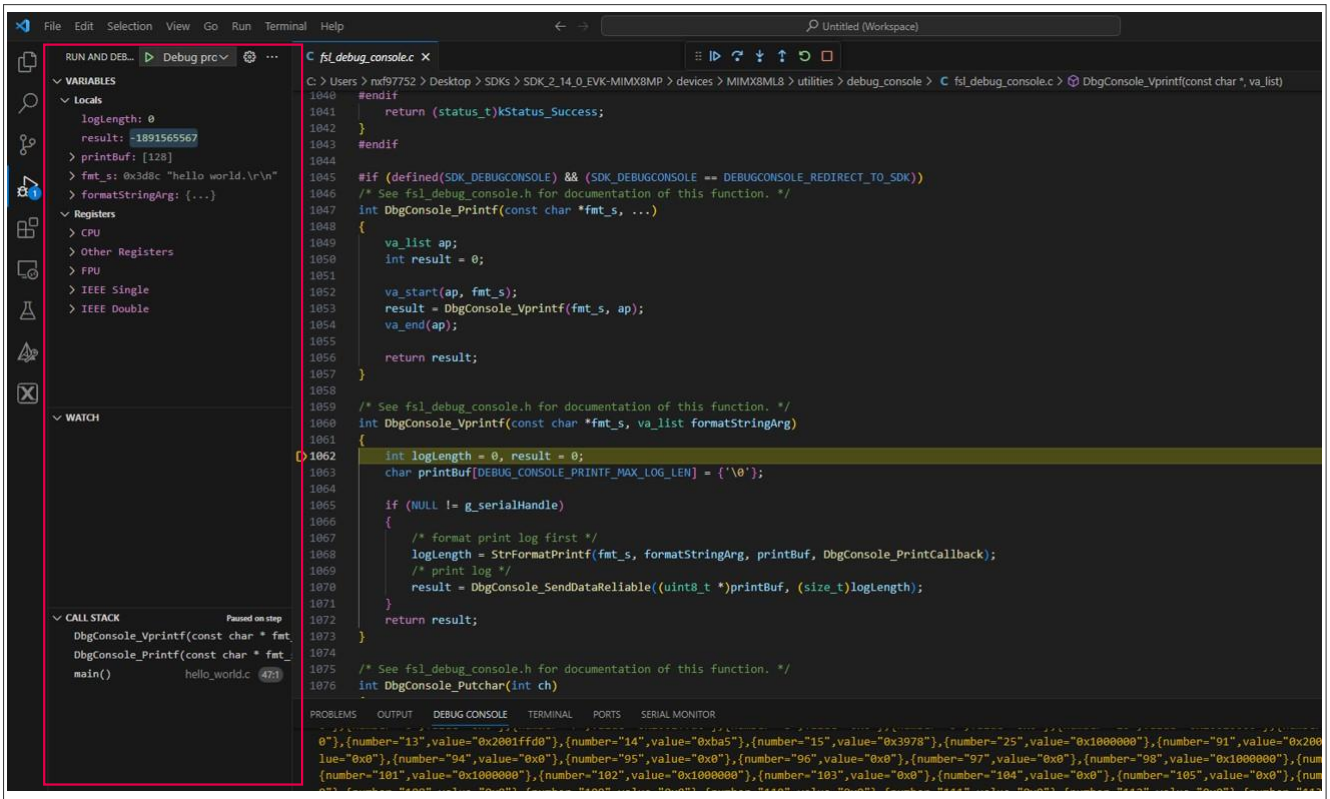

图13. 变量、寄存器和调用协议栈

6 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。

7 修订历史

表1汇总了对本文的修订。

表1. 修订历史

文档ID	发布日期	说明
AN14120 v.2.0	2024年3月13日	更新了MCUXSDK 2_15_000版本的第2.4节和第2.5节
AN14120 v.1.0	2023年11月24日	首次公开发布

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.

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.

i.MX — is a trademark of NXP B.V.

J-Link — is a trademark of SEGGER Microcontroller GmbH.

Microsoft, Azure, and ThreadX — are trademarks of the Microsoft group of companies.

目录

For more information, please visit: https://www.nxp.com.cn

Date of release: 13 March 2024
Document identifier: AN14120