NXP 半导体 应用手册 文章序号: AN12107 Rev. 5, 07/2019

# 如何用八进制SPI闪存和SD卡启用启动

## 1. 简介

i.MX RT系列是NXP提供的业界首个跨界处理器。本 文档介绍如何将可引导映像编程到外部存储设备中。

i.MX RT1050 Flashloader是一个应用程序,可以加载 到i.MX RT1050设备的内部RAM中。Flashloader被设 计成i.MX RT1050设备的第二阶段启动加载程序,它 支持检测一种外设(USB-HID和UART)上的通信流 量,下载用户应用程序,并将应用程序写入外部串行 NOR或串行 NAND Flash设备。Flashloader在第一阶段 由MfgTool加载,在第二阶段与MfgTool一起进行Flash 编程。

该版本包括PC机运行管的MfgTool应用程序,该应用 程序用于在开发阶段和生产阶段将应用程序下载到 Flash设备。此版本还包括elftosb命令行应用程序,它 用于为i.MX RT1050 ROM生成可引导映像,并生成 Flashloader1.1支持的可编程映像。

本文档中使用的示例软件基于i.MXRT1050 SDK 2.4.0。

开发环境为IAR Embedded Workbench 8.22.2,硬件开 发环境为IMXRT1050-EVKB。

## 内容

简介		1
i.MXI	RT1050 启动概述	. 2
2.1.	启动功能	2
2.2.	启动 ROM 概述	3
2.3.	启动相关地址	3
2.4.	启动设置	3
2.5.	引导映像	5
2.6.	图像生成工具	7
程序]	〔具	7
3.1.	DAP-Link (OpenSDA MSD 拖拽/拖放)	7
3.2.	MFG 工具	7
3.3.	OpenSDA 拖拽/拖放 和 从Hyper Flash启动	.9
3.4.	从Hyper Flash启动MFG	15
3.5.	从SD Card启动MFG	26
3.6.	从带有DCD用于SDRAM 35的Hyper Flash 启动	
	MFG	
八进制	引SPI Flash支持列表	46
结论		46
修订员	历史	47
	简介 i.MXH 2.1. 2.2. 2.3. 2.4. 2.5. 2.6. 程序 3.1. 3.2. 3.3. 3.4. 3.5. 3.6. 八进集 修订历	<ul> <li>简介</li></ul>



© 2019 NXP B.V.

本文档描述了三种典型的启动用例:

- SD卡
  - ITCM存储的代码
  - DTCM存储的数据
- Hyper Flash
  - Hyper Flash存储的代码XIP
  - DTCM存储的数据
- 启用了SDRAM的Hyper Flash (带有DCD)
  - Hyper Flash存储的代码XIP
  - SDRAM存储的数据

## 2.i.MXRT1050 启动概述

## 2.1. 启动功能

启动过程从上电复位(POR)开始,在此硬件复位逻辑强制ARM内核从片上启动ROM开始执行。 启动ROM使用BOOT\_MODE寄存器和eFUSEs的状态来确定启动设备。出于开发目的,可以使 用GPIO引脚输入来覆盖用于确定启动设备的eFUSEs。启动ROM代码还允许下载要在设备上运 行的程序。该示例是一个配置程序,可以进一步利用串行连接为启动设备提供新的映像。

## 2.1.1. 设备配置数据 (DCD)

DCD功能允许启动ROM代码从驻留在启动设备上的外部程序映像获取SOC配置数据。例如, DCD可用于对SDRAM控制器(SEMC)进行编程以实现最佳设置,从而提高启动性能。DCD仅 限于内存区域和外设地址,这些地址对于启动来说是必不可少的。

## 2.1.2. 安全启动(高安全启动)

在HAB允许执行用户映像之前,必须对映像进行签名。签名过程是在图像构建过程中由私钥持 有者完成的,然后将签名作为最终程序图像的一部分包含在内。如果配置为这样做,则ROM使 用程序映像中包含的公共密钥来验证签名。除了支持数字签名验证来验证程序映像之外,还支 持加密启动。加密的启动可用于防止直接从启动设备克隆程序映像。除串行下载程序以外,还 可以在芯片上支持的所有启动设备上执行HAB的安全启动。

#### i.MXRT1050 启动概述

启动ROM中的HAB库还提供API功能,从而允许其他启动链组件(启动加载程序)扩展安全启动链。

## 2.2. 启动ROM概述

启动Rom的主要功能包括:

支持从各种启动设备启动 支持串行下载器(USB OTG 和 UART) 设备配置数据(DCD)和插件 基于数字签名和加密的高安全性启动(HAB) 从低功耗模式唤醒 通过由总线加密引擎(BEE)供电的FlexSPI接口在串行NOR上就地加密(XIP) 数据协同处理器(DCP)控制器(串行NOR除外)上的设备上的加密引导启动

Rom支持以下启动设备:

通过FlexSPI的串行NOR闪存 通过FlexSPI串行NAND闪存 通过智能外部存储器控制器(SEMC)并行NOR闪存 通过SEMC的RAW\_NAND闪存 SD/MMC SPI NOR/EEPROM

als la d

## 2.3. 启动相关地址

		Table I.	<b>石列相大地址</b>
起始地址	结束地址	大小	说明
0x80000000	0xDFFFFFFF	1.5GB	SEMC 外部存储器(SDRAM, NOR, PSRAM, NAND and 8080) 共享内存空间
0x60000000	0x7F7FFFFF	504MB	FlexSPI/FlexSPI 密码文本
0x20200000	0x2027FFFF	512KB	OCRAM
0x20000000	0x2007FFFF	512KB	DTCM
0x00000000	0x0007FFFF	512KB	ITCM

┍) - H H + H + H

## 2.4. 启动设置

通过在POR\_B的上升沿对BOOT\_MODE0和BOOT\_MODE1输入进行采样来初始化 BOOT\_MODE,并将其存储在内部BOOT\_MODE寄存器中(可以从SRC\_SBMR2[BMOD[1: 0]]中读取)。

启动模式[1:0]	启动类型
00	从保险丝启动
01	串行下载器(来自USB或UART)
10	内部启动(继续从内部启动ROM执行启动代码)
11	预留

Table 2. 启动模式设置引脚

#### 注意

从保险丝启动类似于内部启动模式,但有一个区别:

在这种模式下,GPIO引导覆盖引脚将被忽略。引导ROM代码仅 使用引导eFUSE设置。

对于这四种引导模式(其中一种保留给NXP使用)。根据内部BOOT\_MODE寄存器中存储的二进制值选择引导模式。开关(SW7-3和SW7-4)用于选择MIMXRT1050 EVK板上的启动模式。

启动模式[1:0] (SW7-3 SW7-4)	启动类型
00	从保险丝启动
01	串行下载器
10	内部启动
11	预留

Table 3. 基于MIMXRT1050-EVK的启动模式设置引脚

通常,选择内部引导进行常规引导,这是由外部BOOT\_CFG GPIO配置的。表4显示了典型的启动模式和启动设备设置。

SW7-1	SW7-2	SW7-3	SW7-4	启动设备
OFF	ON	ON	OFF	Hyper Flash
OFF	OFF	ON	OFF	QSPI Flash
ON	OFF	ON	OFF	SD Card

Table 4. 典型的启动模式和启动设备设置

i.MXRT1050 启动概述

#### 注意

有关启动模式配置的更多信息,请参见《<u>IMXRT 1050 Reference</u> <u>Manual</u>》中的"系统启动"一章。

有关MIMXRT1050 EVK启动设备选择和配置的更多信息,请参阅 主板原理图(<u>main board schematic</u>)

## 2.5. 引导映像

i.MX MCU可引导映像有两种类型:

正常引导映像:此类映像可以通过启动ROM直接启动。

插件引导映像:此类型的映像可用于从启动ROM本身不支持的设备中加载引导映像。 可以针对不同的生产阶段和不同的安全级别要求对两种类型的图像进行不签名,签名和加密:

未签名的映像: 该映像不包含与身份验证相关的数据, 在开发阶段使用。

签名图像: 该图像包含与身份验证相关的数据(CSF部分),在生产阶段使用。

加密映像: 该映像包含加密的应用程序数据和与身份验证相关的数据, 在生产阶段使用, 对安全性有更高的要求。

引导映像包括:

图像向量表(IVT): 位于固定地址的指针列表, ROM会检查这些地址以确定程序图像 其他组件的位置。

启动数据:该表指示程序映像位置,程序映像大小(以字节为单位)和插件标志。

设备配置数据 (DCD): IC配置数据 (例如: SDRAM寄存器配置).

用户代码和数据

CSF (可选):由CST生成的安全启动签名块。

• KeyBlob (可选) – 数据结构由包装的DEK组成,用于加密启动。

每个可引导映像均以适当的IVT开头。通常,对于支持XIP功能的外部存储设备,IVT偏移为0x1000,否则为0x400。例如,对于RT1052上的FlexSPI NOR,IVT必须从地址0x60001000开始(起始地址为0x6000 0000,IVT偏移为0x1000)。

#### i.MXRT1050 启动概述



Figure 1. 可引导映像布局

## 2.5.2. IVT 数据结构

Table 5. IVT 数据结构

偏移	场所	说明	
0x00 - 0x03	标头	字节0 标签, 固定为 0xD1         字节长度 1,2位字节序格式,包含IVT的总长度(以字节为单位),固定为 0x00、0x20         字节 3: 形式,有效值: 0x40, 0x41, 0x42, 0x43	
0x04 - 0x07	条目	图像执行的第一条指令的绝对地址或图像的向量地址	
0x08 - 0x0b	预留1	保留供将来使用,设置为0	
0x0c - 0x0f	dcd	图像DCD的绝对地址。它是可选的,因此如果不需要DCD,则可以将该字 段设置为NULL。	
0x10 - 0x13	开机数据	启动数据的绝对地址	
0x14 - 0x17	自身	IVT的绝对地址	
0x18 - 0x1b	csf	HAB库使用的命令序列文件(CSF)的绝对地址	
0x1c - 0x1f	预留2	保留,设置为0	

## 2.5.3. 启动数据结构

#### Table 6.启动数据结构

偏移	场所	说明
0x00-0x03	开端	可引导映像的绝对地址
0x04-0x07	长度	可引导映像的大小
0x08-0x0b	插件	插件标志,如果它是正常的引导映像,则设置为0

## 2.6. 图像生成工具

Elftosb实用程序是一个命令行主机程序,用于为i.MX MCU引导ROM生成i.MX可引导映像。 Elftosb工具支持SREC输入程序映像。

它还可以使用相应的选项和适当的命令文件(称为BD文件)生成带有命令序列和可引导映像 (称为SB文件)的包装二进制文件。(使用此.sb文件的MFGTool)

有关BD文件的更多详细信息,请参考<u>i.MX MCU 制造用户指南(第4.1章)</u>。如何为未签名的正常/签名的正常/加密的正常/插件的可启动映像生成可启动映像,您可以参考<u>i.MX MCU 制造用</u>户指南(第4.2章)。

## 3. 编程工具

## 3.1. DAP-Link (OpenSDA MSD 拖放)

- 仅限EVK上的Hyper Flash/QSPI闪存
- 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闪存 拖放,则应更换DAP-Link的固件。

## 3.2. MFG 工具

MfgTool支持I.MXRT BootROM和基于KBOOT的Flashloader,可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"对连接到I.MX MCU设备的 目标存储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(是MfgTool版本的一部分)的设备连接。仅支持sb文件。

对MFG:

• cfg.ini

配置要使用的设备,线路板和程序列表(在ucl2.xml中)

• ucl2.xml

正在加载Flashloader

程序启动映像

- MfgTool.log
   以备故障的详细记录
- boot\_image.sb
   引导映像放入"OS固件"文件夹中





## 3.2.1. 引导标头的宏

表7显示了在flexspi\_nor目标中添加的三个宏以支持XIP:

Table 7. 引导标头的宏

宏命令	说明
XIP_EXTERNAL_FLASH	1:使用XIP代码。 O:RAM运行。
XIP_BOOT_HEADER_ENABLE	1: 默认情况下,向映像添加flexspi配置块、映像向量表、引导数据和设备配置数据(可选)。 0: 默认情况下,不向图像添加任何内容。

XIP\_BOOT\_HEADER\_DCD\_ENABLE

1: 将设备配置数据添加到映像。 0: 不要将设备配置数据添加到映像。

表8显示了这些宏的不同组合对构建图像的不同影响:

		XIP_BOOT_HEADER_DCD_ENA BLE=1	XIP_BOOT_HEADER_DCD_ENA BLE=0		
(TERNAL_FLASH=1	XIP_BOOT_HEA DER_ENABLE=1	可以通过IDE编程为Hyper Flash, 并且如果Hyper Flash是启动源,则 可以在POR重置后运行。 SDRAM将被初始化。	可以通过IDE编程为Hyper Flash, 并且如果Hyper Flash是启动源,则 可以在POR重置后运行。 SDRAM将 <b>不会</b> 初始化。		
XIP_EX	XIP_BOOT_HEA DER_ENABLE=0	如果POR重置后由IDE编程,即使Hyper Flash是引导源,也 <b>不能</b> 运 行。			
XIP_EXTERNAL_FLASH =0		该映像 <b>无法</b> 执行XIP,因为当此宏设置为1时,它将排除将更改FlexSPI 时钟的代码。			

## Table 8. 不同宏对构建图像的不同影响

## 3.3. OpenSDA拖拽/拖放和从Hyper Flash引导

本章将详细介绍使用OpenSDA拖放将图像编程到Hyper Flash的详细步骤。步骤如下:

## 步骤 1:

在SDK中打开Hello world演示,然后选择项目配置为flexspi\_nor\_debug。



Figure 3. 选择项目配置为flexspi\_nor\_debug

### 步骤 2:

构建项目并生成图像。您可以找到hello\_world.bin,如图4所示。

Contraction Of Cases	· And South New Yorks			X
😋 🔵 🔻 📙 « demo_apps	hello_world → iar → flexspi_nor_debug	► <b>~ ~</b>	earch flexspi_nor_de	ebug 🔎
<u>File Edit View Tools H</u> elp				
Organize 🔹 Include in libra	ary   Share with   New folder		•	0
🔺 Favorites 🔺	Name	Date modified	Туре	Size
📕 Desktop	📙 list	11/7/2017 10:31 A	File folder	
🐌 Downloads 🗏	- obj	11/7/2017 10·21 A	File folder	
ConeDrive	hello_world.bin	11/7/2017 10:31 A	BIN File	
Libraries Documents Git Music	hello_world.out	11/7/2017 10:31 A	PSpice Simulatic	on
4 items				

Figure 4. hello\_world.bin 地址

## 步骤 3:

将开发板配置为串行下载模式,并确保电源来自调试USB。要实现这些目标,SW7-4应该将 其他下拉电路上拉(图5),并且将J1-5,J1-6连接起来(图6)。



Figure 5. SW7-4 上拉和其他下拉





## 步骤 4:

现在,我们可以通过将USB调试电缆连接到J28并使用电脑USB端口为板供电,并确认U盘作为驱动器出现,如图7所示。



Figure 7. RT1050-EVK <u>盘符</u>出现

### 注意

第一次将MBED USB连接到主机时,Windows将要求安装MBED串行 驱动程序。

## 步骤 5:

将hello\_world.bin拖放到RT1050-EVK。 然后RT1050-EVK消失并在几秒钟后再次出现。

## 步骤 6:

断开USB调试电缆的连接,然后将开发板配置为Hyper Flash Boot Mode,这意味着SW7-2和SW7-3上拉其他下拉模块,如图8所示。



Figure 8. Hyper Flash启动模式配置

再次连接USB调试电缆并配置终端窗口:

波特率:115200

数据位:8

终止位:1

奇偶校验:无

流量控制:无

按SW3复位EVK板,"hello world"将被打印到终端,如图9所示。



Serial Port Utility	and after few seconds it will appear again.	
<u>File Edit View Tools H</u> elp		
- 🕂 🔜 📮 🥪		
Serial Port Setting	hello world.	
Port mbed S(COM99) V		
Baudrate 115200 -		
Data Bits 8		
Parity None 🔻		
Stop Bits 1		
Flow Type None		
Receive Setting		
) Text 🔘 Hex		
Auto Feed Line		
Display Send		
🗾 Display Time		
Send Setting		Send
Text  Hex		
🔲 Loop 1000 🚔 ms	欢迎	<b></b> ]
COM99 OPENED, 115200, 8, NONE, 2	I, OFF Rx: 14 Bytes Tx: 0 Bytes	th.

Figure 9. 输出Hello world

## 3.4. 从Hyper Flash启动MFG

本章介绍如何使用MFG Tool将映像编程到Hyper Flash并从Hyper 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(图10),并确保设置如图11 所示。

Workspace		<b>→</b> ậ	hello_world.c × board.c pin_mux.c fsl_iomuxc.h startup_MIMXRT1052.s	
flexspi_nor_debug			main()	
Files  Files Files Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  File	*	•	<pre>35 #include "fsl_device_registers.h" 36 #include "fsl_debug_console.h" 37 #include "board.h" 38 39 #include "pin_mux.h" 40 #include "clock_config.h" 41 = /***********************************</pre>	······································
Build Messages				× 1





Category:					Fa	actory Setti	ngs
General Options Static Analysis	Multi-file Con	npilation Unused Publics					
Runtime Checking	Diagnostics	MISRA-C:2004	MISRA-C:1998	Encod	dings	Extra Op	tions
C/C++ Compiler	Language 1	Language 2 Code	Optimizations	Output	List	Preproc	essor
Build Actions Linker Debugger Simulator CADI CMSIS DAP GDB Server	\$PROJ_DIR \$PROJ_DIR \$PROJ_DIR \$PROJ_DIR \$PROJ_DIR \$PROJ_DIR \$PROJ_DIR	s////CMSIS/In \$////devices \$// \$/// \$///devices/I	olude MIMXRT1052/driv	rers		~	
I-jet/JTAGjet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver	Defined sym G (TERNAL_F )OT_HEADE )OT_HEADE	bols: (one per line) LASH=1 ER_ENABLE=0 ER_DCD_ENABLE=		processo Preserve Generate	<b>r output</b> <u>c</u> omme #line di	to file ents rectives	

Figure 11. hello\_world的定义符号

## 步骤 2:

如下图所示,将默认条目更改为Reset\_Handler。

Category: Fac Seneral Options Static Analysis Runtime Checking C/C++ Compiler Config Library Input Optimizations Advanced Our Assembler Output Converter Custor Build Additioned libraries: (one part line)	tory Settings xtra Options out List
Seneral Options Static Analysis Runtime Checking C/C++ Compiler Config Library Input Optimizations Advanced Our Assembler Output Converter Custom Build Additional libraries (one per line)	xtra Options out List
itatic Analysis     #define     Diagnostics     Checksum     Encodings     Encodings	xtra Options
#define     Diagnostics     Checksum     Encodings     Encodings       C/C++ Compiler     Config     Library     Input     Optimizations     Advanced     Out       Assembler     Output Converter     Automatic runtime library selection     Custom Build     Additioned libraries (one part line)	xtra Options out List
C/C++ Compiler Config Library Input Optimizations Advanced Our Assembler Output Converter Automatic runtime library selection Custom Build	out List
Assembler Output Converter Qutput Converter Qustom Build Additional libraries (one part line)	
Output Converter Automatic runtime library selection	
Custom Build Additional libration: (and par line)	
Additional initialies, tone per men	
Build Actions	
Linker	· · · ·
Debugger	
Simulator	$\sim$
CADI	6
CMSIS DAP	
GDB Server	
I-jet/JTAGjet	
J-Link/J-Trace	
TI Stellaris	
Nu-Link	
PE micro	
STJINK	
51 Elive	
Third-Party Driver	

Figure 12. 将默认条目更改为Reset\_Handler

注意

如果设置了此步骤,则可以跳过步骤5。

## 步骤 3:

生成项目并生成.srec格式的图像。您可以找到hello\_world.srec,如图 13所示:

	= flexspi_nor_debug						×
File	Home Share View						~ 🕐
$\leftarrow \rightarrow$		> flexspi_nor_debug		~ (	)	Search fle	<u>م</u>
^	Name	Date modified	Туре	Size			13
*	📜 list	2018/7/5 10:58	File folder				
6	📜 obj	2018/7/5 10:58	File folder				
	hello_world.out	2018/7/5 10:58	PSpice Simulation		201 KI	В	
	hello_world.srec	2018/7/5 10:58	SREC File		27 K	В	
ų							
4							
<u>a</u> ~							
4 items							

Figure 13. hello\_world.srec 地址

## 步骤 4:

将hello\_world.srec复制到elftosb文件夹:

	≂   win			- 0	×
File H	łome Share View				~ 🕐
$\leftarrow \rightarrow$ v	・ 个 📜 < Flashloader_i.MXRT1	050_GA > Tools > elftosb > win		<ul><li>ン O Search win</li></ul>	Q
^	Name	Date modified	Туре	Size	
× -	elftosb.exe	2018/1/15 18:19	Application	807 KB	
	hello_world.srec	2018/7/5 10:58	SREC File	27 KB	
ļ					
<u>هم</u>					
<b>.</b> .					
1~					
2 items					

Figure 14. 复制hello\_world.srec

## 步骤 5:

## 打开 *imx-flexspinor-normal-unsigned.bd*

路径为Flashloader\_i.MXRT1050\_GA\Tools\bd\_file\imx10xx. 打开它并将entryPointAddress设置为

### 0x60002000 如下图所示



Figure 15. 将 entryPointAddress 设置为 0x60002000

### 注意

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elftosb文件命令生成i.MX可引导映像。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* 



Figure 16. 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的内存区域填充字节(全部为0x00)。

ivt\_flexspi\_nor\_hello\_world\_nopadding.bin:

直接从ivtdata开始,无需在ivt之前进行任何填充。

后面的部分将用于为Hyper FLASH编程生成SB文件。。

## 步骤 7:

这一步,我们将创建用于Hyper Flash编程的SB文件。将生成一个boot\_image.sb文件,供 MfgTool稍后使用。打开cmd.exe并键入以下命令:

elftosb.exe -f kinetis -V -c ../../bd\_file/imx10xx/program\_flexspinor\_image\_HyperFlash.bd -o boot\_image.sb ivt\_flexspi\_nor\_hello\_world\_nopadding.bin

🔀 Windo	ows PowerShell		1-1	$\Box$ $\times$
<pre>➢ Winde PS C:\Us sb\win&gt; erFlash. Boot See FILL ENA ERAS FILL ENA LOAD PS C:\Us sb\win&gt;</pre>	bws PowerShell sers\	<pre>ttop\Flashloader_: kinetis -V -c, sb ivt_flexspi_nd len=0x00000004 cnt=0x00000000000000000000000000000000000</pre>	i. MXRT1050\Flashloader_i. MXRT1050_GA\T //bd_file/imx10xx/program_flexspinor or_hello_world_nopadding.bin   ptn=0xc0233007 flg=0x0900 flg=0x0000 ptn=0xf0000000f flg=0x0900 crc=0x7270d9b5   flg=0x0000 i. MXRT1050\Flashloader_i. MXRT1050_GA\T	ools\elfto _image_Hyp
				~

Figure 17. 创建用于Hyper Flash编程的SB文件

执行上述命令后,将在elftosb文件夹下生成 boot\_image.sb。

☑ I =   win Home Share View			-	
→ 👻 🛉 📕 « Flashloader_i.MXRT1050_GA > Too	ols > elftosb > wir	1	マレ Sear	ch win 🔎
Name	Date modified	Туре	Size	
boot_image.sb	2018/7/5 11:27	SB File	13 KB	
elftosb.exe	2018/1/15 18:19	Application	807 KB	
hello_world.srec	2018/7/5 10:58	SREC File	27 KB	
ivt_flexspi_nor_hello_world.bin	2018/7/5 11:28	BIN File	17 KB	
ivt_flexspi_nor_hello_world_nopadding.bin	2018/7/5 11:28	BIN File	13 KB	
1				
×				
ms				

Figure 18. 生成boot\_image.sb

## 步骤 8:

将 boot\_image.sb 文件复制到OS固件文件夹:

1 1	= = OS Firmware			3. <del></del> 3	
File	Home Share View				~ 🕐
$\leftarrow \rightarrow$	✓ ↑	> MXRT105X > OS	Firmware	∨ Ŭ Sea	rch OS 🔎
^	Name	Date modified	Туре	Size	
	boot_image.sb	2018/7/5 11:27	SB File	13 KB	
<u> </u>	📔 ivt_bootdata.bin	2017/12/6 18:10	BIN File	1 KB	
	📔 ivt_flashloader.bin	2018/2/11 8:52	BIN File	88 KB	
	ucl2.xml	2018/2/11 8:55	XML Document	8 KB	
1					
1					
4					
4 itoms	1 item selected 12.0 KP				
4 items	Them selected 12.3 KD				

Figure 19. 将boot\_image.sb文件复制到OS固件文件夹:

现在,

确保<mfgtool\_root\_dir>文件夹下cfg.ini文件中"[List]"下的"name"改为"MXRT105x-DevBoot"归档。

File Edit Search View Encoding	Language Settings Tools	Macro Run F	lugins Window	2	
	1 D C 8 %	*   🖪 🗟   🖻	ə ¶ 👍 🖾 🛛	7) 🖬 👁 🛛	
🔚 cfg.ini 🗵					4
1 [profiles]					
2 3 chip = MXRT105 4	х				
5 6 7 <b>₽[platform]</b>					
8 9 board = 10 11					
12 L 13 ⊟[LIST] 14					
15 name = MXRT105	X-DevBoot				

Figure 20. 确保名字改为 "MXRT105x-DevBoot"

通过将SW7设置为"1-OFF, 2-OFF, 3-OFF, 4-ON",将EVK板切换为串行下载模式。将UAB 电缆连接到J9,并通过将USB电缆插入J28来给EVK板上电。

打开MfgTool,将显示检测到的设备,如图21所示:

nformation ful
ful
Rate: 0
Start Exit

## Figure 21. 连接了设备的 MfgTool GUI

单击"开始",Mfgtool进程启动。完成后,MfgTool将显示成功状态,如图22所示。单击"停止"并关闭Mfgtool。

Hub 3Port 2	Status Information	
Drive(s):	Successful	1
	Failed	0
Done	Failure Rate:	0.00 %

Figure 22. MfgTool 成功状态

### 步骤9:

通过将SW7设置为"1-OFF, 2-ON, 3-ON, 4-OFF",将RT1050-EVK板切换到内部启动模式,并选择Hyper FLASH作为启动设备。将USB电缆连接到J28并打开一个端子,然后重置开发板。终端上将印有"hello world"。

Serial Port Utility		
<u>File E</u> dit <u>V</u> iew <u>T</u> ools <u>H</u> elp		
👝 🍋 🔚 🚾 🕂 —		
Serial Port Setting	hello world.	
Port mbed S(COM99) 💌		
Baudrate 115200 💌		
Data Bits 8		
Parity None 🔻		
Stop Bits 1		
Flow Type None		
Receive Setting		
Text  Hex		
Auto Feed Line		
Display Send		
Display Time		
Send Setting	Send	
Text Hex		
🔲 Loop 1000 🌩 ms	家迎	
COM99 OPENED, 115200, 8, NONE,	1, OFF Rx: 14 Bytes Tx: 0 Bytes	t.

Figure 23. "hello world" 被打印到终端上

## 3.5. 从SD卡启动MFG

本章将介绍使用MFG工具将映像编程到SD卡并从SD卡启动的步骤。

步骤 1:

Shello_world - IAR Embedded Workbench	IDE - Arm 8.22.2					
File       Edit       View       Project       CMSIS-DAP <td <="" <td="" th=""><th>Tools Window</th><th>Help - &lt; Q</th><th>&gt; \$ \$ • 5 &lt;</th><th>🖸 &gt; हे 🦻 ।</th><th>8 🖷 🖸 🚏</th></td>	<th>Tools Window</th> <th>Help - &lt; Q</th> <th>&gt; \$ \$ • 5 &lt;</th> <th>🖸 &gt; हे 🦻 ।</th> <th>8 🖷 🖸 🚏</th>	Tools Window	Help - < Q	> \$ \$ • 5 <	🖸 > हे 🦻 ।	8 🖷 🖸 🚏
Workspace	▼ ¤ ×					
	·					
Files	<b>\$</b>					
⊢⊞ ∎ board	•					
H doc						
	•					
Hand Startup	•					
L 🖽 📹 Output						
hello_world						
Debug Log					<b>▲</b> ☆ ×	
Log						
Wed Apr 18, 2018 15:08:18: IAR Er	nbedded Workbe	ench 8.22.2 (C:\U	sers\nxf42686\De	esktop\IAR\arm\bin\a	rmproc.dll)	
Wed Apr 18, 2018 15:08:18: Loadir	ig the CMSIS-DAI	<sup>o</sup> driver				
Build Debug Log						
Ready			-	L	n 1, Col 1	

在SDK中打开Hello world演示,然后选择项目配置作为Debug。

Figure 24. 选择项目配置作为调试

## 步骤 2:

如下图所示,将默认条目更改为Reset\_Handler。

Category:						Factory	Settings
General Options							
Static Analysis							
tuntime Checking	#define	Diag	nostics	Checksum	Encodings	Extra	Options
C/C++ Compiler	Config	Library	Input	Optimizations	Advanced	Output	List
Assembler Output Converter	Autor	actic runtin	ao libronu	aclastian			
Custom Build	Mulon	auc runun	le library s	selection			
Build Actions	Addition	al libraries	: (one per	line)			_
							N 1000
Linker						· · · · · · · · · · · · · · · · · · ·	
Linker Debugger							
Linker Debugger Simulator							,
Linker Debugger Simulator CADI							
Linker Debugger Simulator CADI CMSIS DAP	<u> ⊘ver</u>	ide defaul	tprogram	entry			
Linker Debugger Simulator CADI CMSIS DAP GDB Server	  	ide defaul	t program	entry Reset_Handler		ן רו	/
Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet	<mark>⊻ Qverr</mark> ⊚ <u>E</u> ○ N	ide defaul ntry symbo o entry syr	t program	entry Reset_Handler		<u>ן</u>	/
Linker Debugger Simulator CADI CMSIS DAP GDB Server I_jet/JTAGjet J-Link/J-Trace	<mark>⊻ Qverr</mark> ⊚ E ○ N	ide defaul ntry symbo o entry syr	t program )I [ nbol	entry Reset_Handler		]	
Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris	⊻ <u>o</u> ver ⊛ ⊆ ○ N	ide defaul ntry symbo o entry syr	t program al [ nbol	entry Reset_Handler		]	
Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris Nu-Link	⊻ <u>o</u> verr ⊚ <u>E</u> ○ N	ide defaul ntry symbo o entry syr	t program )I I nbol	entry Reset_Handler		]	/
Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris Nu-Link PE micro	⊻ <u>o</u> verr ⊛ E ○ N	ide defaul ntry symbo o entry syr	t program )I [ nbol	entry Reset_Handler		]	/
Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK	⊻ <u>o</u> verr ⊚ E ○ N	ide defaul ntry symbo o entry syr	t program )I [ nbol	entry Reset_Handler		]	/
Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver	⊻ Qverr ⊚ E ○ N	ide defaul ntry symbo o entry syr	t program >I [ nbol	entry Reset_Handler		]	,
Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET	⊻ <u>O</u> verr	ide defaul ntry symbo o entry syr	t program )l [ nbol	entry Reset_Handler		]	/

Figure 25. 将默认条目更改为Reset\_Handler

## 注意

如果设置了此步骤,则可以跳过步骤6。

## 步骤 3:

找到链接文件MIMXRT1052xxxxx\_ram.icf并将起始向量表从0x0000A000更改。

define	symbol	m_interrupts_start	=	0x0000A000;
define	symbol	m_interrupts_end	=	0x0000A3FF;
define	symbol	m_text_start	=	0x0000A400;
define	symbol	m_text_end	=	0x0001FFFF;
define	symbol	m_data_start	=	0x20000000;
define	symbol	m_data_end	=	0x2001FFFF;
define	symbol	m_data2_start	=	0x20200000;
define	symbol	m_data2_end	=	0x2023FFFF;

Figure 26. 将起始向量表从0x0000A000更改

## 步骤 4:

Name	Date modified	Туре	Size
📙 list	2018/7/5 13:22	File folder	
📕 obj	2018/7/5 13:22	File folder	
E hello_world.out	2010/7/5 13.22	PSpice Simulation .	
hello_world.srec	2018/7/5 13:22	SREC File	27 KB

构建项目并生成图像。您可以在以下位置找到 hello\_world.srec:



## 步骤 5:

将 hello\_world.srec 复制到 elftosb 文件夹:

	📜 🔻 🛛 win					1777).	
File	Home Share	View					~ ?
$\leftarrow \rightarrow$	🗸 🛧 📕 < Fla	ashloader_i.MXR	T1050_GA > 1	Гools → elftosb → win		∨ Č Se	arch win 🔎
^	Name	X		Date modified	Туре	Size	
×	elftosb.exe	_		2018/1/15 18:19	Application	807 KB	
	hello_world	l.srec		2018/7/5 10:58	SREC File	27 KB	
2 items	1						

Figure 28. 复制 hello\_world.srec

## 步骤 6:

打开*Flashloader\_i.MXRT1050\_GA\Tools\bd\_file\imx10xx*路径下的 *imx-itcm-unsigned.bd*。 打开它并将entryPointAddress设置为0x0000A000,如下图所示。



Figure 29. 将entryPointAddress 设置为 0x0000A000

#### 注意

如果设置了此步骤,则可以跳过步骤2。

#### 步骤 7:

现在,我们可以使用elftosb文件生成i.MX可引导映像。打开cmd.exe并键入以下命令:

*elftosb.exe -f imx -V -c ../../bd\_file/imx10xx/imx-itcm-unsigned.bd -o ivt\_itcm\_hello\_world.bin hello\_world.srec* 



Figure 30. 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\_itcm\_hello\_world.bin

ivt\_itcm\_hello\_world\_nopadding.bin

ivt\_flexspi\_nor\_hello\_world.bin:

从0到ivt\_offset的内存区域填充字节(全部为0x00)。

ivt\_flexspi\_nor\_hello\_world\_nopadding.bin:

直接从ivtdata开始,无需在ivt之前进行任何填充。

后面的部分将用于为SD卡编程生成SB文件。

### 步骤 8:

这一步,我们将创建用于SD卡编程的SB文件。将生成一个boot\_image.sb文件,供MfgTool稍后使用。打开cmd.exe并键入以下命令:

*elftosb.exe -f kinetis -V -c ../../bd\_file/imx10xx/program\_sdcard\_image.bd -o boot\_image.sb ivt\_itcm\_hello\_world\_nopadding.bin* 

PS C:\Users\\Desktop\Flashloader_i.MXRT1050\Flashloader_i.MXRT1050_GA\Tools\elfto sb\win> .\elftosb.exe -f imx -V -c//bd_file/imx10xx/imx-itcm-unsigned.bd -o ivt_itcm _hello_world.bin hello_world.srec Section: 0x0 iMX bootable image generated successfully PS C:\Users\\Desktop\Flashloader_i.MXRT1050\Flashloader_i.MXRT1050_GA\Tools\elfto sb\win> \elftosb.exe -f kinetis -V -c _/ /bd_file/imx10xx/program_sdcard_image_bdo b
sot_image.sb ivt_itcm_hello_world_nopadding.bin         Boot Section 0x000000000         FILL       adr=0x00000100         FILL       adr=0x00000104         Philoson       ptn=0x0000000         FILL       adr=0x00000104         Philoson       ptn=0x0000000         ENA       adr=0x00000104         Ena       cnt=0x00000004         ENA       adr=0x00000100         cnt=0x0000100       cnt=0x0000004         ERAS       adr=0x00000400         cnt=0x000013c00       flg=0x2010         LOAD       adr=0x00000400         len=0x000041e0       crc=0xc631921f         flg=0x2010       crc=0xc631921f         Solvers\t

Figure 31. 创建用于SD卡编程的SB文件

执行上述命令后,在elftosb文件夹下生成boot\_image.sb。

	ois > eittosb > wir	1	V O Searc
Name	Date modified	Туре	Size
boot_image.sb	2018/7/5 11:27	SB File	13 KB
elftosb.exe	2018/1/15 18:19	Application	807 KB
hello_world.srec	2018/7/5 10:58	SREC File	27 KB
ivt_flexspi_nor_hello_world.bin	2018/7/5 11:28	BIN File	17 KB
ivt_flexspi_nor_hello_world_nopadding.bin	2018/7/5 11:28	BIN File	13 KB

Figure 32. 生成 boot\_image.sb

## 步骤 9:

将boot\_image.sb文件复制到OS固件文件夹:

aner	-	Test Trans	
Correction of the second secon	mfgtools-rel      Profiles      MXRT:	105X 🕨 OS Firmware 👻 🍫	Search OS Firmware
<u>File Edit View 1001s</u>	<u>н</u> еір		
Organize 🔹 📄 Ope	n Share with ▼ New folder		iii • 🔟 🔞
🚖 Favorites	A Name	Date modified	Type Size
E Desktop	boot_image.sb	11/8/2017 10:11 A	SB File
🔈 Downloads	ivt_flashloader.bin	10/20/2017 7:04 Pt	M BIN File
ConeDrive	ucl2.xml	10/17/2017 4:09 PM	M XML Document
门 Libraries			
Documents			
🛃 Git			
🕹 Music		III	•
boot_imag SB File	ge.sb Date modified: 11/8/2017 10: Size: 12.5 KB	11 AM Date created: 11/8/2017 10:1	5 AM

Figure 33. 将boot\_image.sb复制到OS固件文件夹

现在,确保<*mfgtool\_root\_dir*>文件夹下cfg.ini文件中"[List]"下的"name"改为"**MXRT105x-DevBoot**" 归档。

C:\Users\nxf42686\Desktop\Files\RT1	050\Flashloader_i.MXRT1050_GA\Tools\r	nfgtools-rel\cfg.ini - Notepad+-	+	
File Edit Search View Encoding	Language Settings Tools Macro	Run Plugins Window ?	🖸 💿 🔳 🛙	
🔚 cfg.ini 🔀				4
1 [profiles]				
2 3 chip = MXRT105X 4				
5 6				
8 [platform]				
9 board = 10 11 12				
13 [LIST]				
14 15 name = MXRT105X	-DevBoot			
MS ini f length : 100 lines : 15	Ln:1 Col:1 Sel:0 0	Windows (CR LF)	UTF-8	INS

Figure 34. 确保名字改为 "MXRT105x-DevBoot

通过将SW7设置为"1-OFF, 2-OFF, 3-OFF, 4-ON",将SD卡插入J20插槽,并将EVK板切换至 串行下载模式。将UAB电缆连接到J9,并通过将USB电缆插入J28来为EVK板供电。

打开MfgTool,将显示检测到的设备,如图35所示:

MfgTool_MultiPanel (Library: 2.7.0)	0 <u>1-1</u> 3	$\square$ ×
Hub 1Port 1	Status Information	
Drive(s):	Successful	0
	Failed	0
HID-compliant vendor-defined device	Failure Rate:	0 %
	Start	Exit

Figure 35. 连接了设备的MfgTool GUI

单击"开始"。启动Mfgtool进程。一旦完成,MfgTool将显示成功状态,如图36所示。单击"停止" 并关闭Mfgtool。

nfgTool_MultiPanel (Library: 2.7.0)	9 <u>1-1</u> 9	
Hub 1Port 1	Status Information	
Drive(s):	Successful	1
	Failed	0
HID-compliant vendor-defined device	Failure Rate:	0.00 %
	Start	Exit

Figure 36. MfgTool 成功状态

### 步骤 10:

将RT1050-EVK板切换到内部引导模式,通过将SW7设置为"1-ON, 2-OFF, 3-ON, 4-OFF"选择 SD卡作为引导设备。将USB线连接到J28并打开一个终端,然后重置开发板。"hello world"将被 打印到终端上。

Serial Port Utility	
<u>File Edit View Tools H</u> elp	
Serial Port Setting	hello world.
Port mbed S(COM99)	
Baudrate 115200	
Data Bits 8	
Stop Bits 1	
Flow Type None	
Receive Setting	
Text  Hex	
Auto Feed Line	
Display Send	
Display Time	
Send Setting	Send
Loop 1000	awa
COM99 OPENED, 115200, 8, NONE,	۲۸:0 Bytes Tx: 0 Bytes

Figure 37. "hello world" 被打印在终端上

## 3.6. 用带有DCD的Hyper Flash为SDRAM进行MFG引导

本章将介绍使用MFG工具将映像编程到Hyper Flash并从Hyper Flash引导的步骤。

## 步骤 1:

在SDK中打开Hello world演示,并选择项目配置为flexspi\_nor\_debug(Figuire 38),并确保设置如图39所示。





Category:				ſ	Factor	y Settings
General Options	Multi-file Com	npilation				
Static Analysis Runtime Checking	Discard	Unused Publics				
C/C++ Compiler	MISRA-C:	1998	Encodings		Extra Op	otions
Assembler	Language 1	Language 2	Code	Optimiza	ations	Output
Output Converter	List	Preprocessor	Diagnos	tics	MISRA	A-C:2004
Custom Build			-			
Build Actions	Ignore star	ndard include direc	ctories			
Linker	المحين المحيد وتقاولو ال	Jack Hardwood (				
	Additional Incl	Line directones: In	ne ner ine i			
Debugger			IS /Include			
Debugger Simulator	SPROJ_DIRS	ude directories: (o \$/////CMS \$/////devi	ilS/Include ces			<b>^</b>
Debugger Simulator CADI	SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS	ude directories: (o \$////CMS \$///devia \$/	ilS/Include ces			
Debugger Simulator CADI CMSIS DAP	SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS	ude directones: (o \$////CMS \$////devin \$/ \$/	ilS/Include ces			<b>^</b>
Debugger Simulator CADI CMSIS DAP GDB Server	SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS	ude directories: (o \$/////CMS \$/ \$/ \$/// \$/////devi	ne per line) iIS/Include ces ces/MIMXRT1	05 <mark>2/d</mark> rivers	s	•
Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet	SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS Preinclude file	ude directones: (o \$/.//.//MS \$////devii \$/ \$////devii s:	ne per line) SIS/Include ces ces/MIMXRT1	05 <mark>2/d</mark> rivers	\$	•
Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace	SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS Preinclude file	uae arectones: (o \$/././//CMS \$/././/devi \$//. \$///. \$/////devi	ilS/Include ces ces/MIMXRT1	05 <mark>2/d</mark> rivers	5	•
Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris	SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS Preinclude file	uae airectones: (o \$//././CMS \$////devi \$// \$////devi \$////devi	ilS/Include ces ces/MIMXRT1	05 <mark>2/drivers</mark>	S	•
Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris Nu-Link	PROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS Preinclude file	uae anectones: (0 \$//././CMS \$//././devin \$//./. \$//././devin :: ols: (one per line)	ilS/Include ces ces/MIMXRT1	052/drivers	5	•
Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris Nu-Link PE micro	PROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS Preinclude file Defined symbol	uae airectones: (o \$//././CMS \$// \$// \$// \$////devin s: ols: (one per line)	IS/Include ces ces/MIMXRT1	052/drivers	s output to	file
Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK	Additional Inci SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS Preinclude file Defined symbol RNAL_FLAS	Ude airectones: (0 \$//././CMS \$//./. \$//./. \$//./. \$//././devines: (000 per line) SH=1	ilS/Include ces ces/MIMXRT1	052/drivers processor c	s output to	file
Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver	PROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS Preinclude file Defined symbol ERNAL_FLAS T_HEADER T_HEADER	Ude airectones: (o \$//././CMS \$//./ \$//./ \$//././devin \$//././devin \$//././ als: (one per line) SH=1 ENABLE=0 DCD ENABLE=0	ils/include ces/MIMXRT1	05 <mark>2/drivers</mark> processor o Preserve co Generate #	s output to omments #ine dire	file sctives
Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET	Additional Inci SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS SPROJ_DIRS Preinclude file Defined symbol ERNAL_FLAS T_HEADER_ T_HEADER_	Ude airectones: (o \$//././CMS \$//./. \$//./. \$//././devin \$/ \$//././devin \$/.	ils/include ces/MIMXRT1	05 <mark>2/drivers</mark> processor o Preserve co Generate #	s output to omments #line dire	file sctives

#### Figure 39. hello\_world的已定义符号

## 步骤 2:

如下图所示,将默认条目更改为Reset\_Handler。



Figure 40. 将默认项更改为Reset\_Handler

## 注意

如果设置了此步骤,则可以跳过步骤7。

## 步骤 3:

找到链接文件MIMXRT1052xxxxx flexspi nor.icf并将数据区域从TCM更改为SDRAM。

define symbol m_interrupts_start	$= 0 \times 60002000;$
define symbol m_interrupts_end	= 0x600023FF;
define symbol m_text_start	= 0x60002400;
define symbol m_text_end	= 0x63FFFFFF;
define symbol m_data_start	= 0x80000000;
define symbol m_data_end	= 0x8001FFFF;
define symbol m_data2_start	= 0x80200000;
define symbol m_data2_end	= 0x8023FFFF;

#### Figure 41. 将数据区域从TCM更改为SDRAM

## 步骤 4:

构建项目并生成图像。 生成项目并生成图像。 您可以在以下位置找到 hello\_world.srec:

I =   flexspi_nor_debug			33 <del></del> 8	
	o_world > iar > flexspi_nor_debu	g	ע Se	arch fle
Name	Date modified	Туре	Size	
📕 list	2018/7/5 13:47	File folder		
🧯 obj	2018/7/5 13:47	File folder		
hello_world.out	2018/7/5 13:47	PSpice Simulation	201 KB	
hello_world.srec	2018/7/5 13:47	SREC File	27 KB	
u IS	re 42 hollo world			

## 步骤 5:

将 hello\_world.srec 复制到elftosb文件夹

→ → ↑ 📜 < Flashloader_i.MXF	T1050_GA > Tools > elftosb > win		<ul><li>V O Search win /</li></ul>
^ Name	Date modified	Туре	Size
elftosb.exe	2018/1/15 18:19	Application	807 KB
hello_world.srec	2018/7/5 10:58	SREC File	27 KB
•			
•			
~			

Figure 43. 复制 hello\_world.srec

## 步骤 6:

复制 imx-flexspinor-normal-unsigned.bd 并将其重命名为 imx-flexspinor-normal-unsigned-dcd.bd。

ganize 🔻 🛛 🖉 Open 🔻	Share with 👻 Burn New folder		855 • E	
🎉 Git 🔷	Name	Date modified	Туре	
QR	enable_hab.bd	2018/2/11 9:48	BD File	
J RT1020	imx-dtcm-encrypted.bd	2018/2/11 10:14	BD File	
A RT1050	imx-dtcm-signed.bd	2018/2/11 10:14	BD File	
Flashloader_i.MXRT	imx-dtcm-unsigned.bd	2018/2/11 10:14	BD File	
doc 🛁	imx-flexspinor-normal-signed.bd	2018/2/11 10:14	BD File	
example_images	imx-flexspinor-normal-unsigned.bd	2018/2/11 10:15	BD File	
Flashloader	imx-flexspinor-normal-unsigned-dcd.bd	2018/4/18 16:35	BD File	
J. I cols	imx-tlexspinor-plugin-signed.pd	2018/2/11 10:15	BD File	
bd_file	imx-flexspinor-plugin-unsigned.bd	2018/2/11 10:16	BD File	
Imx10xx	imx-itcm-encrypted.bd	2018/2/11 10:16	BD File	
pinost -	•			

查找文件副本并将它重命名

打开 imx-flexspinor-normal-unsigned-dcd.bd 并添加 DCD 路径。



Figure 45. 添加 DCD 路径

#### 步骤 7:

打开路径*Flashloader\_i.MXRT1050\_GA*\*Tools*\*bd\_file*\*imx10xx*下的 *imx-flexspinor-normal-unsigned-dcd.bd*。打开它并将entryPointAddress设置为0x60002000,如下图所示。



Figure 46. 将 entryPointAddress设置为 0x60002000

#### 注意

如果设置了此步骤,则可以跳过步骤2。

#### 步骤 8:

将 dcd.bin 复制到以下路径:

Home Share View		-	
✓ ↑	1.1 > Tools > elftosb > win		✓ U Search w
Name	Date modified	Туре	Size
🗹 dcd.bin	2018/6/7 14:31	BIN File	1 KB
elftosb.exe	2018/1/15 18:19	Application	807 KB

Figure 47. 将 dcd.bin 复制到以下路径

## 步骤 9:

现在,我们可以使用elftosb文件生成i.MX可引导映像。打开cmd.exe并键入以下命令:

*elftosb.exe -f imx -V -c ../../bd\_file/imx10xx/imx-flexspinor-normal-unsigned-dcd.bd -o ivt\_flexspi\_nor\_hello\_world.bin hello\_world.srec* 



Figure 48. 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的内存区域填充字节(全部为0x00)。

ivt\_flexspi\_nor\_hello\_world\_nopadding.bin:

直接从ivtdata开始,无需在ivt之前进行任何填充。

后面的部分将用于为Hyper FLASH编程生成SB文件。

步骤 10:

这一步,我们将创建用于Hyper Flash编程的SB文件。将生成一个boot\_image.sb文件,供 MfgTool稍后使用。打开cmd.exe并键入以下命令:

elftosb.exe -f kinetis -V -c ../../bd\_file/imx10xx/program\_flexspinor\_image\_HyperFlash.bd -o boot\_image.sb ivt\_flexspi\_nor\_hello\_world\_nopadding.bin

E Windows PowerShell			$\times$
PS C:\Users\	50_GA nsigne 50_GA ile/in _worlo	\Flash] ed-dcd. \Flash] nx10xx/ d_nopag	oa bd oa pr ldi
ng.bin Boot Section 0x00000000: FILL   adr=0x00002000   len=0x00000004   ptn=0xc0233007 ENA   adr=0x00002000   cnt=0x00000004   flg=0x0900 ERAS   adr=0x60000000   len=0x00000004   ptn=0xf0000000f ENA   adr=0x00003000   cnt=0x00000004   flg=0x0900 LOAD   adr=0x60001000   len=0x000032b4   crc=0xc5dd3b3d   flg=0x0000 PS C:\Users\. <sup>-</sup> \Desktop\Flashloader_i.MXRT1050_GA\Flashloader_i.MXRT10 der_RT1050_1.1\Tools\elftosb\win>	50_GA\	\F1ash]	.oa ~

Figure 49. 创建用于Hyper Flash编程的SB文件

执行上述命令后,将在elftosb文件夹下生成 boot\_image.sb。

Vame	Date modified	Туре	Size
boot_image.sb	2018/7/9 9:26	SB File	13 KB
dcd.bin	2018/6/7 14:31	BIN File	1 KB
🗉 elftosb.exe	2018/1/15 18:19	Application	807 KB
hello_world.srec	2018/7/5 14:53	SREC File	27 KB
ivt_flexspi_nor_hello_world.bin	2018/7/9 9:26	BIN File	17 KB
🍸 ivt_flexspi_nor_hello_world_nopadding.bin	2018/7/9 9:26	BIN File	13 KB

Figure 50. 生成 boot\_image.sb

## 步骤 11:

	ols-rel > Profiles > MXRT105X > OS	Firmware	Search OS
A			· O Scarch OS /
Name	Date modified	Туре	Size
boot_image.sb	2018/7/5 14:05	SB File	13 KB
🗾 ivt_bootdata.bin	2017/12/6 18:10	BIN File	1 KB
ivt_flashloader.bin	2018/2/11 8:52	BIN File	88 KB
ucl2.xml	2018/2/11 8:55	XML Document	8 KB
1 million			
1			
A			
6			
<b>6</b> 3			

将boot\_image.sb文件复制到OS固件文件夹:

Figure 51. 将 boot\_image.sb 复制到OS固件文件夹

## 现在

确保<mfgtool\_root\_dir>文件夹下cfg.ini文件中"[List]"下的"name"改为"MXRT105x-DevBoot"归档。



通过将SW7设置为"1-关、2-关、3-关、4-开",将EVK板切换到串行下载模式。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53所示:

MfgTool_MultiPanel (Library: 2.7.0)	() <mark>-</mark> 53 <u></u> 55	
Hub 1Port 1	Status Information	
Drive(s):	Successful	0
	Failed	0
HID-compliant vendor-defined device	Failure Rate:	0 %
	Start	Exit
	Start	Exit

Figure 53. 连接了设备的MfgTool GUI

单击"开始",Mfgtool进程启动。完成后,MfgTool将显示成功状态,如图54所示。单击"停止"并关闭Mfgtool。

MfgTool_MultiPanel (Library: 2.7.0)	1 <u>111</u>	×
Hub 1Port 1	Status Information	
Drive(s):	Successful	1
	Failed	0
HID-compliant vendor-defined device	Failure Rate:	0.00 %
	Start	Exit

Figure 54. MfgTool 成功状态

## 步骤 12:

将RT1050-EVK板切换到内部引导模式,通过将SW7设置为"1-ON, 2-OFF, 3-ON, 4-OFF"选择 SD卡作为引导设备。将USB线连接到J28并打开一个终端,然后重置开发板。"hello world"将被 打印到终端上。

#### Conclusion

Serial Port Utility	
<u>File Edit View Tools H</u> elp	
👝 🔛 🔚 🚾 🕂 —	
Serial Port Setting	hello world.
Port mbed S(COM99) 🔻	
Baudrate 115200 💌	
Data Bits 8	
Parity None 🔻	
Stop Bits 1	
Flow Type None	
Receive Setting	
Text  Hex	
Auto Feed Line	
Display Send	
Display Time	
Send Setting	Send
Text     Hex	
🔲 Loop 1000 🍦 ms	家迎
COM99 OPENED, 115200, 8, NONE,	1, OFF Rx: 14 Bytes Tx: 0 Bytes

Figure 55. "hello world" 被打印到终端上

## 4. 八进制SPI Flash支持列表

除板载EVK Hyper Flash外,还支持以下Flash:

Table 9. 八进制 SPI Flash 支持列表

供应商	Flash
ISSI (Hyper Flash)	IS26KS256
SPANSION (Hyper Flash)	KS512SBPHI02
Macronix	MX25UM513
Micron	MT35X
Adesto	ATXP032/ ATXP128
GigaDevice	GD25LX256E

## 5.结论

本应用说明主要介绍如何逐步使用Flashloader。更多信息,您可以参考<u>i.MX MCU 制造用户指</u><u>南</u>。

**Revision history** 

## 6.修订历史

版本序号	日期	Substantive changes
0	12/2017	初始发行
1	06/2018	应用手册的名称更改为: • 如何启用八进制SPI闪存和SD卡启 动 • 更新文档以适应SDK版本2.3.1 • 更新文档以适应Flashloader 1.1版 • 表9的标题已更改为 表9.八进制SPI闪存支持列表
2	07/2018	<ul> <li>增加了更改输入地址的步骤。</li> <li>使用.srec文件而不是.out文件作为源 文件。</li> </ul>
3	09/2018	更新了表9.八进制SPI Flash支持 列表。
4	09/2018	更新了表9.八进制SPI Flash支持列表中的Adesto详细信息。
5	07/2019	更新了表9.八进制SPI Flash支持 列表。

### Table 10. 修订历史

How to Reach Us:

Home Page: nxp.com

Web Support: nxp.com/support Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorlQ, QorlQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, UMEMS, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamlQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) 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. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

> Document Number: AN12107 Rev. 5 07/2019



# arm