# AN13116 如何在单核 RT1170 上进入 STBY 模式?

Rev. 0 — February 2, 2021

Application Note

# 1 简介

i.MX RT1170 是一款突破性的跨界处理器,它把最高运行频率提高到了 1 GHz, 而且还在结合了高性能运算和多媒体功能的同时,增强了可用性及实时功能。双 核 i.MX RT1170 分别 以 1 GHz,400 MHz 的频率在 Arm<sup>®</sup> Cortex<sup>®</sup>-M7 和 Arm Cortex-M4 上运行,同时还兼具良好的安全性。这款处理器可以在不同的温度下 使用,以及在不同的领域,尤其在工业及汽车行业

RT1170 系列可以分为两个部分:单核和双核。关于双核及其设置,请参考 《RT1170 时钟和低功耗特性》(文档 AN13104)。本文档会介绍如何在 STBY 模式下使用单核 RT1170 以及如何模拟 RT1170 的单核运行状态。双核和单核 的操作差别非常大,所以在运行的时候要注意。

本文档内所使用的硬件是 RT1170 EVK RevC1 (本文档中简称为 EVK) , 软件 是基于 IAR IDE 的 SDK 2.9.0。SDK 为单核和双核提供了不同的 demo, 同时也支持 flash 和 RAM。

#### 2 概述

如 图 1 所示,进入 STBY 的基本原则是使两个 CPU 进入低功耗模式并且发送 STBY 需求。CPU 低功耗状态有 WAIT,STOP以 及 SUSPEND 模式。除此之外,这里要求的低功耗模式 CPU 可以为除了 RUN 之外的任意模式且两个 CPU 可以处于不同的状态,比如:

- CM7 WAIT, CM4 STOP
- · CM7 STOP, CM4 STOP
- · CM7 SUSPNED, CM4 WAIT
- CM7 SUSPEND, CM4 SUSPEND

第二条原则是两个 CPU 已经发送了 STBY 请求。如果任何一个 CPU 没有发送,那么系统就不能进入 STBY 模式。



对于单核的芯片,CM4 不能发送这个请求。所以需要 CM7 发送一些命令来强制系统进入 STBY 模式。接着 CM7 发送进入低功耗 模式然后发送一个 stby\_request 信号。低功耗模式可以是 WAIT,STOP 或者 SUSPEND。示意图如下:



目录	
1	简介1
2	概述1
3	如何在单核情况下进入 STBY 模式?
	2
4	如何模拟 RT1170 EVK 的单核状态?
	2
5	单核芯片与在 RT1176 上模拟单核的
	不同3
6	如何在 SDK 里创建一个单核项目3
6.1	用 IAR 运行 demo3
6.2	用 Keil <sup>®</sup> MDK/µVision 运行程序8
6.3	在 Arm <sup>®</sup> GCC 上运行 demo 12



## 3 如何在单核情况下进入 STBY 模式?

进入 STBY 需要双核都发送 stby\_request,但是单核芯片(比如 RT1172,RT1176)没有 CM4 所以无法发送。CM4 不能做到。 因此,需要用一个名为 Force\_STBY 的函数,该函数可以强制 CM4 请求 STBY,然后当 M7 进入低功耗模式和发送 stby\_request 的时候,整个芯片就进入了 STBY 模式。

GPC\_STBY\_CTRL->STBY\_MISC |= GPC\_STBY\_CTRL\_STBY\_MISC\_FORCE\_CPU1\_STBY\_MASK;

## 4 如何模拟 RT1170 EVK 的单核状态?

通常 RT1170EVK 都是双核芯片。用户可以通过一些设置来模拟单核状态。需要注意的是这只是一个初期的评估。基本的方法是 CM4 进入 Suspend 模式然后发送 STBY 请求。

- 1. 系统启动后,CM7 会配置如下的寄存器:
  - a. Write PGMC CM4 CPC register as CPU mode with power-off at SUSPEND.
  - b. Write GPC1 IRQ mask registers as 0xFFFFFFF so that no interrupt can wake it up.
  - c. Write GPC1 NON IRQ mask register as 0x3 to avoid a pending interrupt from debugger stop entering low power mode.
  - d. Write GPC1 control register to request SUSPEND mode and SBTY mode.
- 2. CM7 通过 SRC\_SCR 寄存器释放出 CM4, 现在是双核模式。
- 3. CM4 运行一个指令:断言 WFI 去触发 GPC1 低功耗时序来让自己进入 SUSPEND 模式以及发送 STBY 请求。
- 4. 当 CM4 进入 SUSPEND 模式时,CM7 会写入一些寄存器来通过软件锁住 CM4:
  - a. Write GPC1 AUTHEN register to lock CM1 register access.
  - b. Write CM4 CCGR slice in CCM to gate off CM4 clock.
  - c. Write CM4 CCGR AUTHEN register 'white\_list' and 'lock' fields meaning no CPU can access CM4 CCGR register again. Thus CM7 software can't turn on CM4 clock again.
  - d. Write PGMC CM4 CPC AUTHEN register to lock any access so that software can't turn on CM4 power again.

本文档会提供详细的代码来提供参考。基本的方法是让 CM4 进入 Suspend 模式然后发送 STBY 请求,在此之上,屏蔽所有在 GPC CM1 中的唤醒源然后并且锁住这些寄存器的权限:任何 CPU 都不能进入和修改。最后,CM4 进入了 Suspend 模式,没有唤醒源 可以唤醒它,也没有 CPU 可以修改这些设定。

下面的 API 可以完成上述的配置.

void Powerdown\_CM4(void);

关于详细信息,请参考 AN13116SW。

# 5 单核芯片与在 RT1176 上模拟单核的不同

单核芯片与模拟单核状态是有一些差别的。首先,单核芯片不能用 RDC 去分配一个外设到一个域。但是模拟单核状态下就可以使用 RDC。对于单核芯片,所有的外设都被分配给 M7 域,这表示在握手过程中,只有一个暂停请求(stop\_req)来自于 M7,所以只 需要检查来自 M7 的 stop\_ack。如果想了解关于握手的信息,请参考《RT1170 时钟和低功耗特性》(文档 AN13104)的 4.5 节。

#### 表 1. 单核芯片与在 RT1176 上模拟单核的不同一

	单核芯片	在 RT1176 上模拟单核
STBY Method	Force_STBY	CM4 进入 Suspend STBY
RDC	不支持	支持
Handshake	只有 CM7 需要发送 stop_req 以及检查 stop_ack	参考《RT1170 时钟和低功耗特性》(文档 AN13104)的 4.5 节。

## 6 如何在 SDK 里创建一个单核项目.

MIMXRT1170 EVK 支持单核程序的运行。本章节将会展示如何用不同的 IDEs(IAR,Keil 和 Armgcc)来设置和运行单核的 demo。

#### 6.1 用 IAR 运行 demo

下面将展示如何运行 SDK 中的 power\_mode\_switch demo:

- 1. 目标 demo 的地址为 *<install\_dir>lboards\evkmimxrt1170\demo\_apps\power\_mode\_switch\bm\core0\iar\* power\_mode\_switch\_bm\_core0.eww
- 2. 将项目导入 IAR IDE。

Vorkspace 🔻 🕈 🗙	IAR Information Center for Arm power_mode_switch_core0.c x startup_MIMXRT1176_cm7.s
single_core ~	
Files       Image: Component status         Image: Component status       Image: Component status	<pre>/*  * Copyright 2020 NXP  * All rights reserved.  *  * SPDX-License-Identifier: BSD-3-Clause  */  #include "fsl_common.h"  #include "fsl_debug_console.h"  #include "fsl_debug_console.h"  #include "lpm.h"  #include "pin_mux.h"  #include "clock_config.h"  #include "clock_config.h"  #include "fsl_mu.h"  #include "fsl_mu.h"  #include "fsl_mu.h"  #include "fsl_pgmc.h"  #include "fsl_soc_src.h"  /************************************</pre>

3. 默认的建立目标应该是 debug, 如下图所示:

Workspace	▼ ‡ ×
debug	~
debug	
riexspi_nor_debug release	
flexspi_nor_release	
- 🕀 🛋 component	
- te doc	
🖃 🖬 drivers	
h chip_init_def.h	
t le lpm.c	
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	
Setnoint table def h	
─────────────────────────	
_	

4. 在任务栏的 Project 选择 Edit Configurations:



5. 添加一个新的目标并命名为 single\_core,然后点击 OK:

Configurations for "power_mode_switch_bm_c × Configurations: UK Respi_nor_debug release Remove New Configuration × Name: UK Single_core! Cance! Tool chain: Arm Based on configuration: debug Factory settings Debug Release Configurations for "power_mode_switch_bm_c × Configurations for "power_mode_switch_bm_c × Remove
Configurations: OK   flexspi_nor_debug New   flexspi_nor_release Remove   New Configuration ×   Name: OK   single_core Cancel   Tool chain: Cancel   Tool chain: Cancel   Tool chain: Cancel   Tool chain: Factory settings   Debug Factory settings   Debug Release   Configurations for "power_mode_switch_bm_c   K   debug New   flexspi_nor_debug   release   indexpi_nor_debug   release   single_core   Remove
Idebug       New         Idessig_nor_debug       Remove         Idessig_nor_release       Remove         New Configuration       X         Name:       OK         single_core       Cancel         Tool chain:       Cancel         Tool chain:       Cancel         Based on configuration:       Edbug         Image:       Debug         Image:       Debug         Image:       Debug         Image:       Debug         Image:       Image:         Image:
Image: Processe       Remove         Remove       New Configuration         Name:       OK         single_core       Cancel         Tool chain:       Arm         Arm       Based on configuration:         Based on configuration:       Pactory settings         Obebug       Release         Release       OK         Image: Release       Release         Image: Release       Remove         Release       Remove         Remove       Remove
New Configuration   Name:   Single_corel   Tool chain:   Arm   Based on configuration:   debug   Factory settings   O Debug   Release   Configurations for "power_mode_switch_bm_c × Configurations:    Configurations:   debug   flexspi_nor_debug   flexspi_nor_debug   flexspi_nor_release   single_core
Name: OK   single_corel Cancel   Tool chain: Arm   Arm Based on configuration:   debug Image: Second Seco
Configurations for "power_mode_switch_bm_c × Configurations: Configurations: Configurations for "power_mode_switch_bm_c × Configurations: Configurat
Arm       Image: Second and the second an
Based on configuration: debug Factory settings Debug Release Configurations for "power_mode_switch_bm_c × Configurations: debug flexspi_nor_debug release flexspi_nor_release single_core Remove
Gebug   Factory settings   Debug   Release   Configurations for "power_mode_switch_bm_c × Configurations:    OK   Gebug   flexspi_nor_debug   release   flexspi_nor_release   single_core   Remove
Octory settings     O Debug     O Release     Configurations for "power_mode_switch_bm_c ×      Configurations:     OK     debug     flexspi_nor_debug     release     flexspi_nor_release     single_core     Remove
Configurations for "power_mode_switch_bm_c × Configurations: Configurations
Configurations for "power_mode_switch_bm_c × Configurations: OK debug flexspi_nor_debug release flexspi_nor_release single_core Remove
Configurations: OK debug flexspi_nor_debug release flexspi_nor_release single_core Remove
debug flexspi_nor_debug release flexspi_nor_release single_core
release flexspi_nor_release Remove
single_core
(Drag to order)
图 6. 添加一个新的目标

6. 进入项目的 Options 然后做以下设置:

board      CMSIS	Options	nc.
E component	Make	
E device	Compile	
🗉 🖬 doc	Rebuild All	L
🗉 🖬 drivers	Clean	-
🖓 🖬 source		h.
E chip_init_def.h	C-STAT Static Analysis	
—⊕ le lpm.c	Stop Build	ar
- ⊡ i i power_mode_sw	Add	-
🗉 🖬 startup	Remove	-
E dutilities	Rename	
🗉 🖬 Output	Version Control System	
	Open Containing Folder	15 K 1
	File Properties	4P
	Set as Active	V.P.

#### a. 添加宏 SINGLE\_CORE\_M7 到 Preprocessor of C/C++ Compiler。

alogoly.				Facto	ory Settings
General Options	📃 🗌 Multi-file Comp	pilation			
Static Analysis	🗌 Discard U	nused Publics			
Runtime Checking					
C/C++ Compiler	MISRA-C:	1998	Encodings	Extra C	Options
Assembler	Language 1	Language 2	Code	Optimizations	Output
Output Converter	List	Preprocessor	Diagnos	stics MISR.	A-C:2004
Custom Build	1				
Build Actions	Ignore stand	dard include direct	ories		
Linker	Additional inclu	ide directories: (on	e per line)		
Debugger	\$PROJ DIR\$		- p-:		•
Simulator	\$PROJ_DIR\$	/			
CADI	\$PROJ_DIR\$	//////devi	ces/MIMXRT1	176/drivers	
CMSIS DAP	\$PROJ_DIR\$	//////devi	ces/MIMXRT1	176	
GDB Server	\$PROJ_DIR\$	//////CM	SIS/Include		*
I-jet	Preinclude file:				
J-Link/J-Trace					
TI Stellaris					
Nu-Link	Defined symbol	ols: (one per line)			
PE micro	DEBUG			processor output to	file
ST-LINK	SINGLE_COP	RE_M7	F	Preserve comments	S
Third-Party Driver		TTYDE HART-		Generate #line dire	ctives
TI MSP-FET	SERIAL_FOR				
TI XDS					

b. 删除所有在 Linker -> Input 里的信息,结果应该如下图所示:

Category:	Factory Settings
General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Output Converter Custom Build Build Actions Linker Debugger Simulator CADI CMSIS DAP GD8 Server I-jet Julie/L/Targe	#define Diagnostics Checksum Encodings Extra Options Config Library Input Optimizations Advanced Output List Keep symbols: (one per line)
TI Stellaris	Raw binary image
PE micro	File: Symbol: Section: Align:
S I-LINK Third-Party Driver TI MSP-FET TI XDS	File: Symbol: Section: Align.
	OK Cancel

c. 在 Debugger 部分,选择 Multicore 里的 Disabled:

Category: General Options Static Analysis	Factory Settings
Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Neu Ink	Setup       Download       Images       Multicore       Extra Options       Plugins         Symmetric multicore       Number of cores:       1       1       1         Asymmetric multicore       Images       Images       1       1         Asymmetric multicore       Images       Images       1       1         Object       Disabled       Images       Images       1         Partner workspace:       SPROJ_DIR\$/./j./core1/iar/power_mode_:       Images       1         Partner project:       power_mode_switch_bm_core1       Images       1         Partner configuration:       debug       Images       Partner cores       1
PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	O Advanced Session configuration:

d. 设置完成之后,调试及运行程序,在串口将会显示以下内容:

```
🖉 COM7 - PuTTY
                                                               \times
This is single core M7.
CPU wakeup source 0x0...
         *****
      Power Mode Switch Demo for iMXRT1176
System previous setpoint is O
System current setpoint is 0
M7 previous CPU mode is RUN
M7 current CPU mode is RUN
M7 CLK is 696 MHz
Please select the desired operation:
Press A to demonstrate typical set point transition.
Press B to demonstrate cpu mode switch in setpoint 0.
Waiting for select...
图 11. 程序输出
```

- 6.2 用 Keil<sup>®</sup> MDK/µVision 运行程序
  - 1. 用 keil 打开 power\_mode\_switch 程序。程序的地址在 <install\_dir>lboards\evkmimxrt1170\demo\_apps\power\_mode\_switch\bm\core0\mdk\power\_mode\_switch\_bm\_core0.uv mpw



2. 把目标改为 flexsip\_nor\_debug:

📓 🗼 🗕 🔛 🔤	power_mode_switch_bm 🔽 🐔 🖶 🖶 💠 🐡 🎒	
	power_mode_switch_bm_core0 debug	
	power mode switch bm core0 flexspi nor debug	
kSpace	power_mode_switch_bm_core0 release	
Project: nower mode	power_mode_switch_bm_core0 flexspi_nor_release	
power_mode_swite source board doc	ch_bm_core0 3 * All rights reserved. 4 * 5 * SPDX-License-Identifier: E 6 */ 7 2	
图 13. 修改目标		

3. 打开目标的 Options:

<u> </u>	
power_mode_switch_b	
₽ 🛛	Options for Target 'power_mode_switch_bm_core0 flexspi_nor_debug'         X
switch_bm_c	Device Target Output Listing User C/C++ (AC6) Asm Linker Debug Utilities
	NXP MIMXRT1176DVMAA:cm7 Code Generation ARM Compiler: Use default compiler version 6
	<u>X</u> tal (MHz): 12.0
	Operating system: None
	System Viewer File: 🔽 Use MicroLIB 🗖 Big Endian
	Floating Point Hardware: Double Precision
	Use Custom File
art	Read/Only Memory Areas
erial_manager	default off-chip Start Size Startup default off-chip Start Size Nolnit
,τs	□ ROM1: 0x30000000 0x1000000
	□ ROM2: □ C □ RAM2: 0x20240000 0x80000 □
<u> </u>	□ ROM3: □ □ RAM3: 0x202C0000 0x80000 □ ×*******
🛛 🔶 Templa   <	on-chip on-chip
	□ IROM1: C
50 RO-data=7378	□ IROM2: □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
er mode switch )	
0:01:04	OK Cancel Defaults Help
3//bower_mode_s/	
图 14 Open opti	ons

4. 在 C/C++ 里, 添加 SINGLE\_CORE\_M7 到 Define :

Options for Target 'power_mode_switch_bm_core0 flexspi_nor_debug'	$\times$				
Device Target Output Listing User C/C++ (AC6) Asm Linker Debug Utilities					
Preprocessor Symbols					
Undefine:					
Language / Code Generation         Execute-only Code       Warnings:         AC5-like Warnings       Language C:         Cotimization:       C1					
University       University       University       Language C++:       C++11         Link-Time Optimization       Image: Plain Char is Signed       Image: Short enums/wchar					
Split Load and Store Multiple Read-Only Position Independent use RTTI					
✓ One ELF Section per Function ☐ Read-Write Position Independent ☐ No Auto Includes					
Include Paths Misc Controls Include atherican and a sections of the standing for builtin anthumb					
Compiler control string					
OK Cancel Defaults Help					
图 15. 添加宏到 define					

5. 打开 fsl\_incbin.S 的 Options :



6. 去掉选项 Include in Target Build:

Properties Aam Properties Aam Properties Aam Properties Aam Path: ///////devices/MIMXRT1176/Adilities/ncbin/fsl_mcbin.5 File Type: Assembly language file Size: 899 Bytes Size: 899 Bytes Size: Sun Jan 10 09:45:18 2021 Stop on Exit Code: Not specified Stop on Exit Code: Not specified	
Path: ////////////////////////////////////	
Path: V/////devces/MIMXRT1175/utilities/incbin/fs/uncbin	
File Type:     Assembly language file     Include in Target Build       Size:     899 Bytes     If Always Build       last change:     Sun Jan 10 09:45:18 2021     If Generate Assembler SRC File       Stop on Exit Code:     Not specified     If Lange Build Commission	
Size: 1839 Bytes Provide Size: 1839 Bytes Prov	
Itatichange: Sun Jan 10 09/45/18 2021	
Stop on Exit Code: Not sneefied	
Interpreter Compression	
Custom Arguments:	
Memory Assignment:	
Code / Const: <a> </a> <a> </a>	
Zero Initialized Data: <pre><default></default></pre>	
Other Data:   <default></default>	
Layer: <pre> dnot assigned&gt; </pre>	
OK Cancel Defaults He	-lp
Layer: <a href="https://www.cancel.com">www.cancel.com</a> U Cancel Defaults He	slp

7. Debug 然后运行程序。

Putty	—	×
This is single core M7.		
CPU wakeup source 0x0		
**************************************		
System previous setpoint is 0 System current setpoint is 0 M7 previous CPU mode is RUN M7 current CPU mode is RUN M7 CLK is 696 MHz		
Please select the desired operation: Press A to demonstrate typical set point transition. Press B to demonstrate cpu mode switch in setpoint 0.		
Waiting for select		
图 18. 程序输出		

6.3 在 Arm<sup>®</sup> GCC 上运行 demo

本节展示了如何用 GCC IDE 运行 demo。



1. 在项目的文件夹里打开 cmakelist.txt:

Name	Date modified	Туре	Size
output.map	2021/1/21 14:17	MAP File	259 KB
📄 build_log.txt	2021/1/21 14:17	Text Document	0 КВ
🗋 flags.cmake	2021/1/21 14:11	CMAKE File	8 KB
CMakeLists.txt	2021/1/21 14:05	Text Document	5 KB
🐁 build_all.bat	2021/1/10 9:44	Windows Batch File	2 KB
🔊 build_debug.bat	2021/1/10 9:44	Windows Batch File	1 KB
🖲 build_flexspi_nor_debug.bat	2021/1/10 9:44	Windows Batch File	1 KB
build_flexspi_nor_release.bat	2021/1/10 9:44	Windows Batch File	1 KB
🐁 build_release.bat	2021/1/10 9:44	Windows Batch File	1 KB
🐁 clean.bat	2021/1/10 9:44	Windows Batch File	1 KB
🗋 config.cmake	2021/1/10 9:44	CMAKE File	1 KB
MIMXRT1176xxxxx_cm7_flexspi_nor_ramfunc.ld	2021/1/10 9:44	LD File	9 KB
MIMXRT1176xxxxx_cm7_ram.ld	2021/1/10 9:44	LD File	7 KB
build_all.sh	2021/1/10 9:44	SH File	2 KB
build_debug.sh	2021/1/10 9:44	SH File	1 KB
build_flexspi_nor_debug.sh	2021/1/10 9:44	SH File	1 KB
build_flexspi_nor_release.sh	2021/1/10 9:44	SH File	1 KB
build_release.sh	2021/1/10 9:44	SH File	1 KB
📄 clean.sh	2021/1/10 9:44	SH File	1 KB

- 2. 有三部分的程序需要被删除或者注释掉:
  - a. 包含 core1 的代码。



b. Incbin 的路径。

```
62
     set (CMAKE MODULE PATH
63
          ${ProjDirPath}/../../../../devices/MIMXRT1176/drivers
64
          #${ProjDirPath}/../../../../../devices/MIMXRT1176/utilities/incbin
65
          ${ProjDirPath}/../../../../devices/MIMXRT1176
          ${ProjDirPath}/../../../../CMSIS/Include
66
67
          ${ProjDirPath}/../../../../devices/MIMXRT1176/utilities
68
          ${ProjDirPath}/../../../../components/uart
                the market of the second se
                                                         . . . . . .
图 21. 删除或注释 64 行
```

c. include incbin。

87	<pre>include(driver_soc_src_MiMAKili/0_cm/)</pre>
88	
89	<pre>include(driver_pmu_1_MIMXRT1176_cm7)</pre>
90	
91	<pre>#include(utility_incbin_MIMXRT1176_cm7)</pre>
92	
93	<pre>include(driver_clock_MIMXRT1176_cm7)</pre>
94	
95	<pre>include(driver_common_MIMXRT1176_cm7)</pre>
96	
97	<pre>include(device_MIMXRT1176_CMSIS_MIMXRT1176_cm7)</pre>
图 22 则除式计容 04	
图 22. 厕际乳注稗 91	13

3. 打开 flags.cmake 文件:

Name	Date modified	Туре	Size
📄 output.map	2021/1/21 14:17	MAP File	259 KB
build_log.txt	2021/1/21 14:17	Text Document	0 KB
flags.cmake	2021/1/21 14:11	CMAKE File	8 KB
CMakeLists.txt	2021/1/21 14:05	Text Document	5 KB
Size: 7.25 KB	2021/1/10 9:44	Windows Batch File	2 KB
S build_debug.b Date modified: 2021/1/21 14:11	2021/1/10 9:44	Windows Batch File	1 KB
🔊 build_flexspi_nor_debug.bat	2021/1/10 9:44	Windows Batch File	1 KB
louild_flexspi_nor_release.bat	2021/1/10 9:44	Windows Batch File	1 KB
🖲 build_release.bat	2021/1/10 9:44	Windows Batch File	1 KB
🐁 clean.bat	2021/1/10 9:44	Windows Batch File	1 KB
config.cmake	2021/1/10 9:44	CMAKE File	1 KB
MIMXRT1176xxxxx_cm7_flexspi_nor_ramfunc.ld	2021/1/10 9:44	LD File	9 KB
MIMXRT1176xxxxx_cm7_ram.ld	2021/1/10 9:44	LD File	7 KB
build_all.sh	2021/1/10 9:44	SH File	2 KB
build_debug.sh	2021/1/10 9:44	SH File	1 KB
build_flexspi_nor_debug.sh	2021/1/10 9:44	SH File	1 KB
build_flexspi_nor_release.sh	2021/1/10 9:44	SH File	1 KB
build_release.sh	2021/1/10 9:44	SH File	1 KB
📄 clean.sh	2021/1/10 9:44	SH File	1 KB
Clean.sn	2021/1/10 9:44	2H File	1 KB

#### 图 23. 打开 flags.cmake

4. 添加宏 SINGLE\_CORE\_M7 到目标:

67	-fdata-sections \
68	-ffreestanding \
69	-fno-builtin \
70	-mapcs \
71	-std=gnu99 \
72	")
73	SET (CMAKE_C_FLAGS_DEBUG " \
74	-DSINGLE CORE M7 \
75	-DDEBUG \
76	-DCPU_MIMXRT1176DVMAA_cm7 \
77	-DSERIAL_PORT_TYPE_UART=1 \
78	-g \
79	-00 \
80	-mcpu=cortex-m7 \
81	-Wall \
82	-mfloat-abi=hard \
83	-mfpu=fpv5-d16 \
84	-mthumb \
85	-MMD \
86	-MP \
87	-fno-common \
88	-ffunction-sections \
89	-fdata-sections \
90	-ffreestanding \
91	-fno-builtin \
92	-mapcs \
93	-std=gnu99 \
94	")

5. 修改并保存之后,双击 build\_debug.bat 来建立目标。

6. 如果成功的话,有两个新的文件夹(CmakeFiles, debug) 被建立。

		21	
] output.map	2021/1/21 14:47	MAP File	259 KB
build_log.txt	2021/1/21 14:47	Text Document	0 КВ
] cmake_install.cmake	2021/1/21 14:47	CMAKE File	2 KB
CMakeCache.txt	2021/1/21 14:47	Text Document	19 KB
] Makefile	2021/1/21 14:47	File	71 KB
] flags.cmake	2021/1/21 14:11	CMAKE File	8 KB
CMakeLists.txt	2021/1/21 14:05	Text Document	5 KB
Juild_all.bat	2021/1/10 9:44	Windows Batch File	2 KB
🔊 build_debug.bat	2021/1/10 9:44	Windows Batch File	1 KB
build_flexspi_nor_debug.bat	2021/1/10 9:44	Windows Batch File	1 KB
build_flexspi_nor_release.bat	2021/1/10 9:44	Windows Batch File	1 KB
build_release.bat	2021/1/10 9:44	Windows Batch File	1 KB
lean.bat	2021/1/10 9:44	Windows Batch File	1 KB
] config.cmake	2021/1/10 9:44	CMAKE File	1 KB
MIMXRT1176xxxxx_cm7_flexspi_nor_ramfunc.ld	2021/1/10 9:44	LD File	9 KB
MIMXRT1176xxxxx_cm7_ram.ld	2021/1/10 9:44	LD File	7 KB
] build_all.sh	2021/1/10 9:44	SH File	2 KB
build_debug.sh	2021/1/10 9:44	SH File	1 KB
] build_flexspi_nor_debug.sh	2021/1/10 9:44	SH File	1 KB
] build_flexspi_nor_release.sh	2021/1/10 9:44	SH File	1 KB
] build_release.sh	2021/1/10 9:44	SH File	1 KB
] clean.sh	2021/1/10 9:44	SH File	1 KB
CMakeFiles	2021/1/21 14:47	File folder	
debug	2021/1/21 14:47	File folder	

7. 打开 J-Link GDB server 然后连接设备。

гие неі	0				
GDB	Waiting for connection			Stay on top	
J-Link	Connected	SWD	4000 kHz	Show log win	ndow
Device	MIMXRT1176xxxA_M7 (	3.30V	little endian	Generate logf	file
				Verify downlo	bad
Downlo	ading 16080 bytes @ ad ading 10476 bytes @ ad	dress 0x000042E0 dress 0x000081B0			
Downlo Downlo Downlo Downlo Downlo Downlo Writin Receiv Starti	ading 16080 bytes @ ad ading 10476 bytes @ ad ading 8 bytes @ addres ading 4 bytes @ addres ading 4 bytes @ addres ading 176 bytes @ addr g register (PC = 0x ed monitor command: go ng target CPU	dress 0x000042E0 dress 0x000081B0 s 0x0000AA9C s 0x0000AA4 s 0x0000AAA8 ess 0x0000AAA8 ess 0x0000AAAC 4bc)			
Downlo Downlo Downlo Downlo Downlo Writin Receiv Starti GDB cl	ading 16080 bytes @ ad ading 10476 bytes @ ad ading 8 bytes @ addres ading 4 bytes @ addres ading 4 bytes @ addres ading 176 bytes @ addre g register (PC = 0x ed monitor command: go ng target CPU osed TCP/IP connection	dress 0x000042E0 dress 0x000081B0 s 0x0000AA9C s 0x0000AA4 s 0x0000AA8 ress 0x0000AA8 4bc)			~

8. 打开 GCC ARM Embedded tool chain command window。打开的方式是从 **Start menu** 里,找到 **GNU Tools ARM Embedded <version>** 然后选择 **GCC Command Prompt**。

GCC Command Prompt	_	×
C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-q4-major>_		ŕ
图 27. 打开 gcc command prompt		

9. 运行 arm-none-eabi-gdb.exe <application\_name>.elf. .elf 文件会在 debug 文件夹里。比如这个例子就是 arm-none-eabigdb.exe power\_mode\_switch\_bm\_core0.elf。



#### 10. 运行以下命令:

- **a**. Target remote localhost:2331
- **b.** Monitor reset
- $\boldsymbol{\mathsf{C}}.$  Monitor halt
- **d.** Load

```
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from C:\SDK_2.9.0_MIMXRT1170-EVK\boards\evkmimxrt1170\demo_apps\pow
(gdb) target remote localhost:2331
Remote debugging using localhost:2331
 xdeadbeee in ?? ()
(gdb) monitor reset
Resetting target
(gdb) monitor halt
(gdb) <u>l</u>oad
Loading section .interrupts, size 0x400 1ma 0x0
Loading section .text, size 0xa69c 1ma 0x400
Loading section .ARM, size 0x8 1ma 0xaa9c
Loading section .init_array, size 0x4 1ma 0xaaa4
Loading section .fini_array, size 0x4 lma 0xaaa8
Loading section .data, size 0xb0 lma 0xaaac
Start address 0x000004bc, load size 43868
Transfer rate: 96 KB/sec, 5483 bytes/write.
(gdb)
图 29. 运行 debug 命令
```

11. 运行命令 monitor go 来开始程序,结果将会显示在串口上。

🖻 COM10 - PuTTY	—		×
This is single core M7.	Minim	ize	$\sim$
CPU wakeup source 0x0			
**************************************			
System previous setpoint is O System current setpoint is O M7 previous CPU mode is RUN M7 current CPU mode is RUN M7 CLK is 696 MHz			
Please select the desired operation: Press A to demonstrate typical set point transition. Press B to demonstrate cpu mode switch in setpoint 0.			
Waiting for select			
图 30. 程序输出			

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.

**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.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. 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, 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, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, 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, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorlQ, QorlQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, elQ, 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, 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 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.

© NXP B.V. 2021.

#### All rights reserved.

For more information, please visit: http://www.nxp.com For sales office addresses, please send an email to: salesaddresses@nxp.com

> Date of release: February 2, 2021 Document identifier: AN13116

# arm