AN14393

在i.MX RT1180中使用基于LPSPI的串行下载器选项

第1.0版—2004年9月4日

应用笔记

文档信息

信息	内容
关键词	AN14393、MIMXRT1180-EVK、i.MX RT1180、串行下载器、MCU-Link Pro、BusPal、USBSIO
摘要	本文介绍了在i.MX RT1180上实现基于LPSPI的串行下载器的两种方法。

1 介绍

本文介绍了在i.MX RT1180上通过LPSPI使用串行下载器的两种方法，如下所示：

- 用BusPal和MIMXRT1010-EVK实现基于LPSPI的串行下载器：BusPal充当UART和SPI之间的总线转换器。它由软件实现，并且必须在一个硬件平台上运行。BusPal可以通过UART接收blhost 命令，并通过SPI将其发送到MIMXRT1180-EVK平台。
- 用USBSIO和MCU-Link Pro实现基于LPSPI的串行下载器：USBSIO是MCU-Link Pro所支持的功能。它是一个基于LPC55S69的固件，可以与串行I/O设备进行通信。USBSIO可以通过USB接收blhost命令，并通过SPI将其发送到MIMXRT1180-EVK平台。

1.1 串行下载器

i.MX RT1180串行下载器（串行启动）可帮助将启动映像下载到启动设备，然后可以在这些设备上运行该映像。

表1列出了i.MX RT1180上的串行下载器外设的PinMux。

表1. 串行下载器外设的PinMux

外设	实体	端口(IO功能)	PAD
LPUART	1	LPUART1_TX	GPIO_AON_08
		LPUART1_RX	GPIO_AON_09
LPSPI	1	LPSPI1_SCK	GPIO_AON_04
		LPSPI1_PCS0	GPIO_AON_05
		LPSPI1_SDO	GPIO_AON_06
		LPSPI1_SDI	GPIO_AON_07
USB	1	USB1_DN	USB1_DN
		USB1_DP	USB1_DP

图1所示为串行下载器的示意图。PC上的blhost可以通过USB和LPUART直接与i.MX RT1180进行通信。然而，如果用户想通过LPSPI进行通信，则需要使用USBSIO或BusPal。MCU-Link Pro集成了USBSIO功能，而BusPal则可以在许多平台上进行定制。

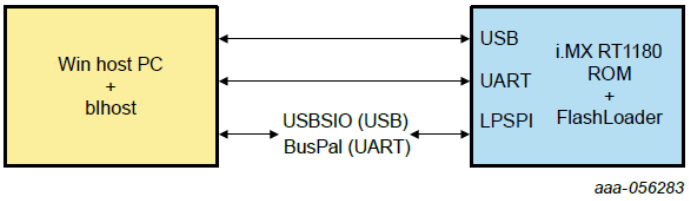


图1. 串行下载器图

1.2 MCU-Link Pro

恩智浦携手Embedded Artists开发了MCU-Link Pro。MCU-Link Pro是一款功能齐全的调试探头，可与MCUXpresso IDE和支持CMSIS-DAP或J-Link协议的第三方IDE配合使用。图2所示为MCU-Link Pro。

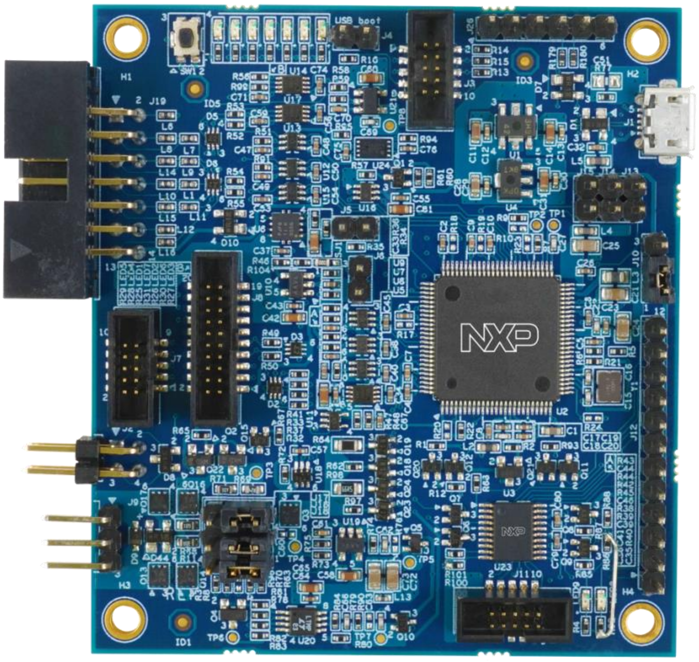


图2. MCU-Link Pro概览

MCU-Link Pro具有以下功能：

- CMSIS-DAP固件支持带有SWD调试接口的所有恩智浦基于Arm Cortex-M的MCU
- SEGGER J-Link固件选项
- 高速USB主机接口
- 双USB转目标UART桥接器（VCOM）
- 同时进行目标供电和电流测量
- SWO分析和I/O功能
- 支持CMSIS-SWO
- USB SPI和I2C桥接器，用于编程/预处理和基于主机的应用程序开发
- 可为目标系统提供高达350mA的电源（在1.8V或3.3V下）
- 模拟信号跟踪输入
- 用于外设仿真的板载用户可编程LPC804
- 用于诊断问题的多个状态LED
- 目标复位按钮

1.3 Blhost

blhost应用程序用于在主机上向运行MCU引导加载程序实现的恩智浦平台发出命令。blhost应用程序与MCU引导加载程序配合使用，允许用户在无需编程工具的情况下将固件应用程序编程到MCU设备上。Blhost可从恩智浦官方网站下载。

blhost应用程序可通过IIC、SPI、UART和USB与MCU引导加载程序连接。blhost还支持BusPal和USBSIO，因为IIC和SPI命令只对Arm-Linux版本的blhost有效。如果使用主机PC，则需要BusPal和USBSIO。

1.4 BusPal

BusPal是与blhost配套使用的嵌入式软件工具。该工具作为总线转换器，通过UART与blhost建立连接，并通过I2C和SPI与目标设备建立连接。它协助blhost传递命令和来自目标设备的响应。

BusPal的源代码随MCU引导加载程序发布版本一起提供，并且此源代码可以进行定制，以便在其他平台上运行。图3所示为BusPal在blhost与目标设备通信中所起的作用。

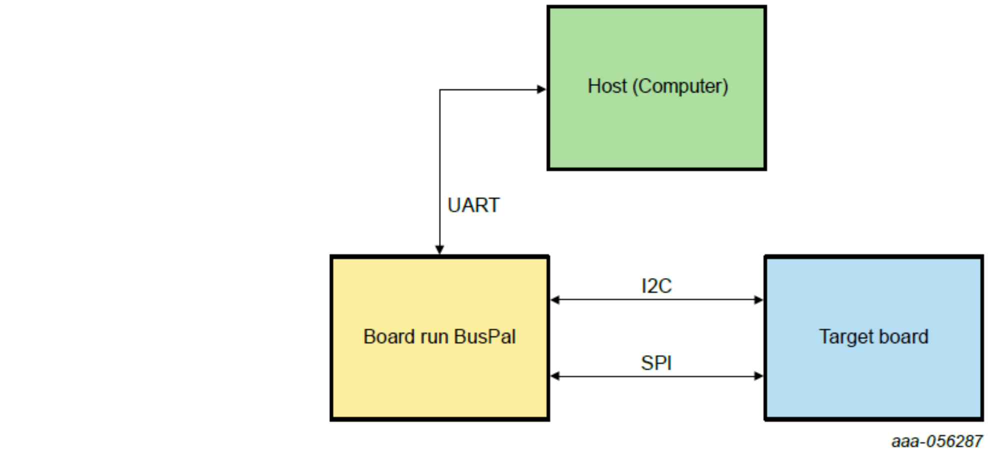


图3. BusPal示意图

1.5 USBSIO

USBSIO也被称为LPC USB串行I/O（LPCUSBSIO），它是一个基于LPC55S69的固件，可以与连接到MCU-Link Pro的串行I/O设备进行通信。

USBSIO包含两个部分：

- *LPC USB串行I/O库*：它是为PC应用程序提供的通用API。这部分内置于blhost中。
- *LPC串行I/O端口*：它是接收USB信息并使用I2C、SPI和GPIO接口将其传输到设备的固件。这部分内置于MCU-Link Pro中，而MCU-Link不支持USBSIO。

2 快速入门

本章介绍了MCU-Link Pro、MIMXRT1010-EVK开发板、MIMXRT1180-EVK开发板和硬件准备的快速入门。

2.1 准备MCU-Link Pro

确保MCU-Link Pro固件已下载到电路板上。MCU-Link Pro固件可从《MCU-Link Pro快速入门》中下载。

图4所示为MCU-Link Pro上的数字I/O连接器（J19）。

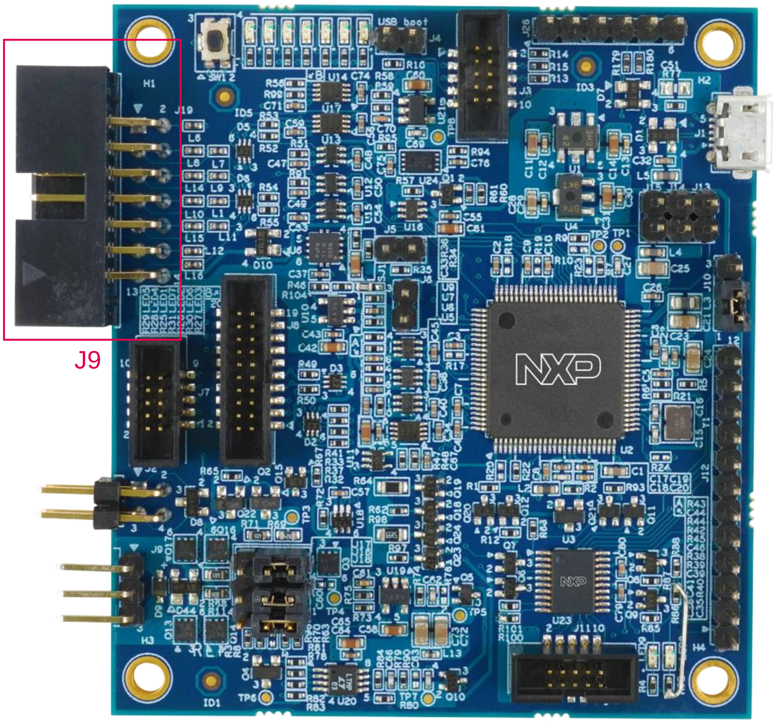


图4. MCU-Link Pro上的数字I/O连接器（J19）

表2所示为数字I/O连接器（J19）上的SPI连接引脚。

表2. 数字I/O连接器（J19）上的SPI连接引脚

J19引脚	信号
1	GND
2	SPI PCS
3	SPI MOSI
4	SPI CLK
5	SPI MISO

2.2 准备MIMXRT1010-EVK开发板

在本文档中，BusPal运行在MIMXRT1010-EVK平台上。请确保BusPal代码已下载到MIMXRT1010-EVK开发板上。

从NXP_Kinetis_Bootloader_2.0.0软件包中下载BusPal示例代码。也可以使用相同链接将此代码移植到其他定制平台上。

***注：**MIMXRT1010-EVK并不包含在此软件包中，因此需要移植BusPal。*

图5显示了MIMXRT1010-EVK上的数字I/O连接器J56、J57和J60。

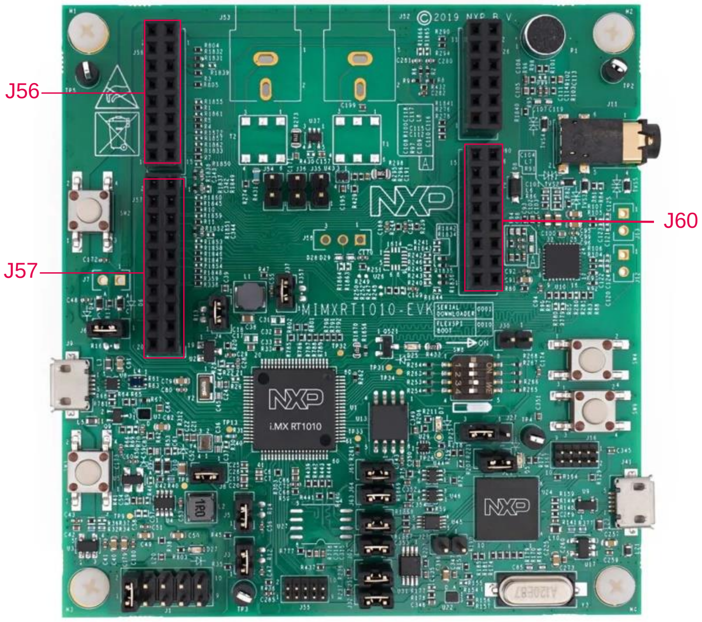


图5. MIMXRT1010-EVK上的数字I/O连接器J56、J57和J60

表3所示为J56上的UART连接引脚。

表3. J56上的UART连接引脚

J56引脚	信号
2	LPUART1_RXD
4	LPUART1_TXD

表4所示为J57上的SPI连接引脚。

表4. J57上的SPI连接引脚

J57引脚	信号
6	LPSPI1_PCS0
8	LPSPI1_MOSI
10	LPSPI1_MISO
12	LPSPI1_SCK

2.3 准备MIMXRT1180-EVK开发板

MIMXRT1180-EVK开发板是一个为展示i.MX RT1180处理器的常用功能而设计的平台。然而，EVK上的LPSPI1引脚无法直接连接。

图6显示了LPSPI1引脚可通过U13连接。

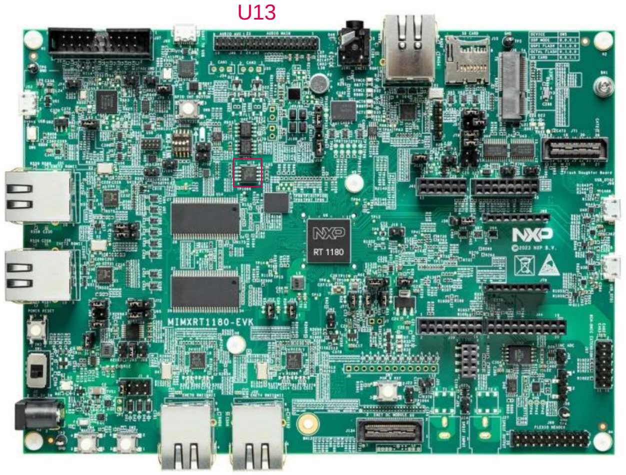


图6. MIMXRT1180-EVK上的U13

表5所示为LPSPI的连接引脚。

表5. U13上的LPSPI连接引脚

U13	信号
1	LPSPI1_PCS0
2	LPSPI1_MISO
5	LPSPI1_MOSI
6	LPSPI1_SCK

这些引脚上的飞线可用于连接该开发板上的LPSPI信号。

2.4 基于LPSPI和BusPal的串行下载器的硬件准备

表6列出了在使用BusPal时，MIMXRT1180-EVK和MIMXRT1010-EVK之间的引脚连接。

表6. MIMXRT1180-EVK和MIMXRT1010-EVK之间的硬件连接

MIMXRT1180-EVK	引脚	MIMXRT1010-EVK	引脚
LPSPI1_PCS0	U13-1	LPSPI1_PCS0	J57-6
LPSPI1_MISO	U13-2	LPSPI1_MOSI	J57-8
LPSPI1_MOSI	U13-5	LPSPI1_MISO	J57-10
LPSPI1_SCK	U13-6	LPSPI1_SCK	J57-12
GND	J37-20	GND	J57-14

表7列出了MIMXRT1010-EVK通过USB转UART连接器与主机PC之间的引脚连接。

表7. 主机PC和MIMXRT1010-EVK之间的硬件连接

MIMXRT1010-EVK开发板	引脚
LPUART1_RXD	J56-2
LPUART1_TXD	J56-4
GND	J60-14

2.5 基于LPSPI和USBSIO的串行下载器的硬件准备

如果使用串行下载器的USBSIO，则需要使用MCU-Link Pro。表8列出了MIMXRT1180-EVK与MCU-Link Pro之间的引脚连接。

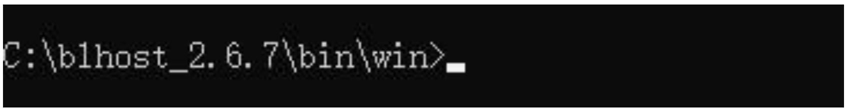
表8. MIMXRT1180-EVK和MCU-Link Pro之间的硬件连接

MIMXRT1180-EVK	引脚	MCU-Link Pro	引脚
LPSPI1_PCS0	U13-1	LPSPI1_PCS0	J19-2
LPSPI1_MISO	U13-2	LPSPI1_MOSI	J19-3
LPSPI1_MOSI	U13-5	LPSPI1_MISO	J19-5
LPSPI1_SCK	U13-6	LPSPI1_SCK	J19-4
GND	J37-20	GND	J19-1

3 用BusPal和MIMXRT1010-EVK实现基于LPSPI的串行下载器

硬件准备完成后，要用BusPal和MIMXRT1010-EVK启动基于LPSPI的串行下载器，请执行以下步骤：

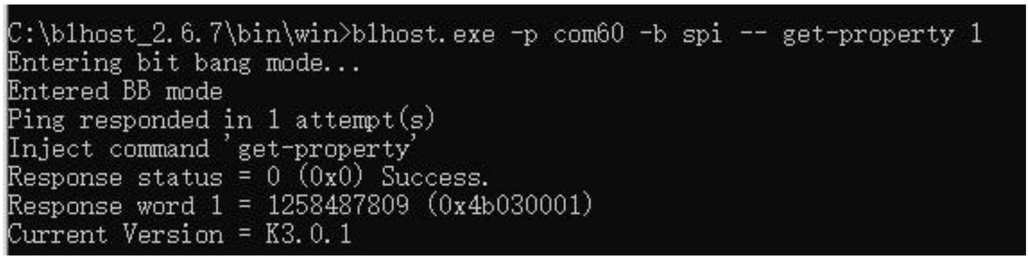
1. 将MIMXRT1180-EVK开发板切换到串行下载器模式。
2. 运行命令行工具，并转到`blhost/bin/win`目录，如图7所示。



```
C:\blhost_2.6.7\bin\win>_
```

图7. 转到blhost窗口目录

3. 输入`blhost.exe -p comi -b spi -- get-property 1`。检查ROM是否能成功通信。参见图8。



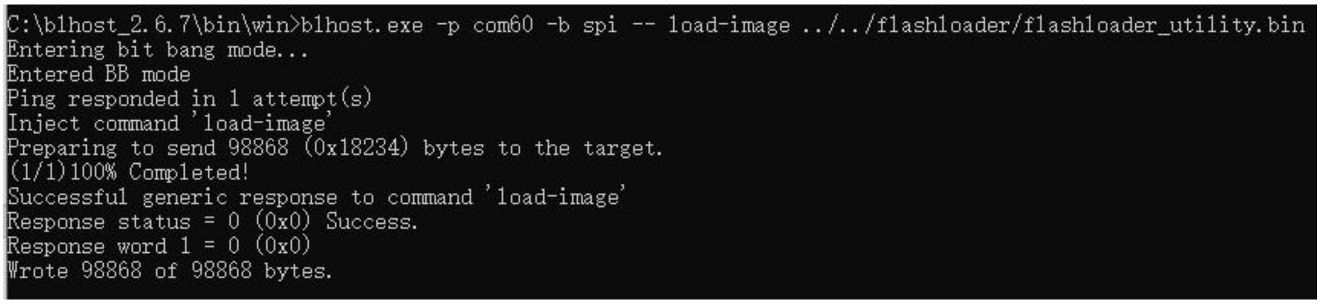
```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- get-property 1
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'get-property'
Response status = 0 (0x0) Success.
Response word 1 = 1258487809 (0x4b030001)
Current Version = K3.0.1
```

图8. 通过BusPal与ROM通信

SPI默认参数如下：

- 端口0 (0)
- 引脚0 (0)
- 100kbit/s (100)
- 低电平有效极性 (1)
- 下降沿采样 (1)
- 发送命令 = `blhost.exe -p comi -b spi,0,0,100,1,1 -- get-property 1`

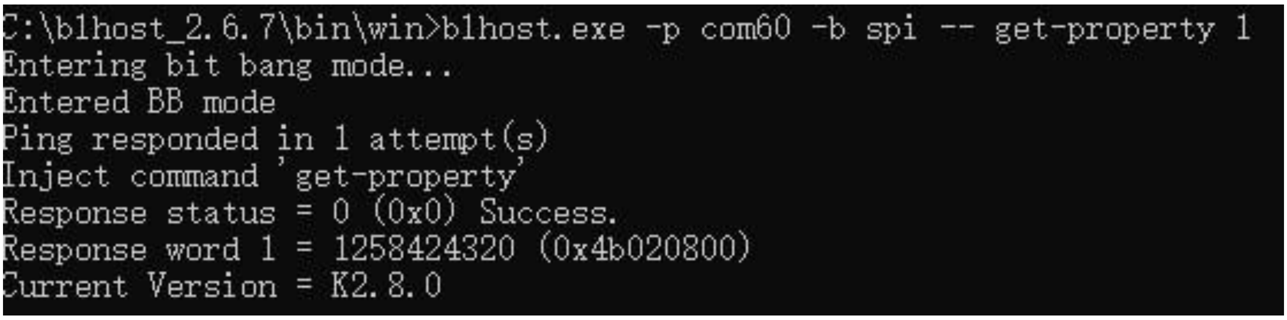
4. 输入`blhost.exe -p comi -b spi -- load-image <file>`。此步骤下载并运行flashloader。确保将容器添加到flashloader。SDK生成flashloader。请参见图9。



```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- load-image ../../flashloader/flashloader_utility.bin
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'load-image'
Preparing to send 98868 (0x18234) bytes to the target.
(1/1)100% Completed!
Successful generic response to command 'load-image'
Response status = 0 (0x0) Success.
Response word 1 = 0 (0x0)
Wrote 98868 of 98868 bytes.
```

图9. 通过BusPal下载并运行flashloader

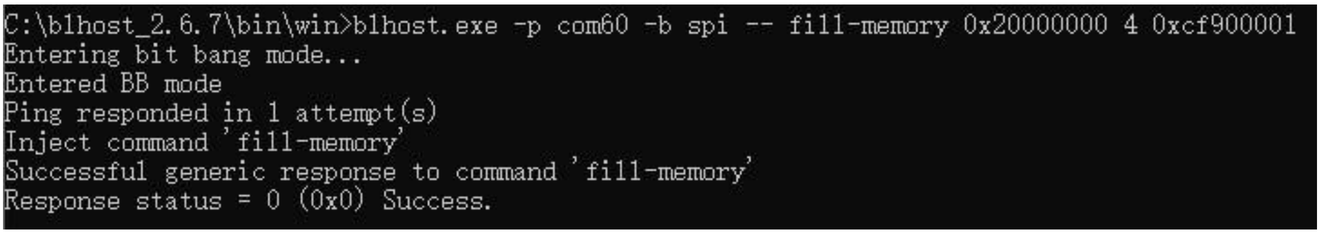
5. 现在，blhost可以与flashloader通信了。再次输入`blhost.exe -p comi -b spi -- get- property 1`。检查flashloader是否能成功通信。请参见图10。



```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- get-property 1
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'get-property'
Response status = 0 (0x0) Success.
Response word 1 = 1258424320 (0x4b020800)
Current Version = K2.8.0
```

图10. 通过BusPal与flashloader通信

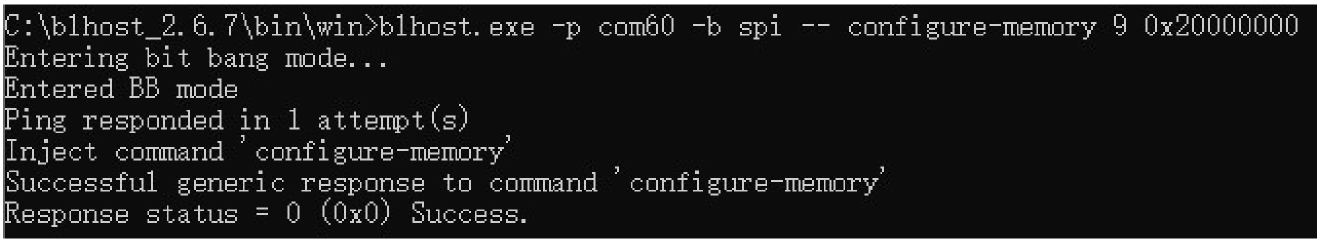
6. 在TCM中设置FlexSPI实体，输入`blhost.exe -p comi -b spi -- fill-memory 0x20000000 4 0xcf900001`。参见图11。



```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- fill-memory 0x20000000 4 0xcf900001
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'fill-memory'
Successful generic response to command 'fill-memory'
Response status = 0 (0x0) Success.
```

图11. 通过BusPal在TCM中设置FlexSPI实体

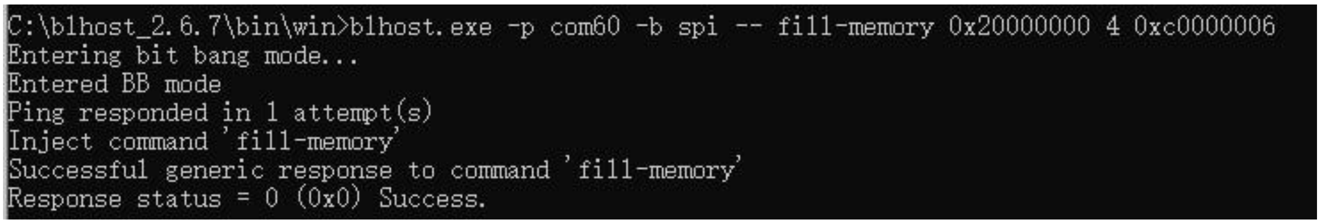
7. 配置内存，输入`blhost.exe -p comi -b spi -- configure-memory 9 0x20000000`。参见图12。



```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- configure-memory 9 0x20000000
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'configure-memory'
Successful generic response to command 'configure-memory'
Response status = 0 (0x0) Success.
```

图12. 通过BusPal配置内存

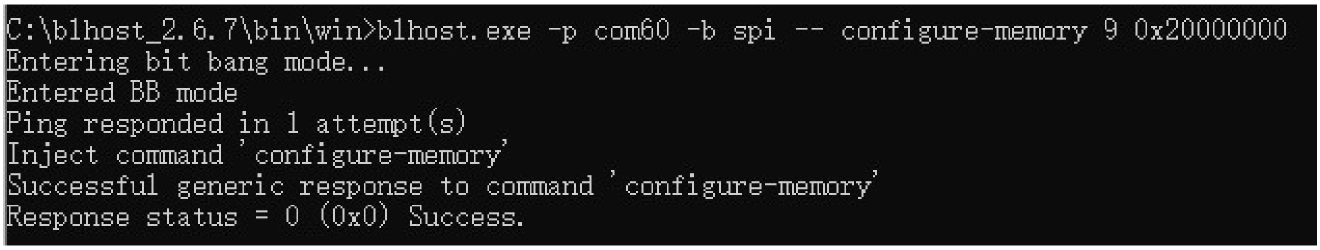
8. 在TCM中设置NOR闪存配置参数，输入`blhost.exe -p comi -b spi -- fill-memory 0x20000000 4 0xc0000006`。参见图13。



```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- fill-memory 0x20000000 4 0xc0000006
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'fill-memory'
Successful generic response to command 'fill-memory'
Response status = 0 (0x0) Success.
```

图13. 通过BusPal设置NOR闪存配置参数

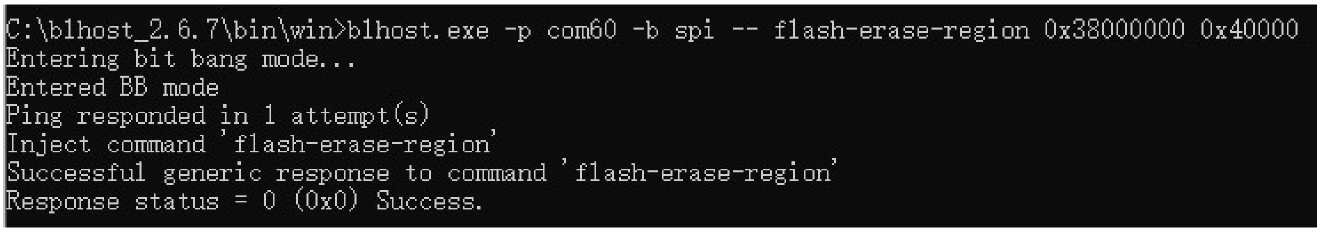
9. 配置内存，输入`blhost.exe -p comi -b spi -- configure-memory 9 0x20000000`。参见图14。



```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- configure-memory 9 0x20000000
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'configure-memory'
Successful generic response to command 'configure-memory'
Response status = 0 (0x0) Success.
```

图14. 通过BusPal配置内存

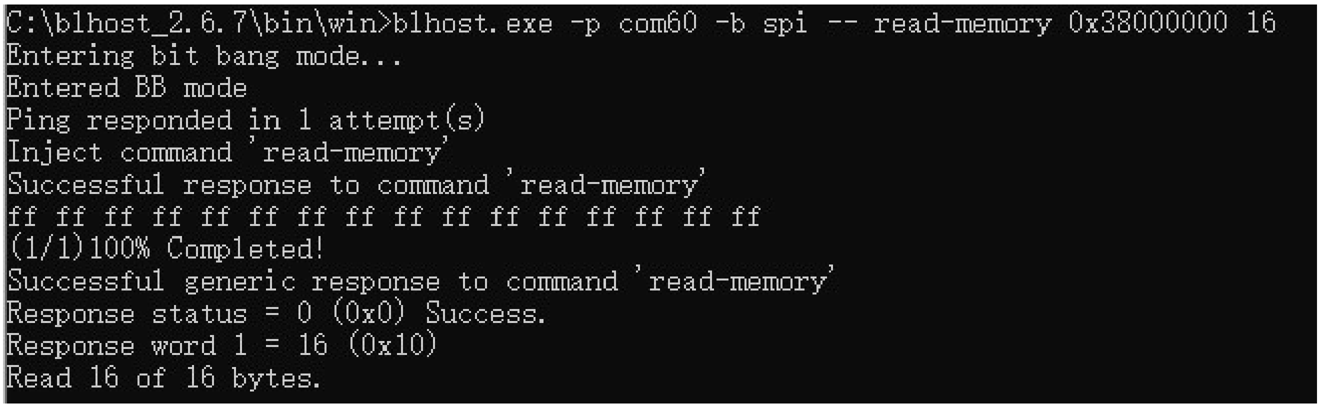
10. 擦除闪存区域，输入`blhost.exe -p comi -b spi -- flash-erase-region 0x38000000 0x40000`。参见图15。



```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- flash-erase-region 0x38000000 0x40000
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'flash-erase-region'
Successful generic response to command 'flash-erase-region'
Response status = 0 (0x0) Success.
```

图15. 通过BusPal擦除闪存区域

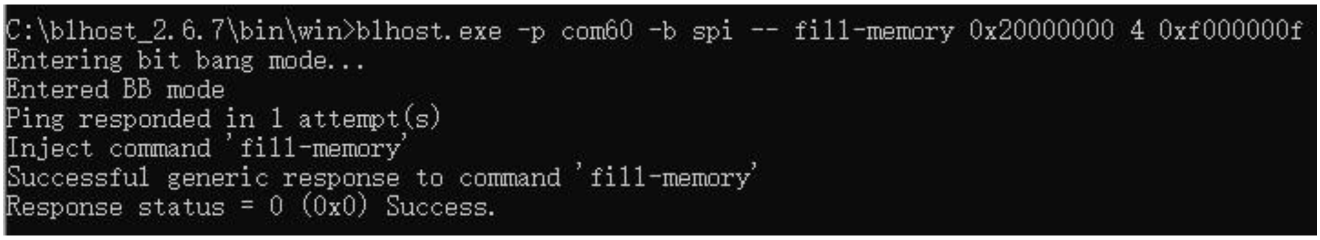
11. 要确认闪存已被擦除，请读取闪存。输入`blhost.exe -p comi -b spi -- read-memory 0x38000000 16`。参见图16。



```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- read-memory 0x38000000 16
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'read-memory'
Successful response to command 'read-memory'
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
(1/1)100% Completed!
Successful generic response to command 'read-memory'
Response status = 0 (0x0) Success.
Response word 1 = 16 (0x10)
Read 16 of 16 bytes.
```

图16. 通过BusPal读取闪存

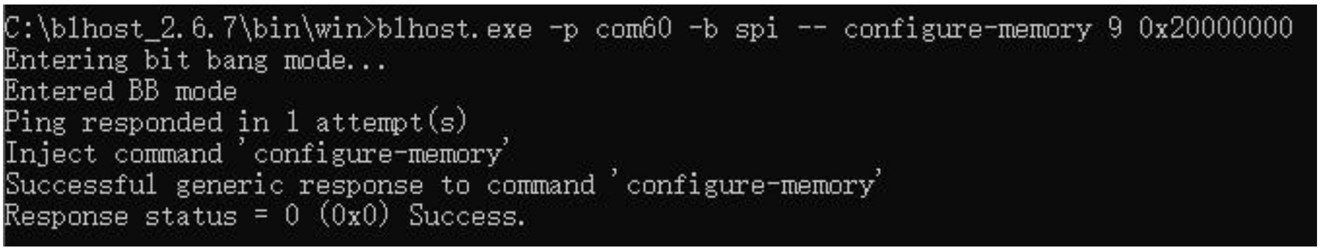
12. 添加所用的FDCB XIP，输入`blhost.exe -p comi -b spi -- fill-memory 0x20000000 4 0xf000000f`。参见图17。



```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- fill-memory 0x20000000 4 0xf000000f
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'fill-memory'
Successful generic response to command 'fill-memory'
Response status = 0 (0x0) Success.
```

图17. 通过BusPal添加所用的FDCB XIP

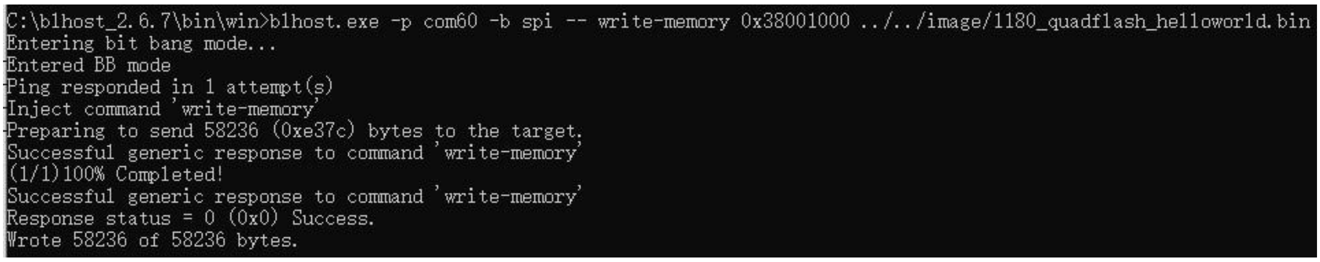
13. 配置内存，输入blhost.exe -p comi -b spi -- configure-memory 9 0x20000000。参见图18。



```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- configure-memory 9 0x20000000
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'configure-memory'
Successful generic response to command 'configure-memory'
Response status = 0 (0x0) Success.
```

图18. 通过BusPal配置内存

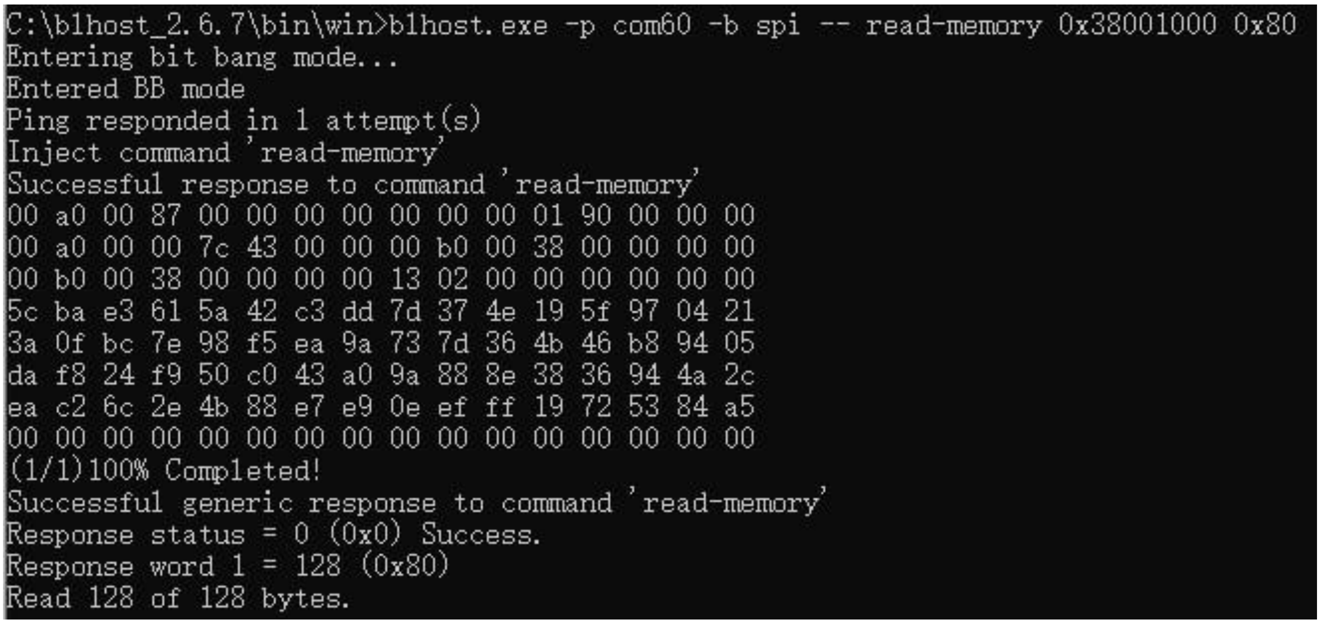
14. 写入映像，输入blhost.exe -p comi -b spi -- write-memory <path>。参见图19。



```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- write-memory 0x38001000 ../../image/1180_quadflash_helloworld.bin
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'write-memory'
Preparing to send 58236 (0xe37c) bytes to the target.
Successful generic response to command 'write-memory'
(1/1)100% Completed!
Successful generic response to command 'write-memory'
Response status = 0 (0x0) Success.
Wrote 58236 of 58236 bytes.
```

图19. 通过BusPal写入内存

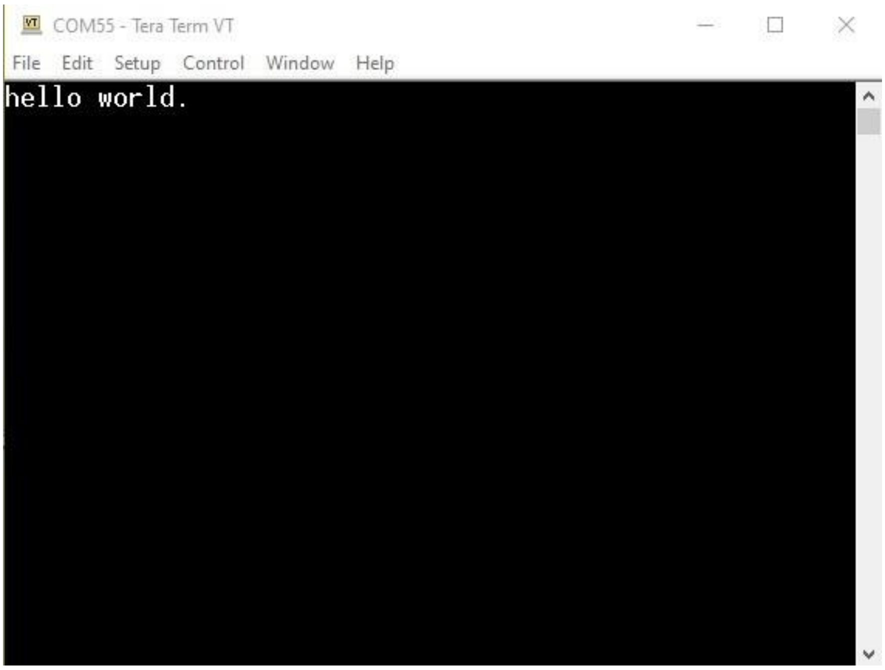
15. 要确认写入是否成功，请读取映像。输入blhost.exe -p comi -b spi -- read-memory 0x38001000 0x400。参见图20。



```
C:\blhost_2.6.7\bin\win>blhost.exe -p com60 -b spi -- read-memory 0x38001000 0x80
Entering bit bang mode...
Entered BB mode
Ping responded in 1 attempt(s)
Inject command 'read-memory'
Successful response to command 'read-memory'
00 a0 00 87 00 00 00 00 00 00 00 01 90 00 00 00
00 a0 00 00 7c 43 00 00 00 b0 00 38 00 00 00 00
00 b0 00 38 00 00 00 00 13 02 00 00 00 00 00 00
5c ba e3 61 5a 42 c3 dd 7d 37 4e 19 5f 97 04 21
3a 0f bc 7e 98 f5 ea 9a 73 7d 36 4b 46 b8 94 05
da f8 24 f9 50 c0 43 a0 9a 88 8e 38 36 94 4a 2c
ea c2 6c 2e 4b 88 e7 e9 0e ef ff 19 72 53 84 a5
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
(1/1)100% Completed!
Successful generic response to command 'read-memory'
Response status = 0 (0x0) Success.
Response word 1 = 128 (0x80)
Read 128 of 128 bytes.
```

图20 通过BusPal读取内存

16. 将MIMXRT1180-EVK开发板切换到内部启动模式。图21所示为UART显示的消息。



```
hello world.
```

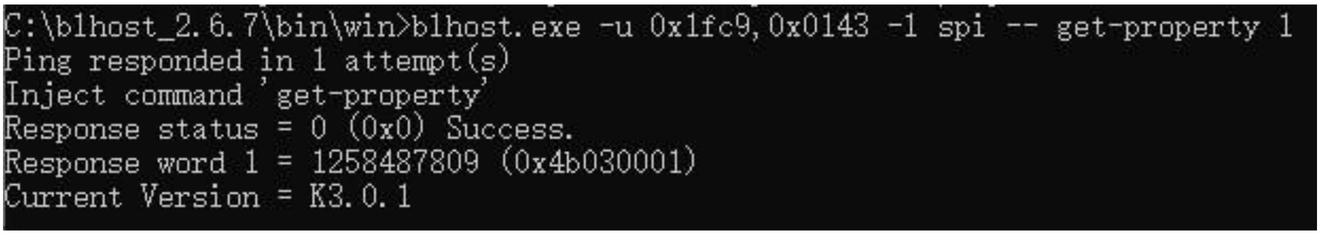
图21. 启动成功

4 用USBSIO和MCU-Link Pro实现基于LPSPI的串行下载器

硬件准备完成后，要用USBSIO和MCU-Link Pro启动基于LPSPI的串行下载器，请执行以下步骤：

1. 将MIMXRT1180-EVK开发板切换到串行下载器模式。

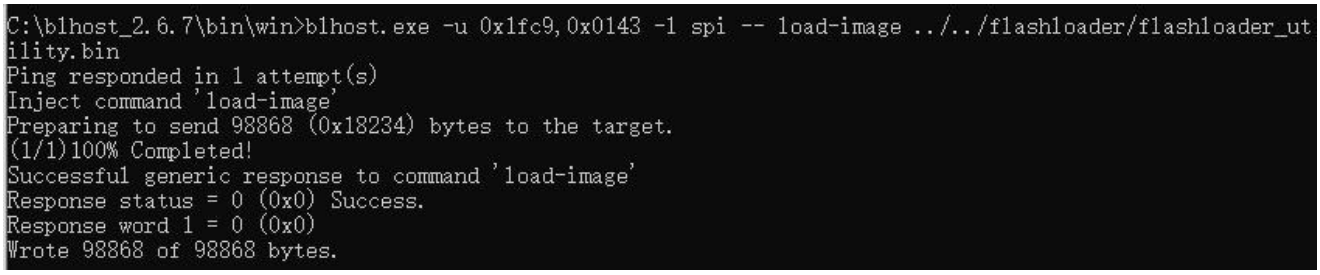
2. 运行命令行工具，并转到`blhost/bin/win`目录，如图7所示。
3. 输入`blhost.exe -u [<vid>,<pid>] -l spi -- get-property 1`。检查ROM是否能成功通信。参见图22。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- get-property 1
Ping responded in 1 attempt(s)
Inject command 'get-property'
Response status = 0 (0x0) Success.
Response word 1 = 1258487809 (0x4b030001)
Current Version = K3.0.1
```

图22. 通过USBSIO与ROM通讯

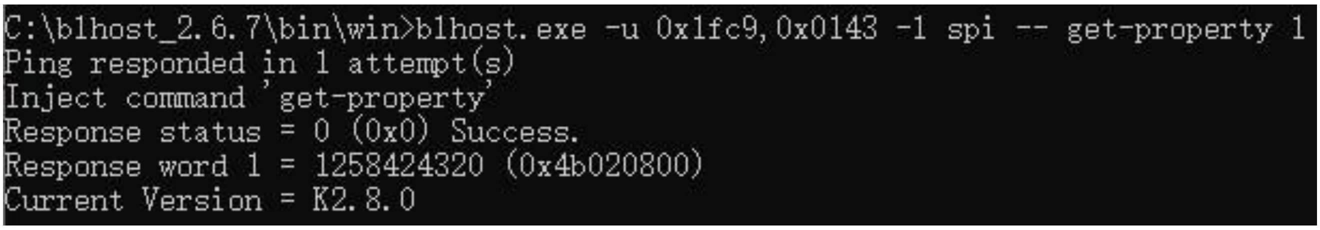
4. 输入`blhost.exe -u [<vid>,<pid>] -l spi -- load-image <file>`。此步骤下载并运行flashloader。确保将容器添加到flashloader。SDK生成flashloader。请参见图23。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- load-image ../../flashloader/flashloader_ut
ility.bin
Ping responded in 1 attempt(s)
Inject command 'load-image'
Preparing to send 98868 (0x18234) bytes to the target.
(1/1)100% Completed!
Successful generic response to command 'load-image'
Response status = 0 (0x0) Success.
Response word 1 = 0 (0x0)
Wrote 98868 of 98868 bytes.
```

图23. 通过USBSIO下载并运行flashloader

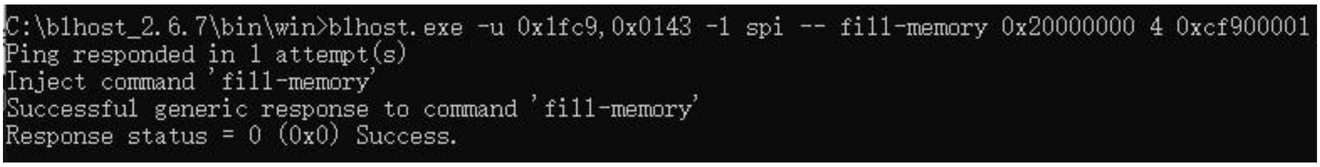
5. 现在，blhost可以与flashloader通信了。再次输入`blhost.exe -u [<vid>,<pid>] -l spi -- get-property 1`。检查flashloader是否能成功通信。请参见图24。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- get-property 1
Ping responded in 1 attempt(s)
Inject command 'get-property'
Response status = 0 (0x0) Success.
Response word 1 = 1258424320 (0x4b020800)
Current Version = K2.8.0
```

图24. 通过USBSIO与flashloader通信

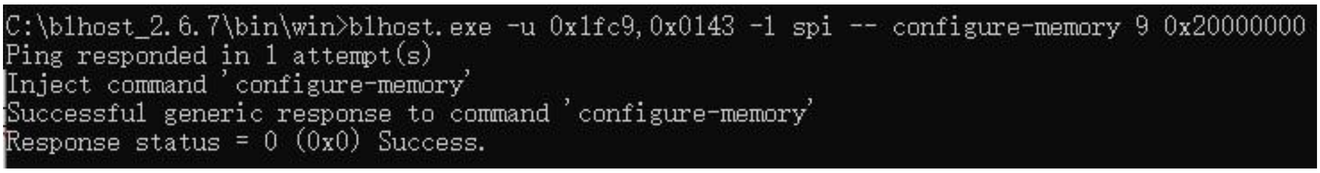
6. 在TCM中设置FlexSPI实体，输入`blhost.exe -u [<vid>,<pid>] -l spi -- fill-memory 0x20000000 4 0xcf900001`。参见图25。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- fill-memory 0x20000000 4 0xcf900001
Ping responded in 1 attempt(s)
Inject command 'fill-memory'
Successful generic response to command 'fill-memory'
Response status = 0 (0x0) Success.
```

图25. 通过USBSIO在TCM中设置FlexSPI实体

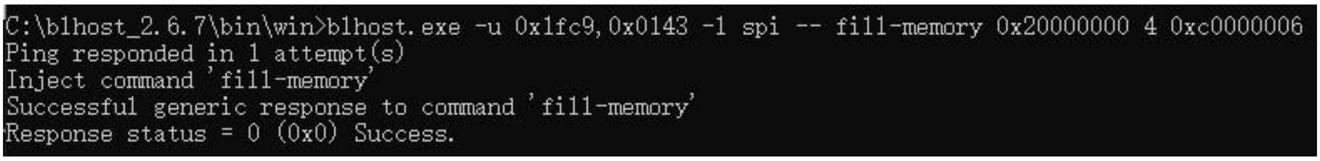
7. 配置内存，输入`blhost.exe -u [<vid>,<pid>] -l spi -- configure-memory 9 0x20000000`。参见图26。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- configure-memory 9 0x20000000
Ping responded in 1 attempt(s)
Inject command 'configure-memory'
Successful generic response to command 'configure-memory'
Response status = 0 (0x0) Success.
```

图26. 通过USBSIO配置内存

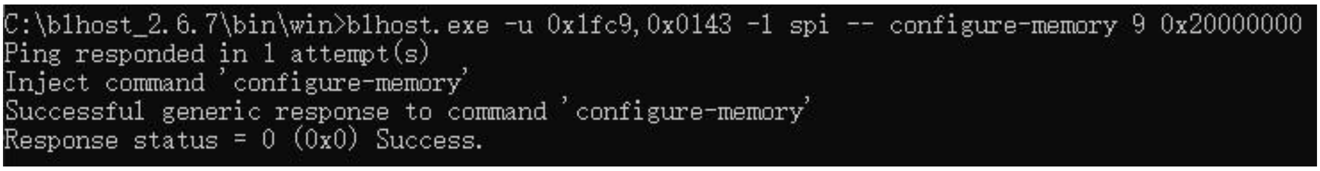
8. 在TCM中设置NOR闪存配置参数，输入`blhost.exe -u [<vid>,<pid>] -l spi -- fill-memory 0x20000000 4 0xc0000006`。参见图27。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- fill-memory 0x20000000 4 0xc0000006
Ping responded in 1 attempt(s)
Inject command 'fill-memory'
Successful generic response to command 'fill-memory'
Response status = 0 (0x0) Success.
```

图27. 通过USBSIO设置NOR闪存配置参数

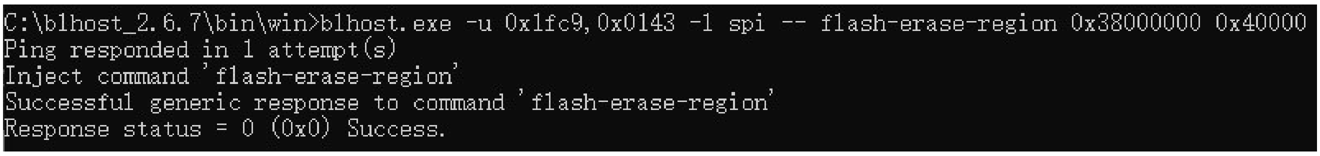
9. 配置内存，输入`blhost.exe -u [<vid>,<pid>] -l spi -- configure-memory 9 0x20000000`。参见图28。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- configure-memory 9 0x20000000
Ping responded in 1 attempt(s)
Inject command 'configure-memory'
Successful generic response to command 'configure-memory'
Response status = 0 (0x0) Success.
```

图28. 通过USBSIO配置内存

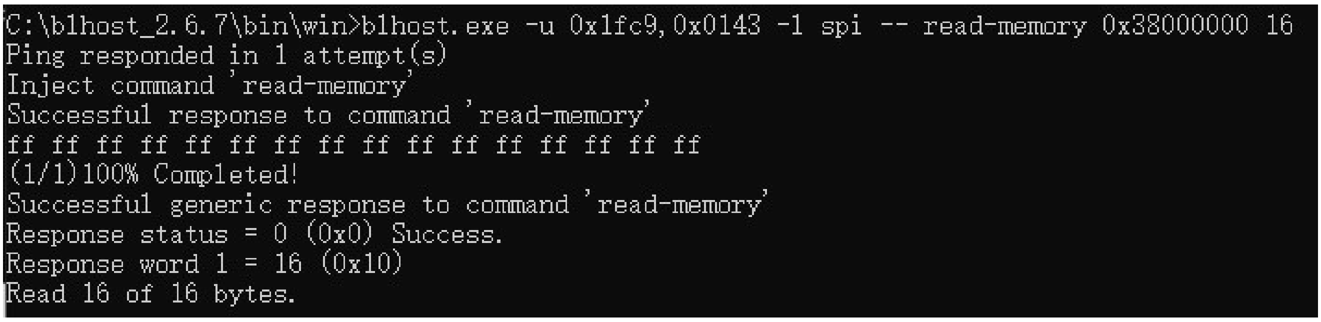
10. 擦除闪存区域，输入`blhost.exe -u [<vid>,<pid>] -l spi -- flash-erase-region 0x38000000 0x40000`。参见图29。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- flash-erase-region 0x38000000 0x40000
Ping responded in 1 attempt(s)
Inject command 'flash-erase-region'
Successful generic response to command 'flash-erase-region'
Response status = 0 (0x0) Success.
```

图29. 通过USBSIO擦除闪存区域

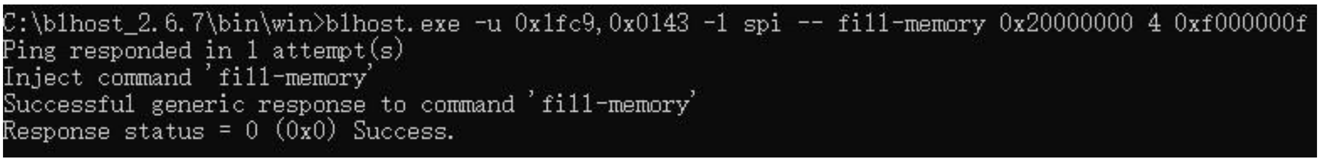
11. 要确认闪存已被擦除，请读取闪存。输入`blhost.exe -u [<vid>,<pid>] -l spi-- read-memory 0x38000000 16`。参见图30。如果成功，其结果与图30所示类似。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- read-memory 0x38000000 16
Ping responded in 1 attempt(s)
Inject command 'read-memory'
Successful response to command 'read-memory'
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
(1/1)100% Completed!
Successful generic response to command 'read-memory'
Response status = 0 (0x0) Success.
Response word 1 = 16 (0x10)
Read 16 of 16 bytes.
```

图30. 通过USBSIO读取闪存

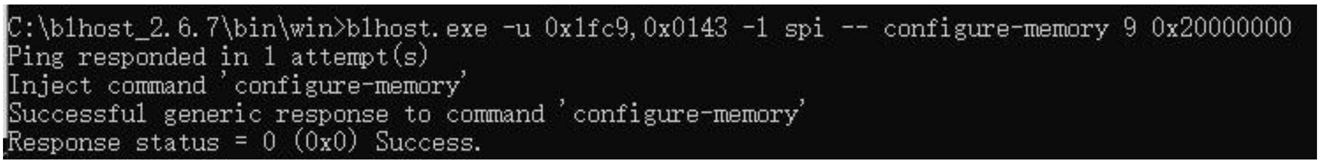
12. 添加所用的FDCB XIP，输入`blhost.exe -u [<vid>,<pid>] -l spi -- fill-memory 0x20000000 4 0xf000000f`。参见图31。如果成功，其结果与图31所示的类似。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- fill-memory 0x20000000 4 0xf000000f
Ping responded in 1 attempt(s)
Inject command 'fill-memory'
Successful generic response to command 'fill-memory'
Response status = 0 (0x0) Success.
```

图31. 通过USBSIO添加FDCB

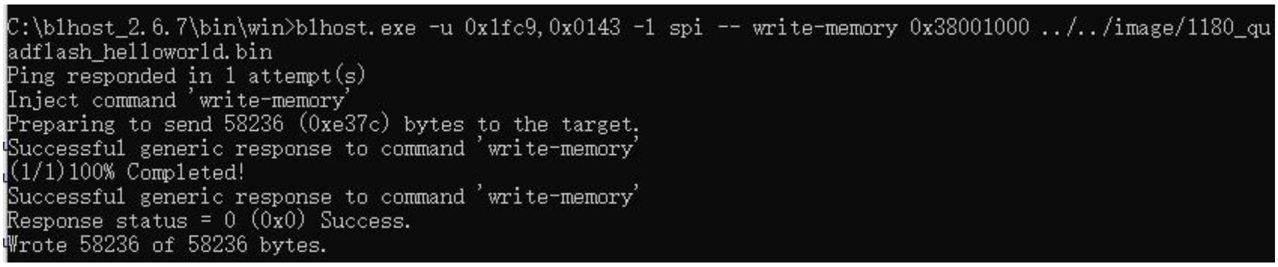
13. 配置内存，输入`blhost.exe -u [<vid>,<pid>] -l spi -- configure-memory 9 0x20000000`。参见图32。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- configure-memory 9 0x20000000
Ping responded in 1 attempt(s)
Inject command 'configure-memory'
Successful generic response to command 'configure-memory'
Response status = 0 (0x0) Success.
```

图32. 通过USBSIO配置内存

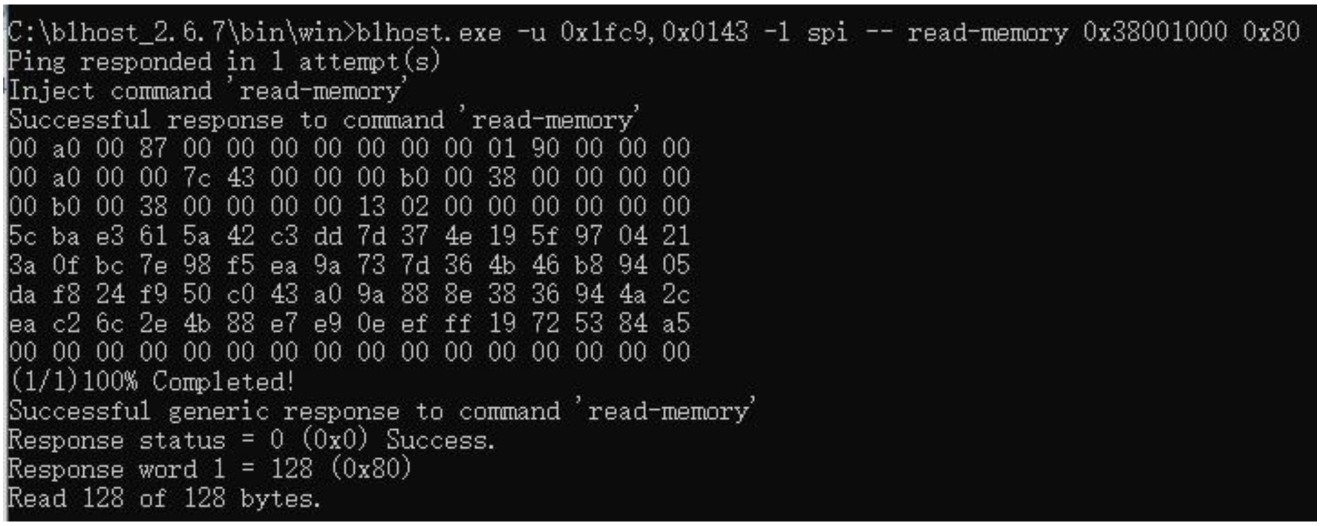
14. 写入映像，输入`blhost.exe -u [<vid>,<pid>] -l spi -- write-memory <path>`。参见图33。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- write-memory 0x38001000 ../../image/1180_qu
adflash_helloworld.bin
Ping responded in 1 attempt(s)
Inject command 'write-memory'
Preparing to send 58236 (0xe37c) bytes to the target.
Successful generic response to command 'write-memory'
(1/1)100% Completed!
Successful generic response to command 'write-memory'
Response status = 0 (0x0) Success.
Wrote 58236 of 58236 bytes.
```

图33. 通过USBSIO写入映像

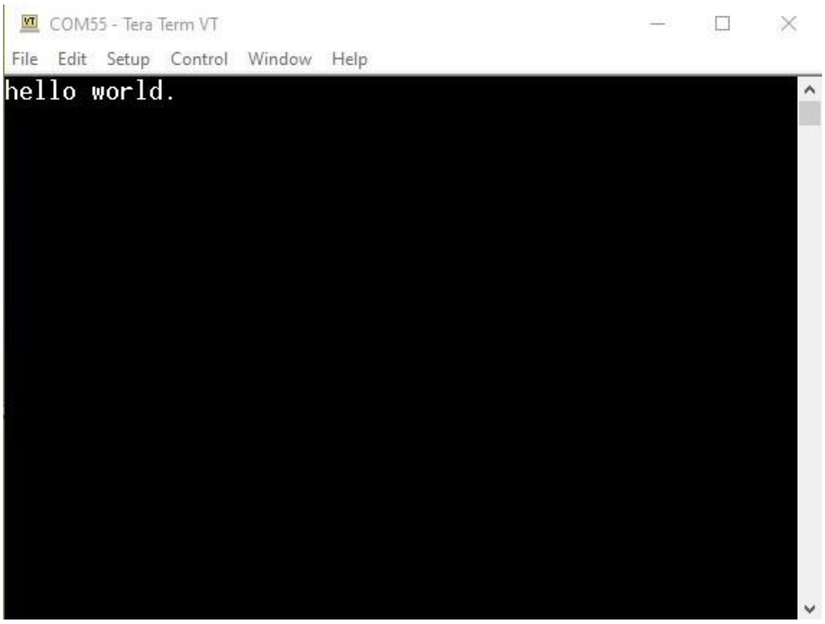
15. 要确认写入是否成功，请读取映像。输入`blhost.exe -u [<vid>,<pid>] -l spi-- read-memory 0x38001000 0x100`。参见图34。



```
C:\blhost_2.6.7\bin\win>blhost.exe -u 0x1fc9,0x0143 -l spi -- read-memory 0x38001000 0x80
Ping responded in 1 attempt(s)
Inject command 'read-memory'
Successful response to command 'read-memory'
00 a0 00 87 00 00 00 00 00 00 00 01 90 00 00 00
00 a0 00 00 7c 43 00 00 00 b0 00 38 00 00 00 00
00 b0 00 38 00 00 00 00 13 02 00 00 00 00 00 00
5c ba e3 61 5a 42 c3 dd 7d 37 4e 19 5f 97 04 21
3a 0f bc 7e 98 f5 ea 9a 73 7d 36 4b 46 b8 94 05
da f8 24 f9 50 c0 43 a0 9a 88 8e 38 36 94 4a 2c
ea c2 6c 2e 4b 88 e7 e9 0e ef ff 19 72 53 84 a5
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
(1/1)100% Completed!
Successful generic response to command 'read-memory'
Response status = 0 (0x0) Success.
Response word 1 = 128 (0x80)
Read 128 of 128 bytes.
```

图34. 通过USBSIO读取映像

16. 将MIMXRT1180-EVK开发板切换到内部启动模式。图35所示为UART显示的消息。



```
hello world.
```

图35. 启动成功

5 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。

6 修订历史

表9汇总了本文的修订情况。

表9. 修订历史

文档ID	发布日期	说明
AN14393 v.1.0	2024年9月4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.

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.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

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

目录

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.