

原文链接: <https://www.nxp.com/docs/en/application-note/AN12108.pdf>

如何启用从QSPI Flash引导

1. 简介

i.MX RT系列是恩智浦提供的业界首个跨界处理器。本文档介绍了如何将可引导映像编程到外部存储设备中。有关Flashloader, MfgTool的信息, 请参考应用笔记 [“How to Enable Boot from Octal SPIFlash and SD Card”AN12107](#)。

本文档中使用的软件基于i.MXRT1050 SDK 2.4.0。开发环境是IAR Embedded Workbench 8.22.2。硬件开发环境是IMXRT1050-EVKB板。Flashloader的版本为V1.1。

内容

1. 简介.....	1
2. MIMXRT1050 EVK 板设置.....	2
2.1. EVK 设置.....	2
2.2. EVKB 设置.....	3
3. 程序工具.....	5
3.1. DAP-Link (OpenSDA MSD drag/drop).....	5
3.2. MFG 工具.....	5
4. 示例.....	5
4.1. 从 QSPI Flash引导的OpenSDA Drag/Drop.....	5
4.2. 从 QSPI Flash引导的MFG.....	10
4.3. 使用SDRAM的DCD从QSPI Flash引导的MFG.....	21
5. QSPI Flash 支持列表.....	21
6. 结论.....	22
7. 修订历史.....	23

2. MIMXRT1050 EVK 板设置

2.1. EVK 设置

为了启用板载QSPI Flash功能，需要更改EVK板设置。

步骤 1:

应移除板载Hyper Flash，否则会影响QSPI Flash的读写时序。

步骤 2:

- 将0Ω电阻焊接到R153至R158的焊盘上。

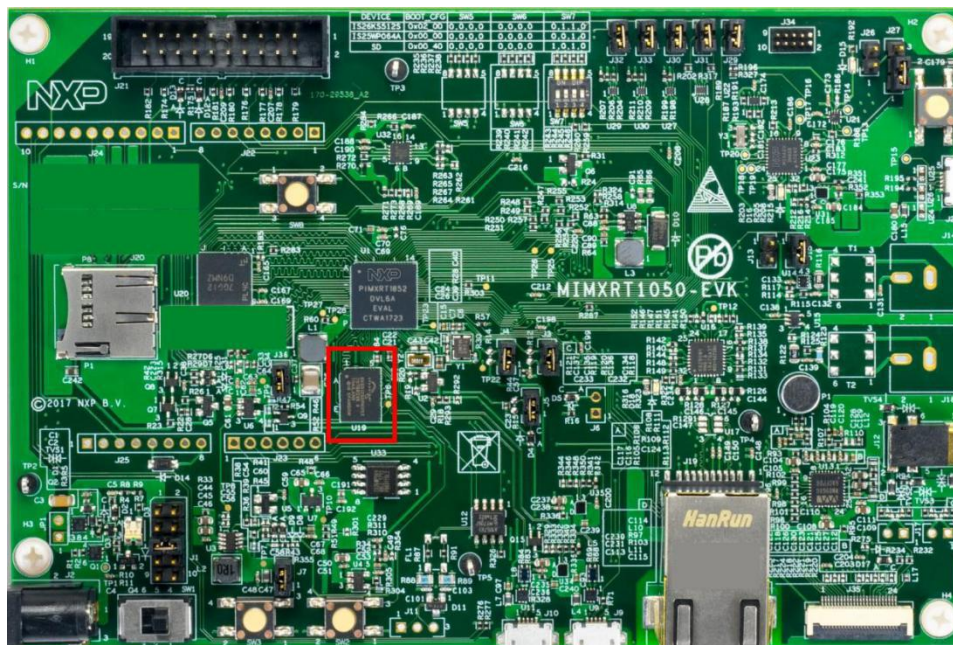


图 1. Hyper Flash

1V8 QSPI Flash

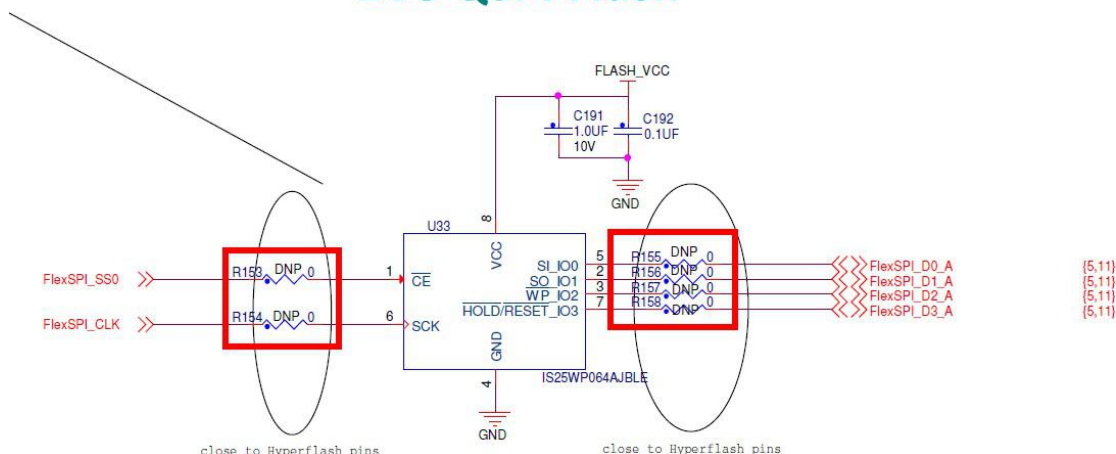


图 2. 将0Ω电阻焊接到R153至R158的焊盘上

步骤 3:

- 需要更换OpenSDA的固件。板载默认固件用于Hyper Flash，应将固件替换为QSPI Flash。Hyper Flash和QSPI Flash的固件可从[NXP Website](#)。

2.2. EVKB 设置

对于EVKB板，板载的Hyper Flash不需要移除。

拆下的电阻：R356、R361-R366。

焊接 0Ω 的电阻器：R153 - R158。

按照第2.1节的步骤3更新OpenSDA固件。

完成这些步骤后，板载QSPI Flash就可以使用了。

NOTE

即使QSPI flash本身没有DQS管脚，也要保持它是浮动的，并使它能够获得更高的读/写频率。请参考[RT1050 datasheet](#)中的表35和表36。如果未使用DQS引脚，则只支持60 MHz的工作频率，而如果DQS引脚用于输入计时，则最高可支持133 MHz的工作频率。

SDR mode with FlexSPIn_MCR0[RXCLKSRC] = 0x0, 0x1

Table 35. FlexSPI input timing in SDR mode where FlexSPIn_MCR0[RXCLKSRC] = 0X0

Symbol	Parameter	Min	Max	Unit
	Frequency of operation	—	60	MHz
T _{IS}	Setup time for incoming data	8.67	—	ns
T _{IH}	Hold time for incoming data	0	—	ns

Table 36. FlexSPI input timing in SDR mode where FlexSPIn_MCR0[RXCLKSRC] = 0X1

Symbol	Parameter	Min	Max	Unit
	Frequency of operation	—	133	MHz
T _{IS}	Setup time for incoming data	2	—	ns
T _{IH}	Hold time for incoming data	1	—	ns

图 3. SDR模式输时序参数

2.2.1. 引导header的宏

表 1 显示了为支持XIP而添加到flexspi\u或目标中的三个宏：

Table 1. Macros for the boot header

XIP_EXTERNAL_FLASH	1: Exclude the code which will change the clock of flexspi. 0: make no changes.
XIP_BOOT_HEADER_ENABLE	1: Add flexspi configuration block, image vector table, boot data and device configuration data(optional) to the image by default. 0: Add nothing to the image by default.
XIP_BOOT_HEADER_DCD_ENABLE	1: Add device configuration data to the image. 0: Do NOT add device configuration data to the image.

表 2 显示了这些宏的不同组合对构建图像的不同影响：

Table 2. Different effect on the built image with difference macros

		XIP_BOOT_HEADER_DCD_ENABLE=1	XIP_BOOT_HEADER_DCD_ENABLE=0
XIP_EXTERNAL_FLASH=1	XIP_BOOT_HEADER_ENABLE=1	Can be programmed to Hyper Flash by IDE and can run after POR reset if Hyper Flash is the boot source. SDRAM will be initialized.	Can be programmed to Hyper Flash by IDE and can run after POR reset if Hyper Flash is the boot source. SDRAM will NOT be initialized.

	XIP_BOOT_HEA DER_ENABLE=0	Can NOT run after POR reset if it is programmed by IDE even if Hyper Flash is the boot source.
	XIP_EXTERNAL_FLASH =0	This image can NOT do XIP because when this macro is set to 1, it will exclude the code which will change the clock of flexspi.

3. 程序工具

3.1. DAP-Link (OpenSDA MSD drag/drop)

- 仅限EVK上的QSPI Flash。
- 仅支持二进制文件。

注意

EVK上DAP-Link的默认固件仅支持Hyper Flash。如果使用QSPI Flash拖放，则应更换DAP-Link的固件。可以从[NXP Web](#) 下载固件。

3.2. MFG 工具

MfgTool支持i.MXRT BootROM和基于KBOOT的Flashloader，可以在工厂生产环境中使用。Mfgtool可以检测到存在连接到PC的BootROM设备，并调用“blhost”来对连接到i.MX MCU设备的目标内存设备上的映像进行编程。

blhost是一个命令行主机程序，用于与运行基于KBOOT的Bootloader的设备连接，仅是MfgTool release.sb文件支持的一部分。

4. 示例

4.1. OpenSDA Drag/Drop 并从 QSPI Flash引导

本章介绍了使用OpenSDA拖放将图像编程到QSPI Flash所需的步骤。步骤如下：

步骤 1:

- 在SDK中打开Hello world演示，并选择项目配置为flexspi_nor_debug ([图 4](#))。

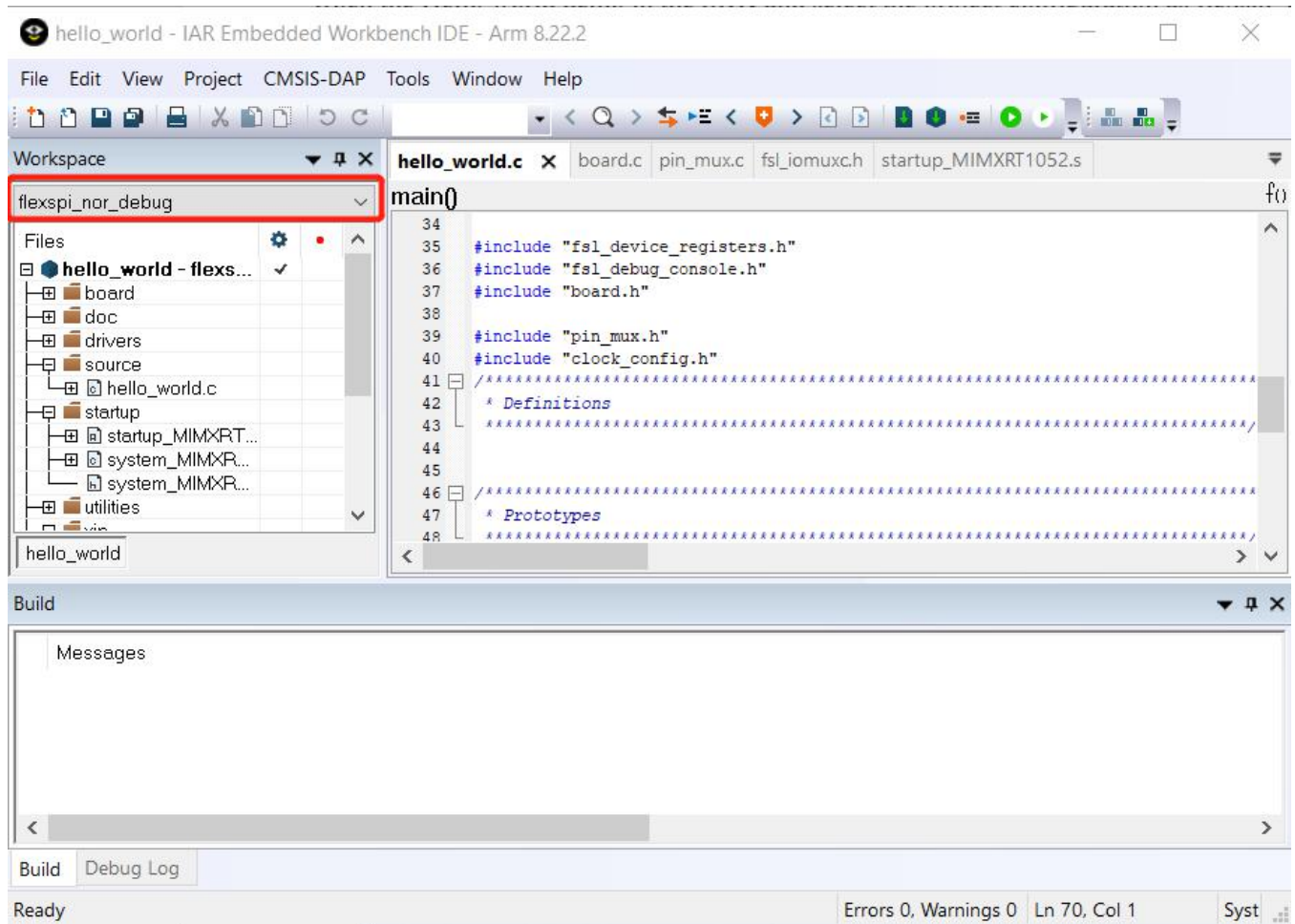


图 4. 选择项目配置为flexspi_nor_debug

步骤 2:

- 构建项目并生成图像。您可以在以下位置找到hello_world.bin (图 5).

注意

在生成图像之前，需要更改flash配置参数。请参考[“How to Enable Debugging for FLEXSPI NOR Flash”, AN12183](#)

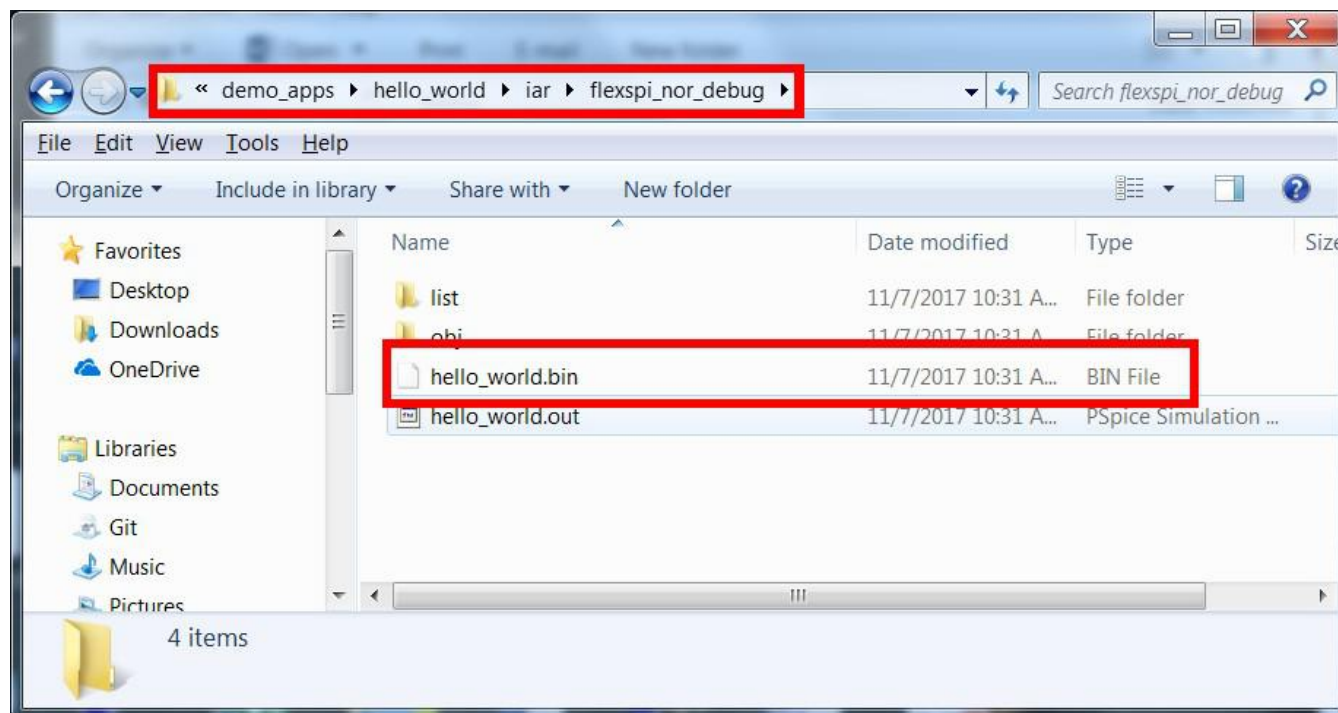


图 5. hello_world.bin 位置

步骤 3:

- 将板配置为串行下载模式，并确保电源来自调试USB。要实现这些，SW7-4应上拉其他下拉 图 6 而J1-5、J1-6应连接 图 7。

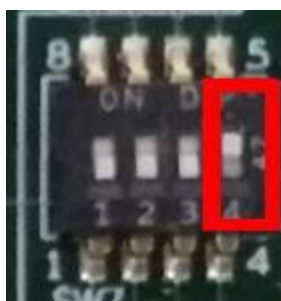


图 6. SW7-4 上拉而其他下拉

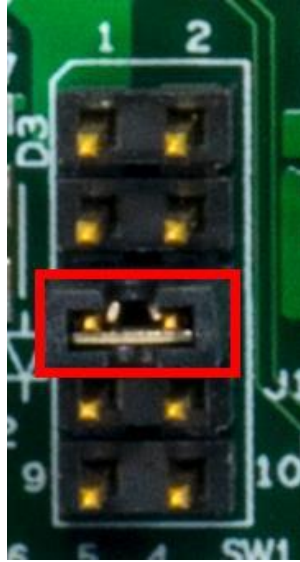
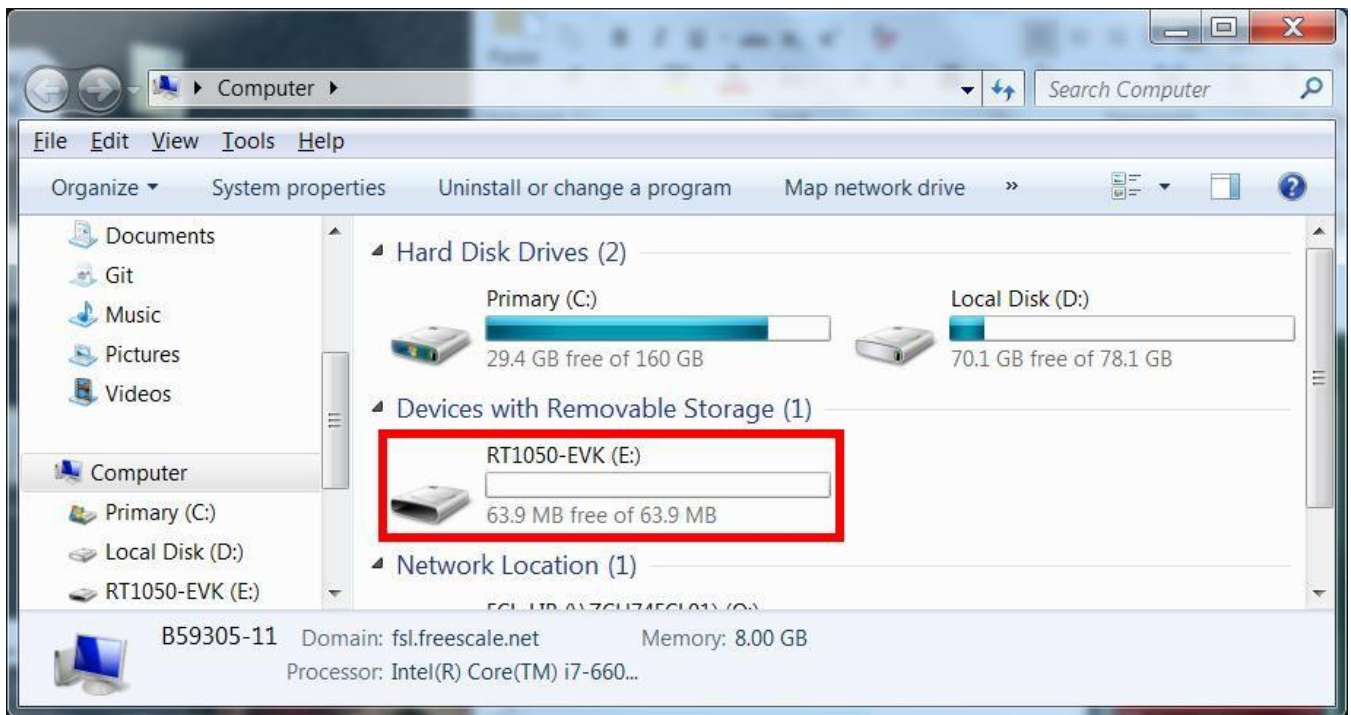


图 7. 电源开关

步骤 4:

- 通过将USB调试电缆连接到J28为板上电，确认U盘显示为如图8所示的驱动器 [图 8](#).

图 8. RT1050-EVK 出现



注意

第一次将MBED USB连接到主机计算机时，Windows会要求安装MBED串行驱动程序。

步骤 5:

- 将hello_world.bin拖放到RT1050-EVK。然后RT1050-EVK消失，几秒钟后它将再次出现。

步骤 6:

- 断开USB调试电缆的连接，然后将开发板配置为QSPI Flash Boot Mode，这意味着SW7-3将上拉，其他上拉 [图 9](#)。



图 9. QSPI Flash 引导模式配置

步骤 7:

- 再次连接USB调试电缆并配置终端窗口。
 - Baud rate: 115200
 - Data bits: 8
 - Stop bit: 1
 - Parity: None
 - Flow control: None

步骤 8:

- 按SW3重置EVK板，终端将打印“hello world”。[图 10](#)

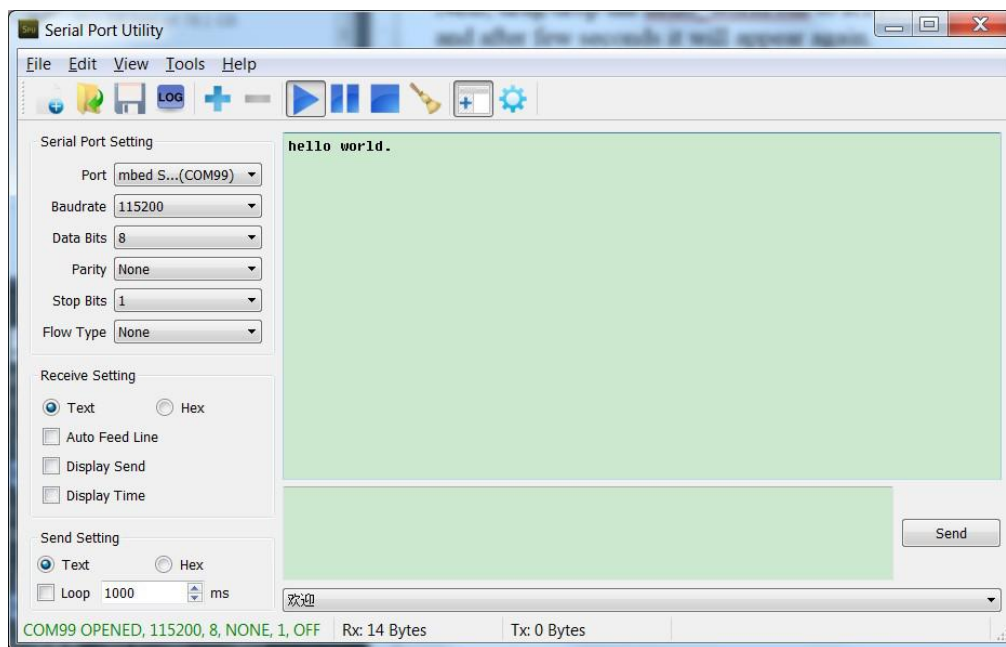


图 10. “hello world” 被打印到终端上

4.2. 从QSPI Flash 引导 MFG

本章介绍了使用MFG工具将映像编程到QSPI Flash并从QSPI Flash引导的步骤。

步骤 1:

- 打开SDK中的Hello world演示，选择项目配置为flexspi_nor_debug，如图11所示，并确保设置如图12所示

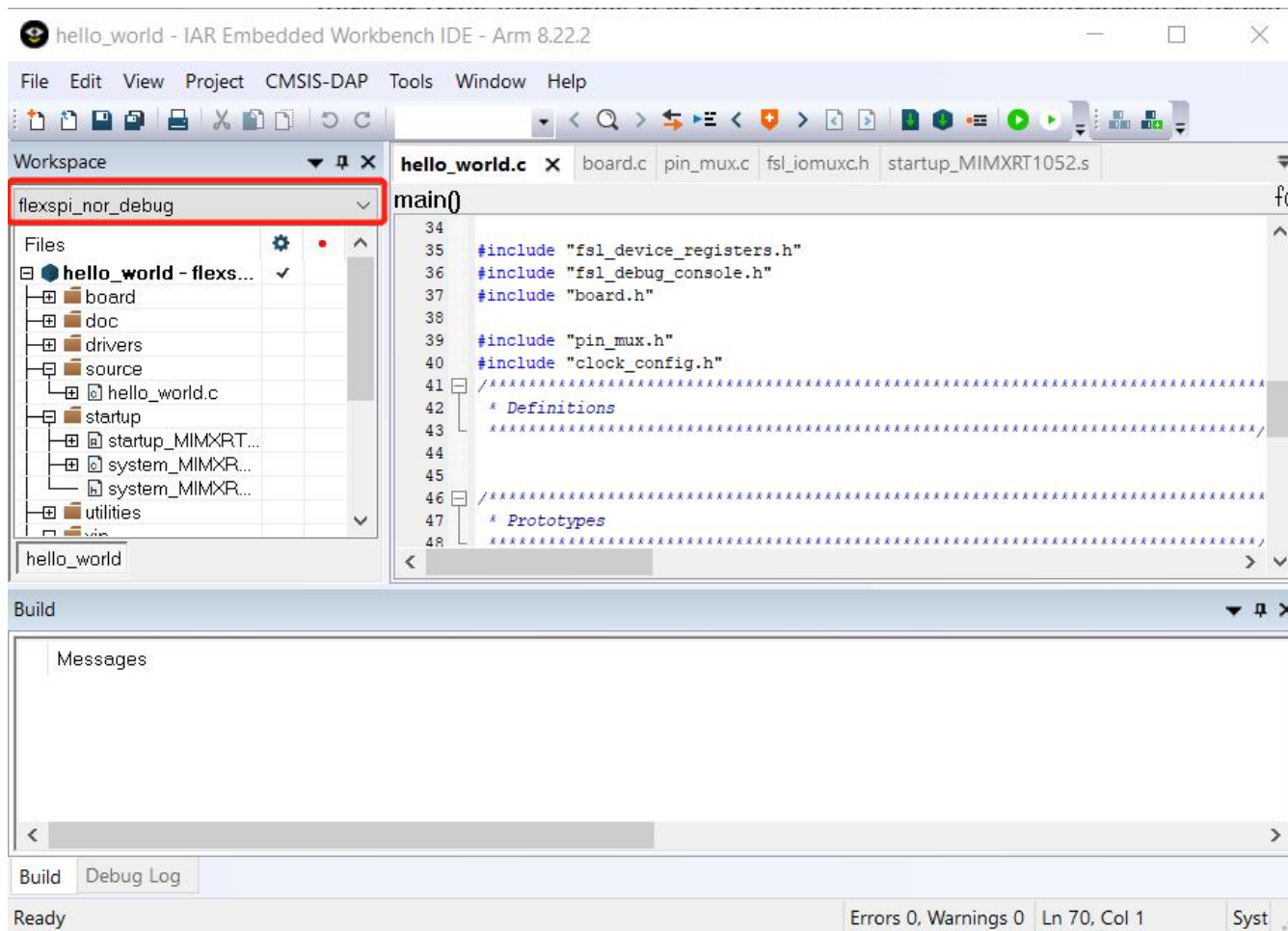


图 11. 选择项目配置为flexspi_nor_debug

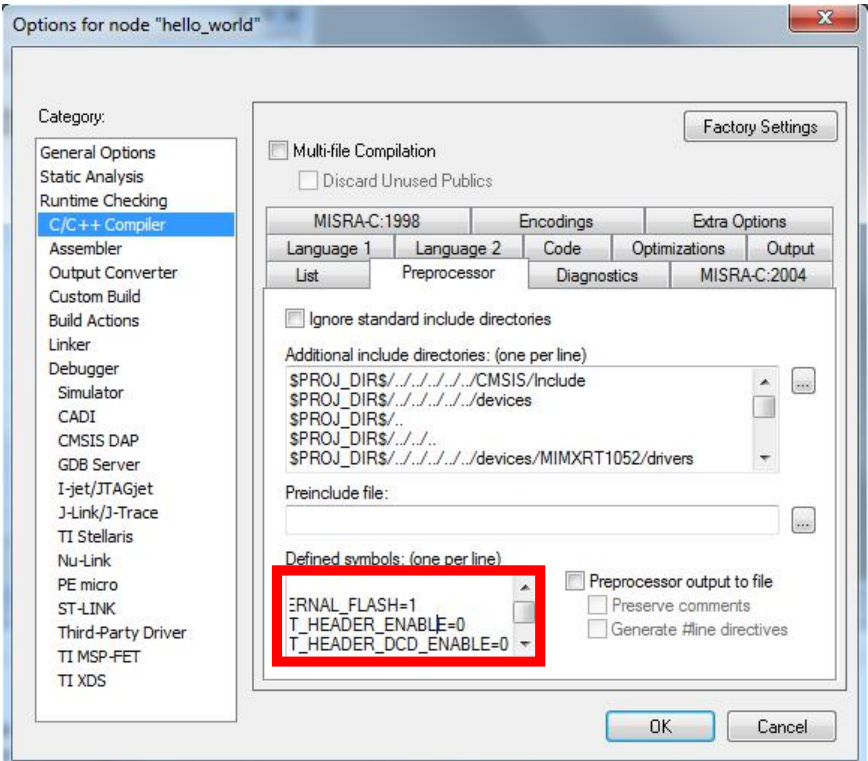


图 12. hello\u world的定义符号

步骤 2:

如下图所示，默认入口地址更改为Reset_Handler。

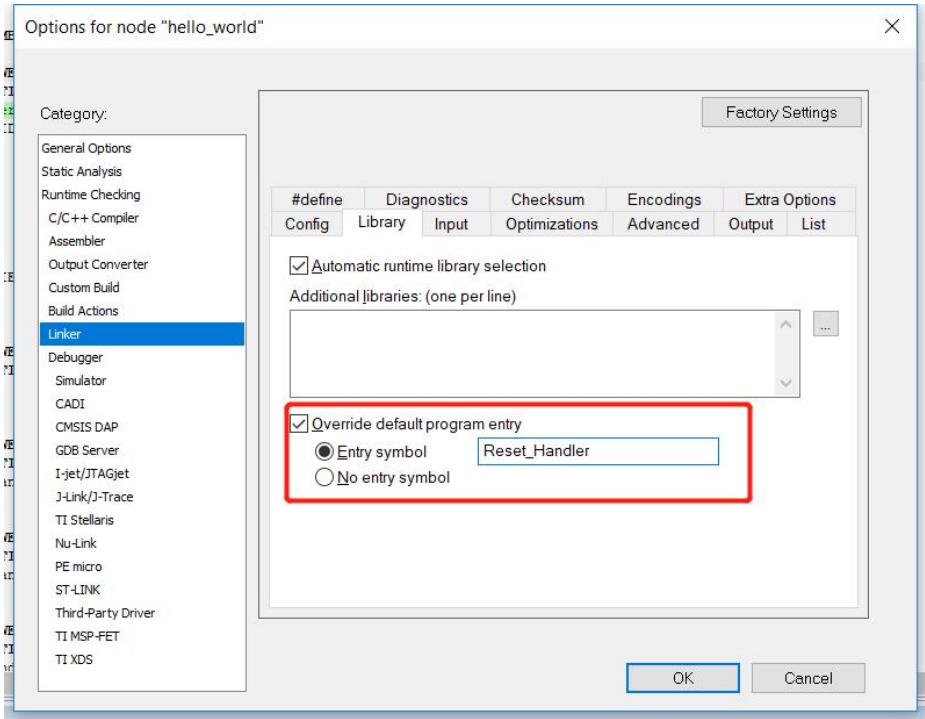


图 13. 将默认条目更改为Reset_Handle

NOTE

如果设置了这步，则可以跳过步骤 5。

步骤 3:

- 生成项目并生成图像。您可以在以下位置找到hello_world.srec
[图 14](#).

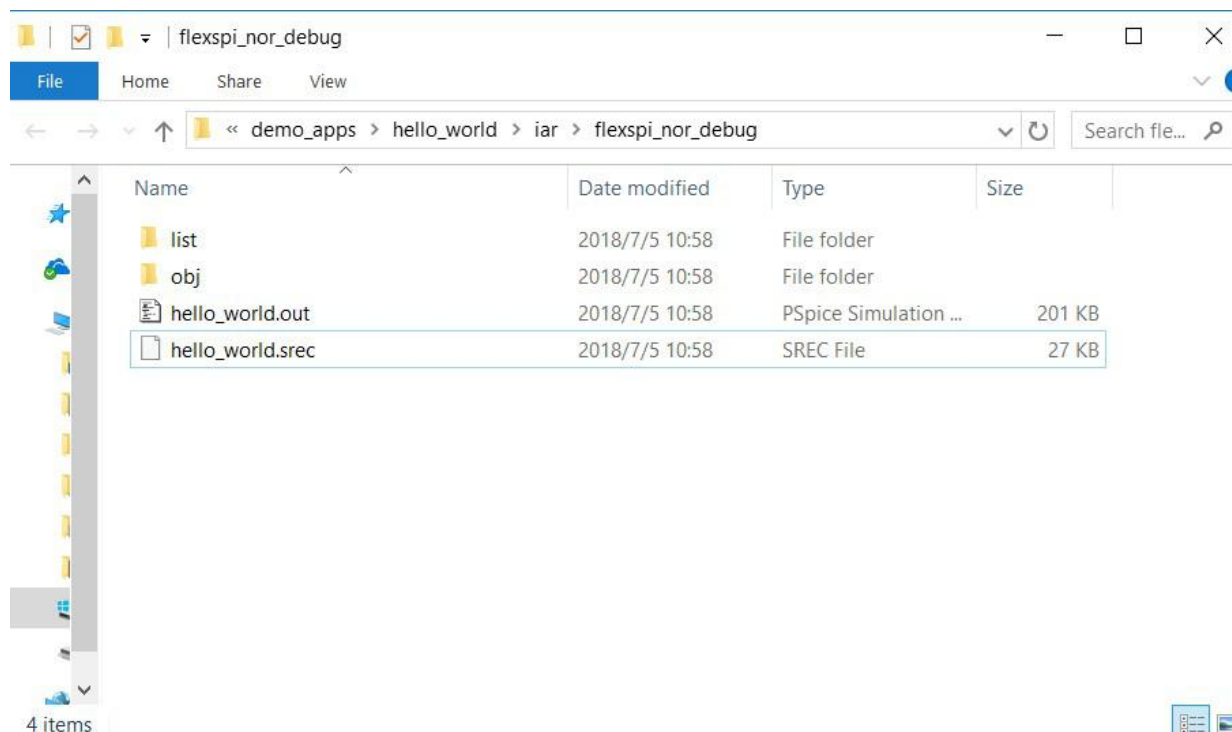
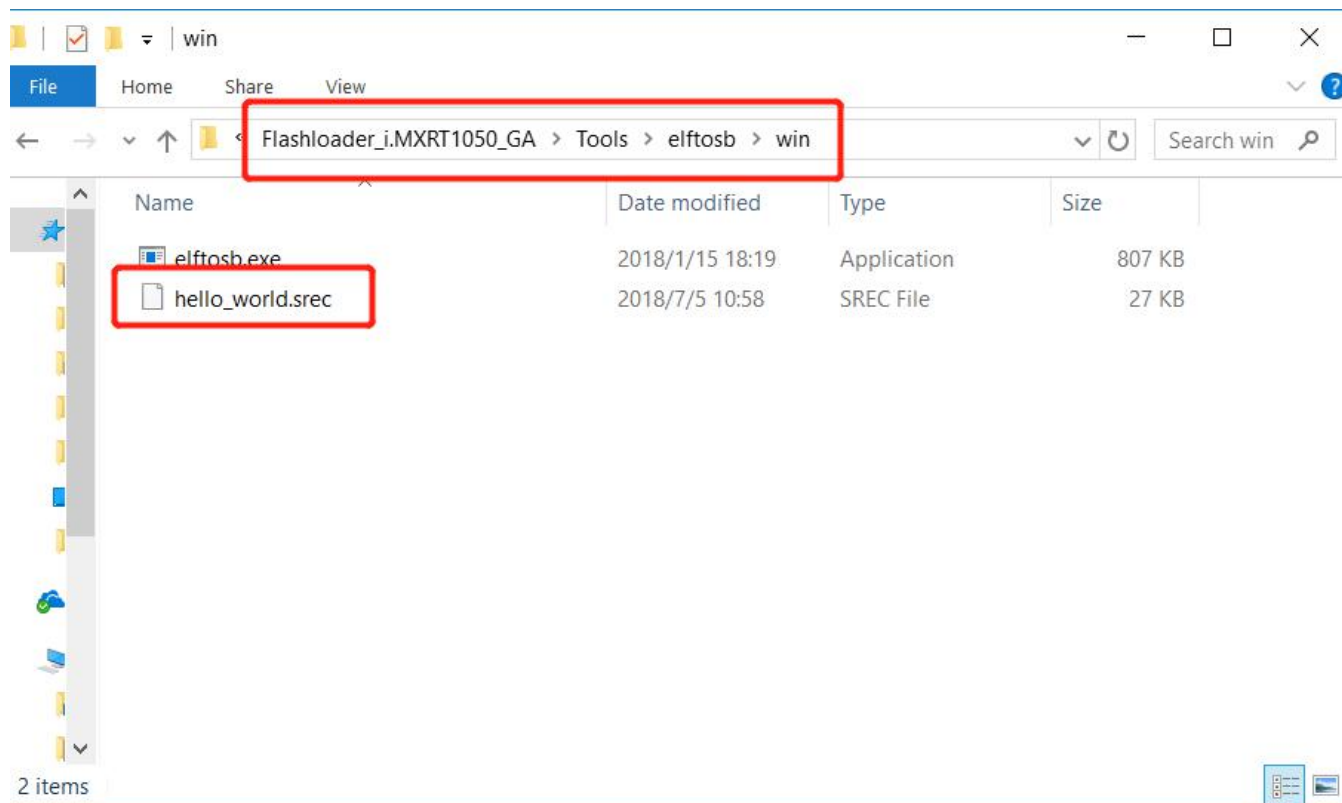


图 14. hello_world.srec location

步骤 4:

- 将hello_world.srec复制到elftosb文件夹:

图 15. 复制 `hello_world.srec`**步骤 5:**

在路径 `Flashloader_i.MXRT1050_GA\Tools\bd_file\imx10xx` 下打开 `imx-flexspinor-normal-unsigned.bd` 打开它，并将 `entryPointAddress` 设置为 `0x60002000` 如下图示

```

1 options {
2     flags = 0x00;
3     startAddress = 0x60000000;
4     ivtOffset = 0x1000;
5     initialLoadSize = 0x2000;
6     # Note: This is required if the default entrypoint is not the
7     #       Please set the entryPointAddress to Reset_Handler add
8     entryPointAddress = 0x60002000;
9 }
10
11 sources {
12     elfFile = extern(0);
13 }
14
15 section (0)
16 {
17 }
18

```

图 16. Set the entryPointAddress to 0x60002000

NOTE

如果设置了此步骤，则可以跳过步骤 2。

步骤 6:

- 现在，我们可以使用命令使用elftosb文件生成i.MX可启动映像。打开cmd.exe并键入以下命令：

```

elftosb.exe -f imx -V -c ../bd_file/imx10xx/imx-flexspinor-normal-unsigned.bd -o
ivt_flexspi_nor_hello_world.bin hello_world.srec

```



```

Windows PowerShell
PS C:\Users\...\Desktop\Flashloader_i.MXRT1050_GA\Flashloader_i.MXRT1050_GA\Flashloa
n> .\elftosb.exe -f imx -V -c ../../bd_file/imx10xx/imx-flexspinor-normal-unsigned.bd -o
ivt_flexspi_nor_hello_world.bin hello_world.srec
Section: 0x0
iMX bootable image generated successfully
PS C:\Users\...\Desktop\Flashloader_i.MXRT1050_GA\Flashloader_i.MXRT1050_GA\Flashloa
der_RT1050_1.1\Tools\elftosb\win>

```

图 17. 生成i.MX可引导映像

执行上述命令后，将生成两个可引导映像：

- ivt_flexspi_nor_hello_world.bin
- ivt_flexspi_nor_hello_world_nopadding.bin

ivt_flexspi_nor_hello_world.bin:

从0到ivt_offset的内存区域用填充字节（全部0x00）填充。

ivt_flexspi_nor_hello_world_nopadding.bin:

直接从ivtdata开始，在ivt之前没有任何填充。

后面的部分将用于生成SB文件，以便在后面进行QSPI Flash编程。

步骤 7:

打开cmd.exe并键入以下命令：

```

elftosb.exe -f kinetis -V -c ../../bd_file/imx10xx/program_flexspinor_image_qspinor.bd -o boot_image.sb
ivt_flexspi_nor_hello_world_nopadding.bin

```

```

Windows PowerShell
PS C:\Users\...\Desktop\Firmware_i.MXRT1050_GA\Firmware_i.MXRT1050_GA\Firmware> .\elftosb.exe -f imx -V -c ../../bd_file/imx10xx/imx-flexspinor-normal-unsigned.bd -o
ivt_flexspi_nor_hello_world.bin hello_world.srec
Section: 0x0
iMX bootable image generated successfully
PS C:\Users\...\Desktop\Firmware_i.MXRT1050_GA\Firmware_i.MXRT1050_GA\Firmware> .\elftosb.exe -f kinetis -V -c ../../bd_file/imx10xx/pr
ogram_flexspinor_image_qspinor.bd -o boot_image.sb ivt_flexspi_nor_hello_world_nopadding.
bin
Boot Section 0x00000000:
FILL | adr=0x00002000 | len=0x00000004 | ptn=0xc0000006
ENA | adr=0x00002000 | cnt=0x00000004 | flg=0x0900
ERAS | adr=0x60000000 | cnt=0x00010000 | flg=0x0000
FILL | adr=0x00003000 | len=0x00000004 | ptn=0xf000000f
ENA | adr=0x00003000 | cnt=0x00000004 | flg=0x0900
LOAD | adr=0x60001000 | len=0x000032b4 | crc=0x9958d743 | flg=0x0000
PS C:\Users\...\Desktop\Firmware_i.MXRT1050_GA\Firmware_i.MXRT1050_GA\Firmware>

```

图 18. 为QSPI Flash编程创建一个SB文件

执行上述命令后，将在elftosb文件夹下生成boot_image.sb，如图19所示。图 19.

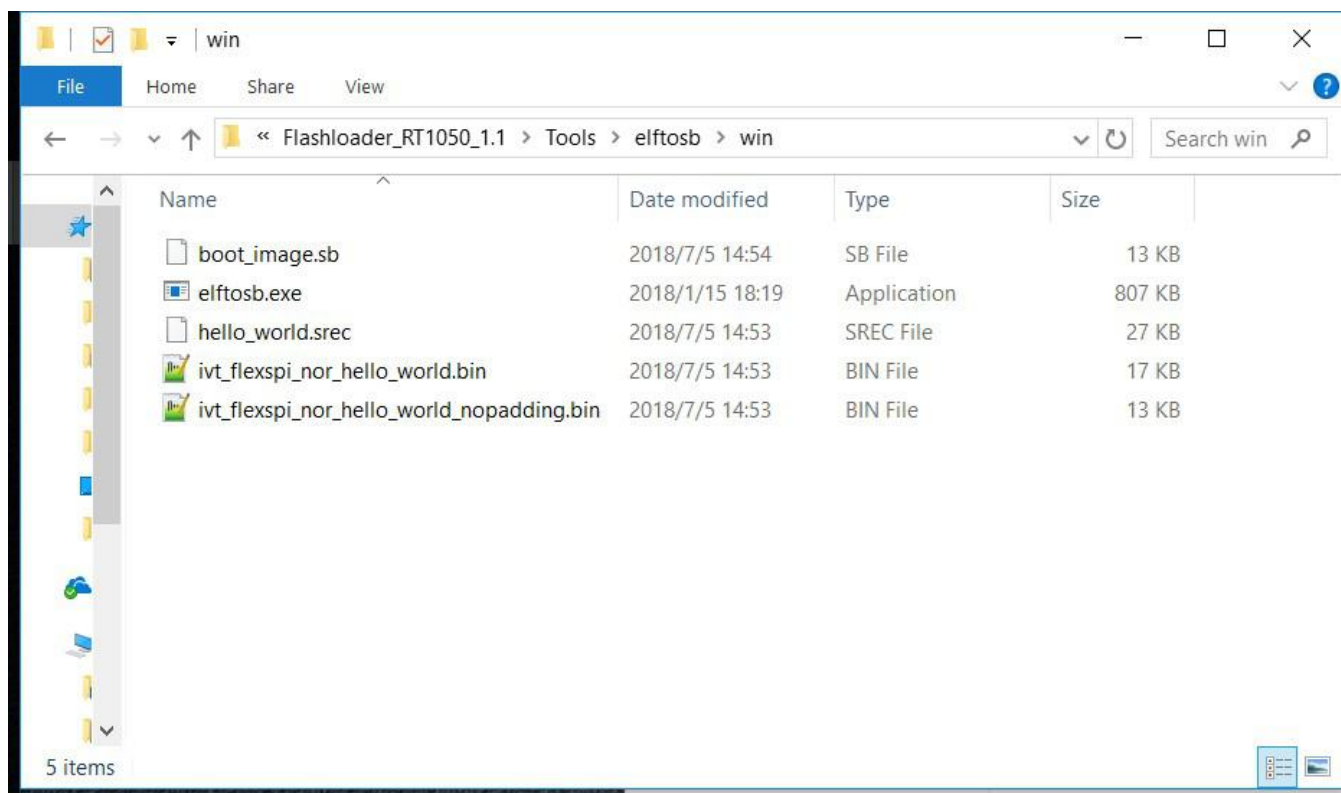


图 19. 生成boot_image.sb

步骤 8:

- 将boot_image.sb文件复制到OS固件文件夹:

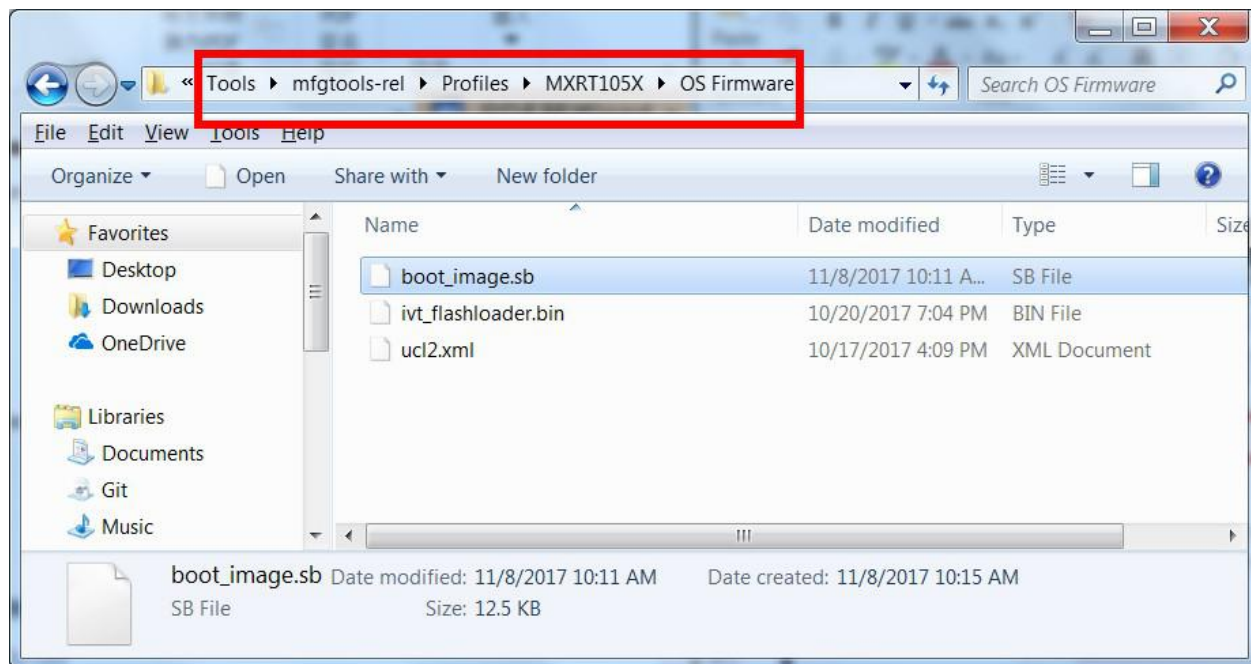


图 20. 复制 boot_image.sb 到 OS Firmware folder

现在确保 *cfg.ini* file下 在<mfgtool_root_dir> 里
“[List]” 的“name” 是“MXRT105x-DevBoot”

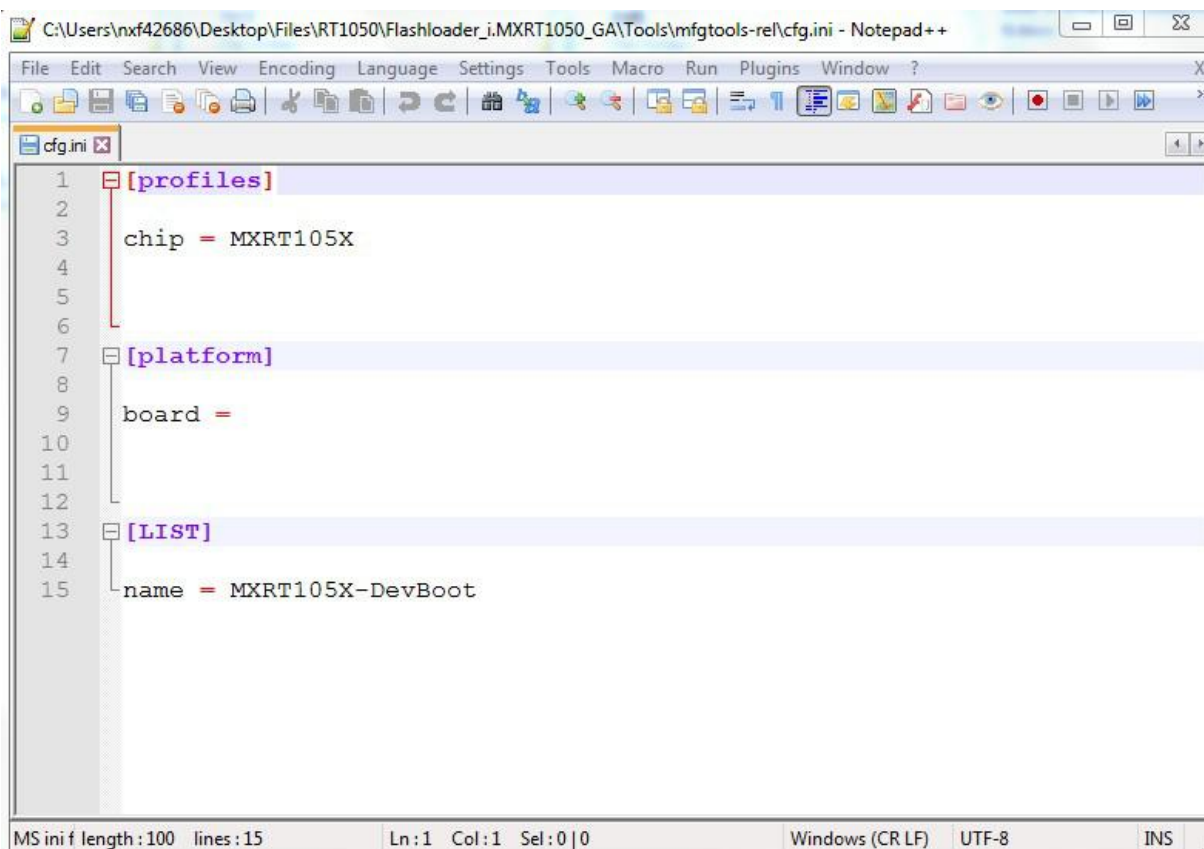


图 21. 确定名称为“MXRT105x-DevBoot”

通过将SW7设置为“1-OFF，2-OFF，3-OFF，4-ON”，将EVK板切换为Serial Downloader模式。将UAB电缆连接到J9，并通过将USB电缆插入J28为EVK板供电。打开MfgTool，将显示检测到的设备，如图22所示。[图 22](#).

NOTE

在某些情况下，无法识别符合HID的设备，这是因为PC仅具有USB根设备而没有USB集线器设备，但此软件限制将在不久的将来得到修复，目前的解决方法是用外部USB集线器作为扩展。

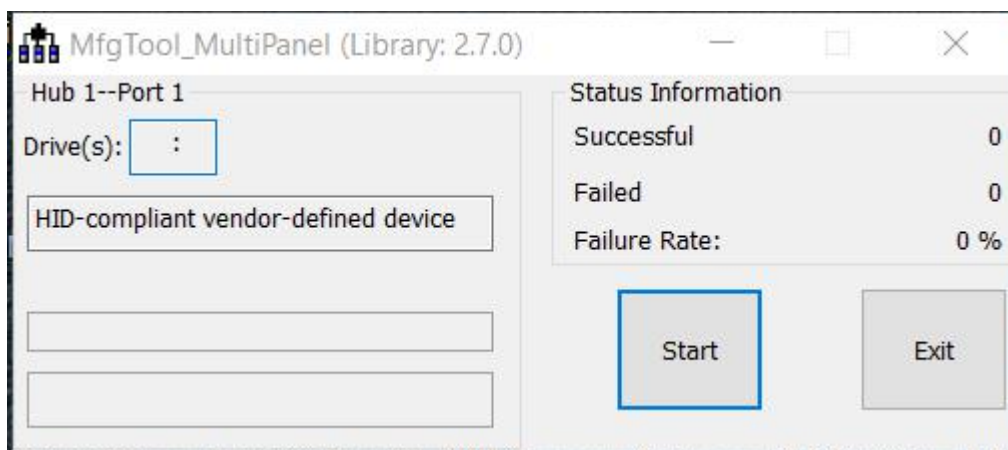


图 22. 已连接设备的MfgTool GUI

单击**Start**。Mfgtool会启动并提示成功状态，如 图 23。单击**Stop**并关闭Mfgtool。

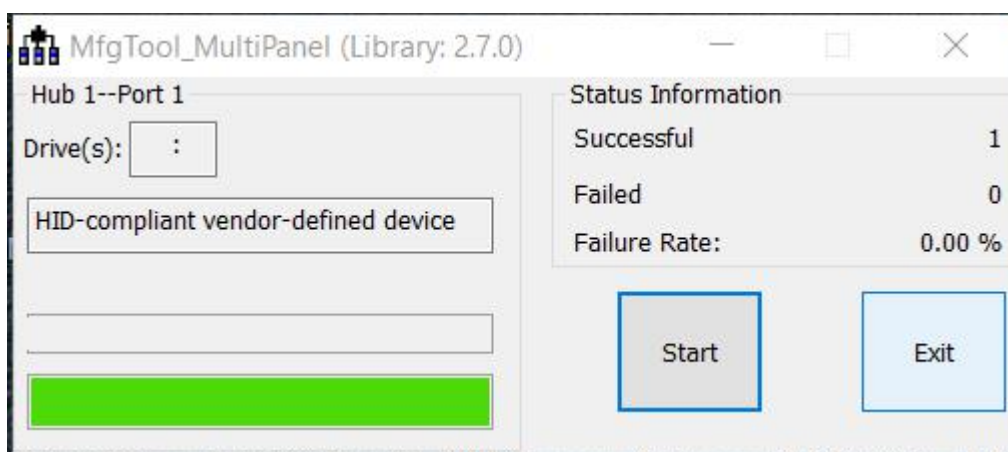


图 23. 成功状态

步骤 9:

- 通过将SW7设置为“1-OFF, 2-OFF, 3-ON, 4-OFF”，将RT1050-EVK板切换到内部引导模式，并选择QSPI Flash作为引导设备。将USB电缆连接到J28并打开终端，然后重置板。我们可以看到“hello world”将被打印到终端上。

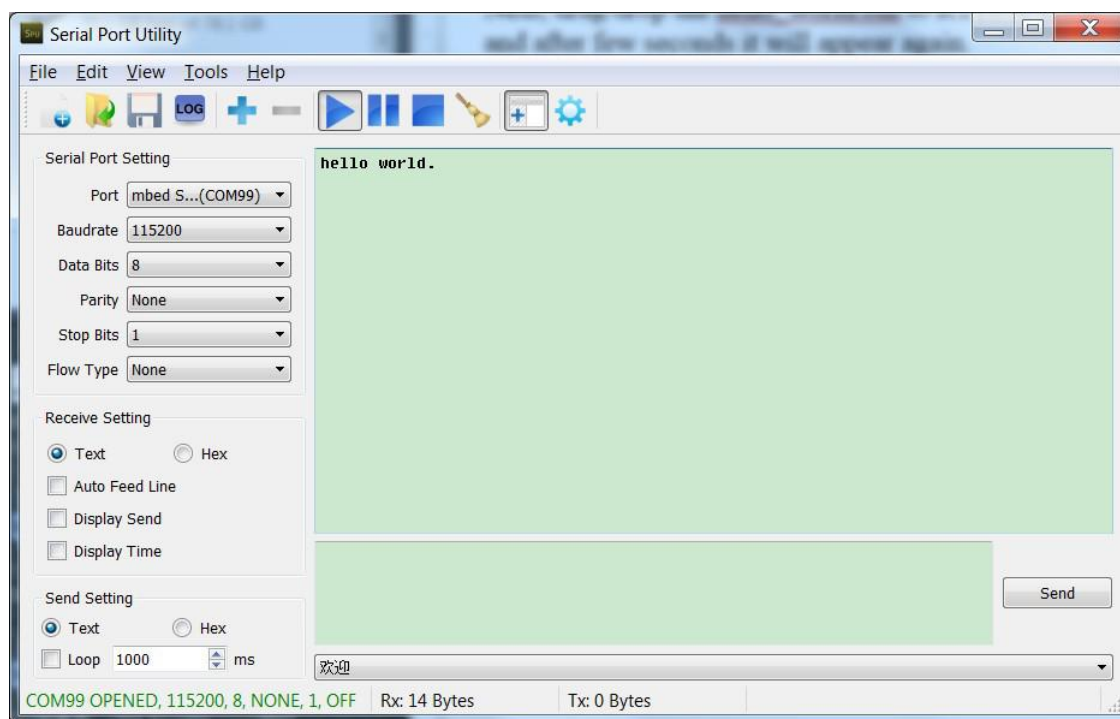


图 24. “hello world” 被打印到终端

4.3. 使用SDRAM的DCD从QSPI Flash引导的MFG

有关步骤请参阅 [“How to Enable Boot from Octal SPIFlash and SD Card”, AN12107](#).

5. QSPI Flash支持列表

除了EVK板载QSPI Flash之外，还支持以下Flash，请注意，这些只是Flash供应商的典型示例，理论上我们可以支持所有符合JESD216/JESD216A/JESD216B的Flash。

同时，RT1050EVK可以通过切换FLASH_VCC电源来支持1.8 V和3.3 V SPI flash设备，如下图25所示。图 25

如果安装了3.3 V SPI flash，则需要安装R301和DNP R49，否则需要安装R49和DNP R301。

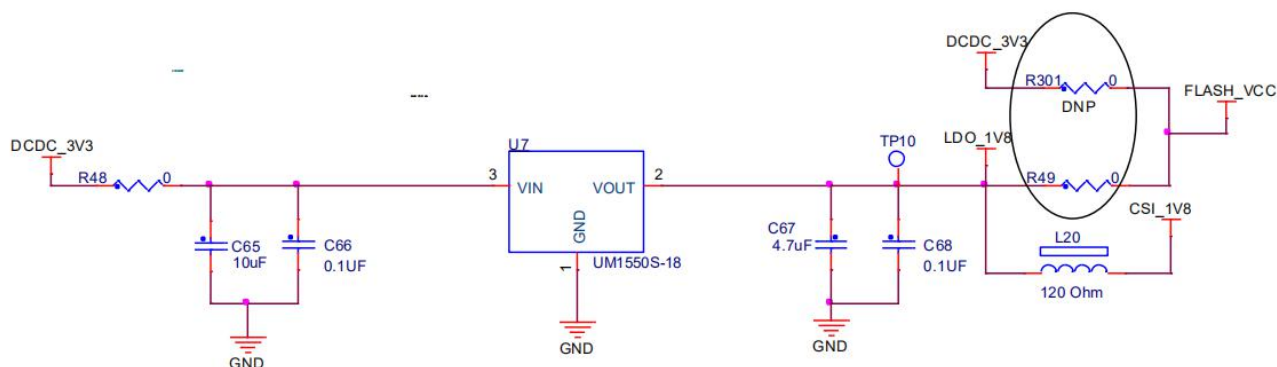


图 25. FLASH_VCC 切换

Table 3. QSPI Flash support list

Vendor	Flash Part Number	Voltage
Macronix	MX25L6433F	3.3 V
Macronix	MX25U6435E	1.8 V
ISSI	IS25LP064A-JBLE	3.3 V
ISSI	IS25WP064AJBLE	1.8 V
GigaDevice	GD25Q64C	3.3 V
GigaDevice	GD25LQ64C, GD25LT256E, GD25LB256E	1.8 V
WINBOND	W25Q64JV	3.3 V
WINBOND	W25Q64FW	1.8 V
Micron	MT25QL128ABA1ESE-0SIT	3.3 V
Micron	MT25QU128ABA1ESE-0SIT	1.8 V
Adesto	AT25QF641-SUB-T	3.3 V
Adesto	AT25QL641-SUE-T	1.8 V

6. 结论

本应用笔记主要介绍如何逐步使用Flashloader。有关更多信息，请参阅[“How to Enable Boot from Octal SPIFlash and SD Card”](#)。

7. 修订历史

Table 4. Revision history

Revision number	Date	Substantive changes
0	12/2017	Initial release
1	06/2018	Adapted SDK version 2.3.1 and Flashloader version 1.1. In Table 1. QSPI Flash support list, changed MX25U6433F to MX25L6433F.
2	07/2018	- Added steps to change the entry address. - Used srec file instead of .out file as the source file.
3	09/2018	Updated Table 3 QSPI Flash support list.
4	07//2019	Updated Table 3 QSPI Flash support list.

如何找到我们:

主页:

nxp.com

网页支持:

nxp.com/support

本文档中的信息仅用于使系统和软件实施者能够使用NXP产品。根据本文档中的信息，根据本文档的设计，没有授予任何明示或暗示的版权许可来设计或制造任何集成电路。恩智浦保留更改的权利，恕不另行通知此处的任何产品。

恩智浦不就其产品在特定目的下的适用性提供任何担保，陈述或担保，也不承担因应用或使用任何产品或电路而引起的任何责任，并且恩智浦明确声明不承担任何责任，包括但不限于结果性或偶然性损害。恩智浦数据表和/或规格中可能提供的“典型”参数在不同的应用中可能会发生变化，并且实际性能可能会随时间变化。必须由客户的技术专家针对每个客户应用验证所有操作参数，包括“典型值”。恩智浦不会根据其专利权或他人的权利转让任何许可。恩智浦根据标准销售条款和条件销售产品，可在以下地址找到该产品：

nxp.com/SalesTermsandConditions.

尽管恩智浦已实现了高级安全功能，但所有产品都可能遭受未识别的漏洞。客户负责其应用程序和产品的设计和操作，以减少这些漏洞对客户的应用程序和产品的影响，并且恩智浦对发现的任何漏洞不承担任何责任。客户应实施适当的设计和操作保护措施，以最小化与其应用程序和产品相关的风险。

恩智浦 (NXP)，恩智浦 (NXP) 徽标，恩智浦 (NXP) 智慧世界的安全连接，COOLFLUX，EMBRACE，GREENCHIP，HITAG，I2C总线，ICODE，JCOP，生活氛围，MIFARE，MIFARE CLASSIC，MIFARE DESFire，MIFARE PLUS，MIFARE FLEX，MANTIS，MIFARE，MIFARE4MOBILE，MIGLO，NTAG，ROADLINK，SMARTLX，SMARTMX，STARPLUG，TOPFET，TRENCHMOS，UCODE，Freescape，Freescape徽标，Altivec，C-5，CodeTEST，CodeWarrior，ColdFire，ColdFire +，C-Ware，节能解决方案徽标，Kinetis，Layerscape，MagniV，mobileni，mobileGT，PEG，PowerQUICC，处理器专家，QorIQ，QorIQ Qonverge，Ready Play，SafeAssure，SafeAssure徽标，StarCore，Symphony，VortiQa，Vybrid，Airfast，BeeKit，BeeStack，CoreNet，Flexis，MXC，打包平台，QUICC引擎，SMARTMOS，塔式，TurboLink，UMEMS，EdgeScale，EdgeLock，eIQ和Immersive3D是NXP BV的商标。所有其他产品或服务名称均为其各自所有者的财产。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是Arm Limited（或其子公司）在美国和/或其他地区的商标或注册商标。相关技术可能受任何或所有专利，版权，设计和商业秘密的保护。版权所有。Oracle和Java是Oracle和/或其分支机构的注册商标。Power Architecture和Power.org文字标记以及Power和Power.org徽标和相关标记是Power.org许可的商标和服务标记。

Document Number: AN12108

Rev. 4

07/2019

