AN13116
如何在单核 RT1170 上进入 STBY 模式?

Rev. 0 — February 2, 2021

Application Note

目录

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® Cortex®-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。

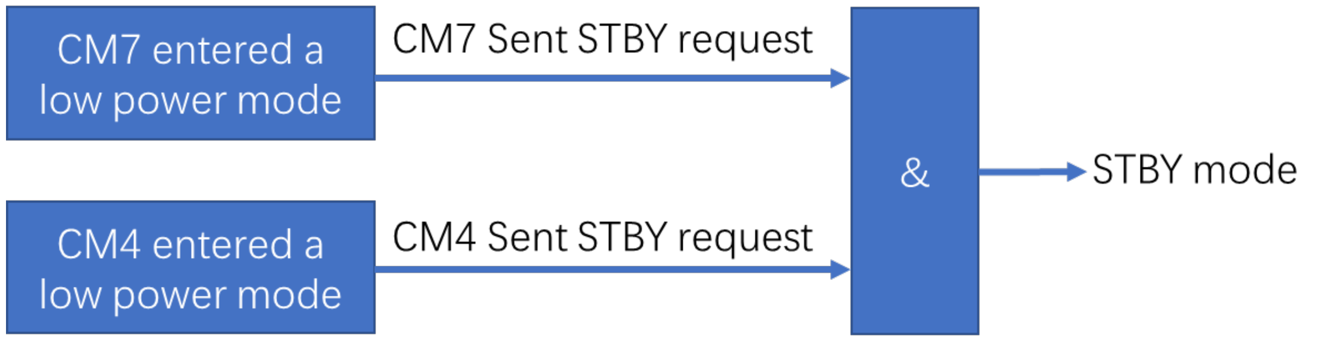


图 1. STBY 双核原则

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:

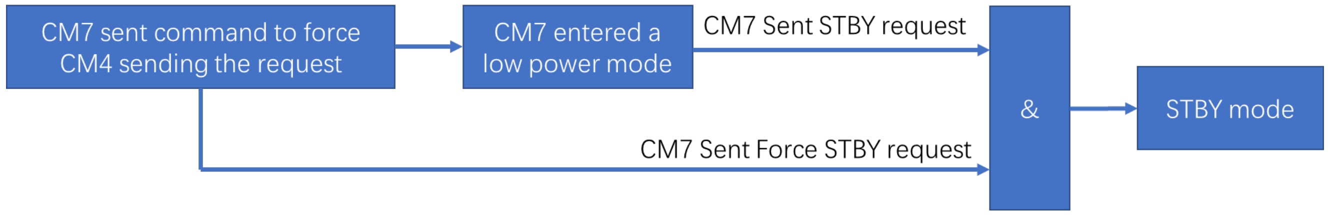

CM7 sent command to force CM4 sending the request
CM7 entered a low power mode
CM7 Sent STBY request
CM7 Sent Force STBY request
&
STBY mode

图 2. STBY 单核原则

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 0xFFFFFFFF 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

	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。

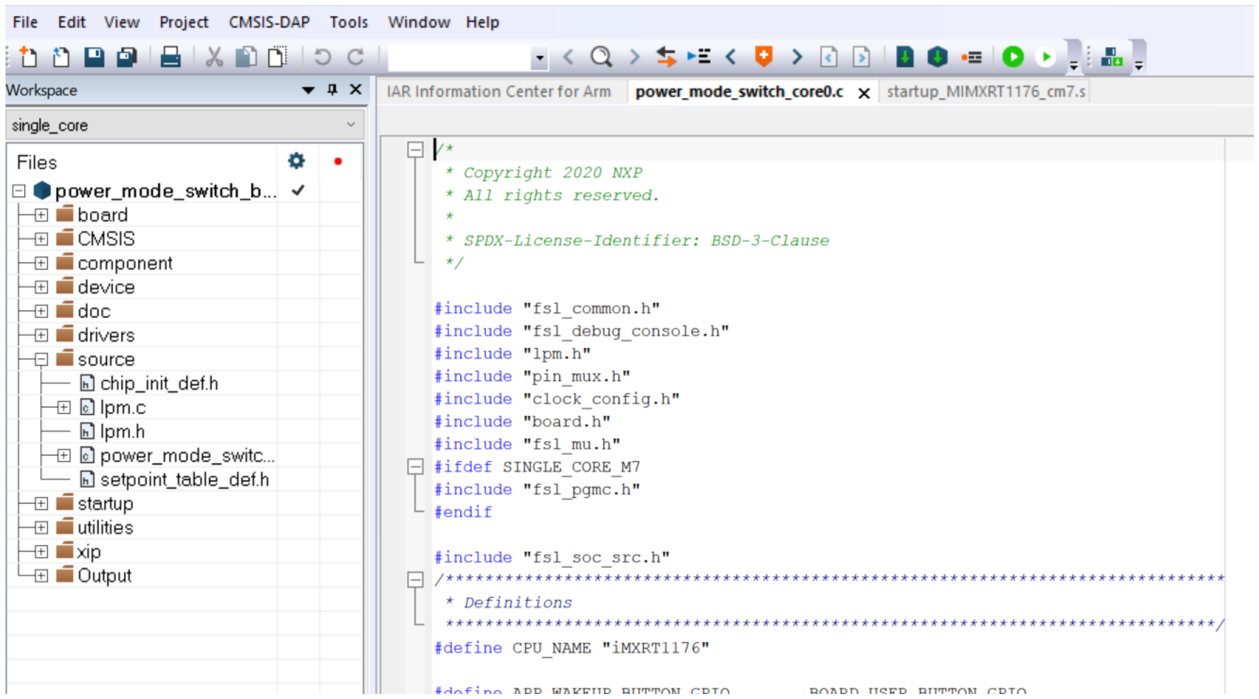
6 如何在 SDK 里创建一个单核项目.

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>\boards\evkmimxrt1170\demo_apps\power_mode_switch\bm\core0\iar\power_mode_switch_bm_core0.eww*
2. 将项目导入 IAR IDE。



```
/*
 * Copyright 2020 NXP
 * All rights reserved.
 *
 * SPDX-License-Identifier: BSD-3-Clause
 */

#include "fsl_common.h"
#include "fsl_debug_console.h"
#include "lpm.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "board.h"
#include "fsl_mu.h"
#ifdef SINGLE_CORE_M7
#include "fsl_pgmc.h"
#endif

#include "fsl_soc_src.h"
/*******************************************************************************
 * Definitions
 ******************************************************************************/
#define CPU_NAME "iMXRT1176"

#define APP_WAKEUP_BUTTON_GPIO          BOARD_USER_BUTTON_GPIO
```

图 3. 在 IAR 中打开 demo

3. 默认的建立目标应该是 debug，如下图所示：

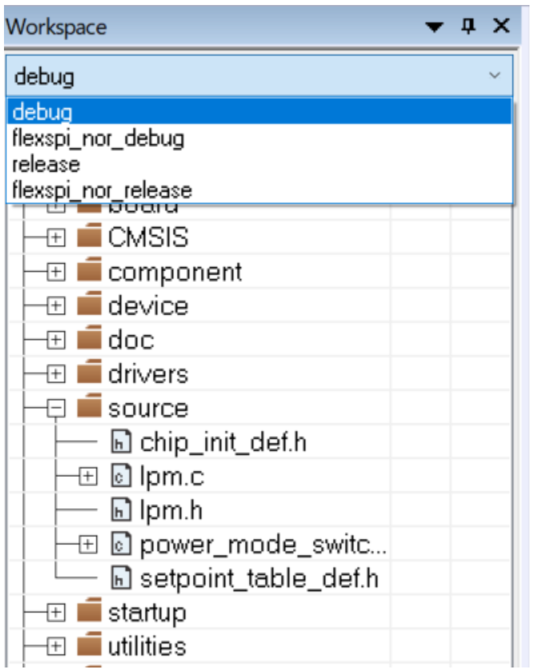


图 4. 建立目标

4. 在任务栏的 Project 选择 Edit Configurations：

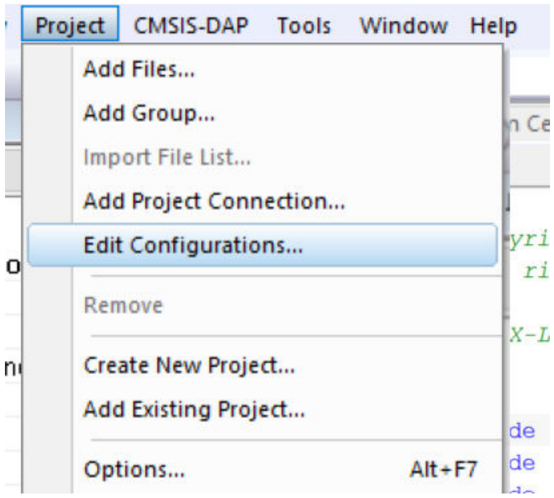
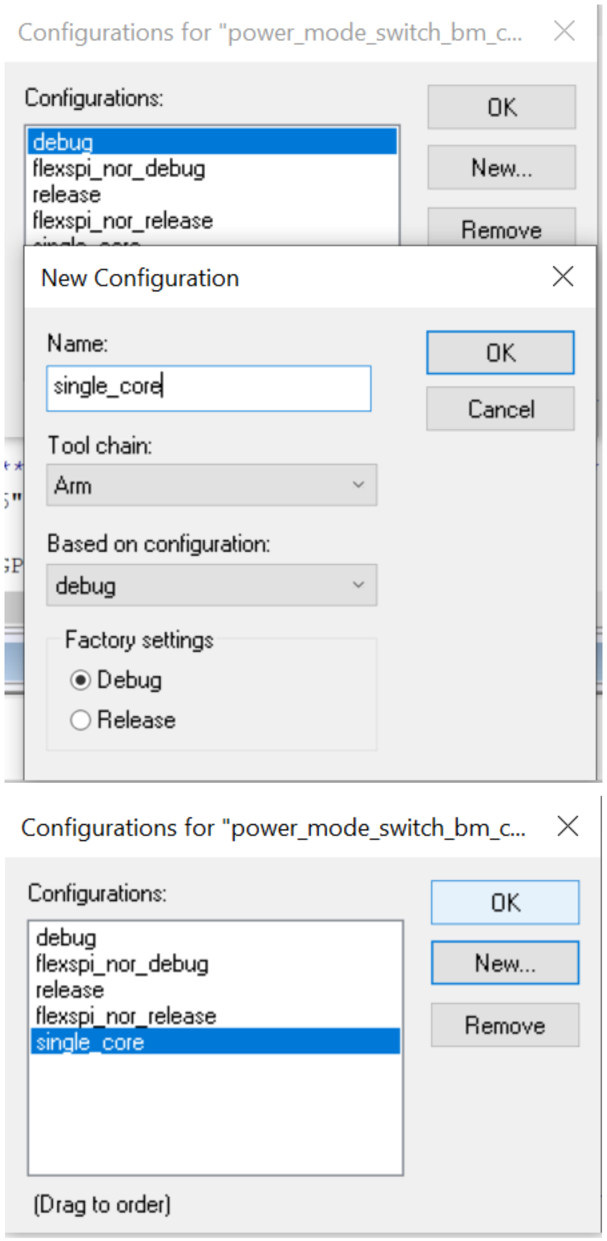


图 5. Edit configurations

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:
OK
New...
Remove

debug
flexspi_nor_debug
release
flexspi_nor_release

New Configuration

Name: single_core

Tool chain: Arm

Based on configuration: debug

Factory settings
(● Debug) ○ Release

OK
Cancel
Configurations for "power_mode_switch_bm_c...
Configurations:
OK
New...
Remove

debug
flexspi_nor_debug
release
flexspi_nor_release
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：

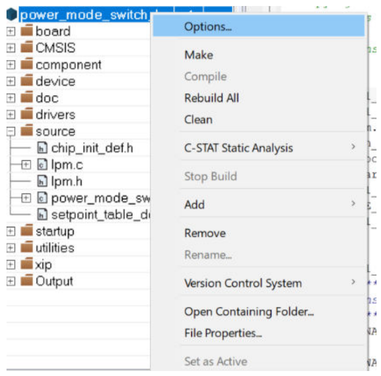
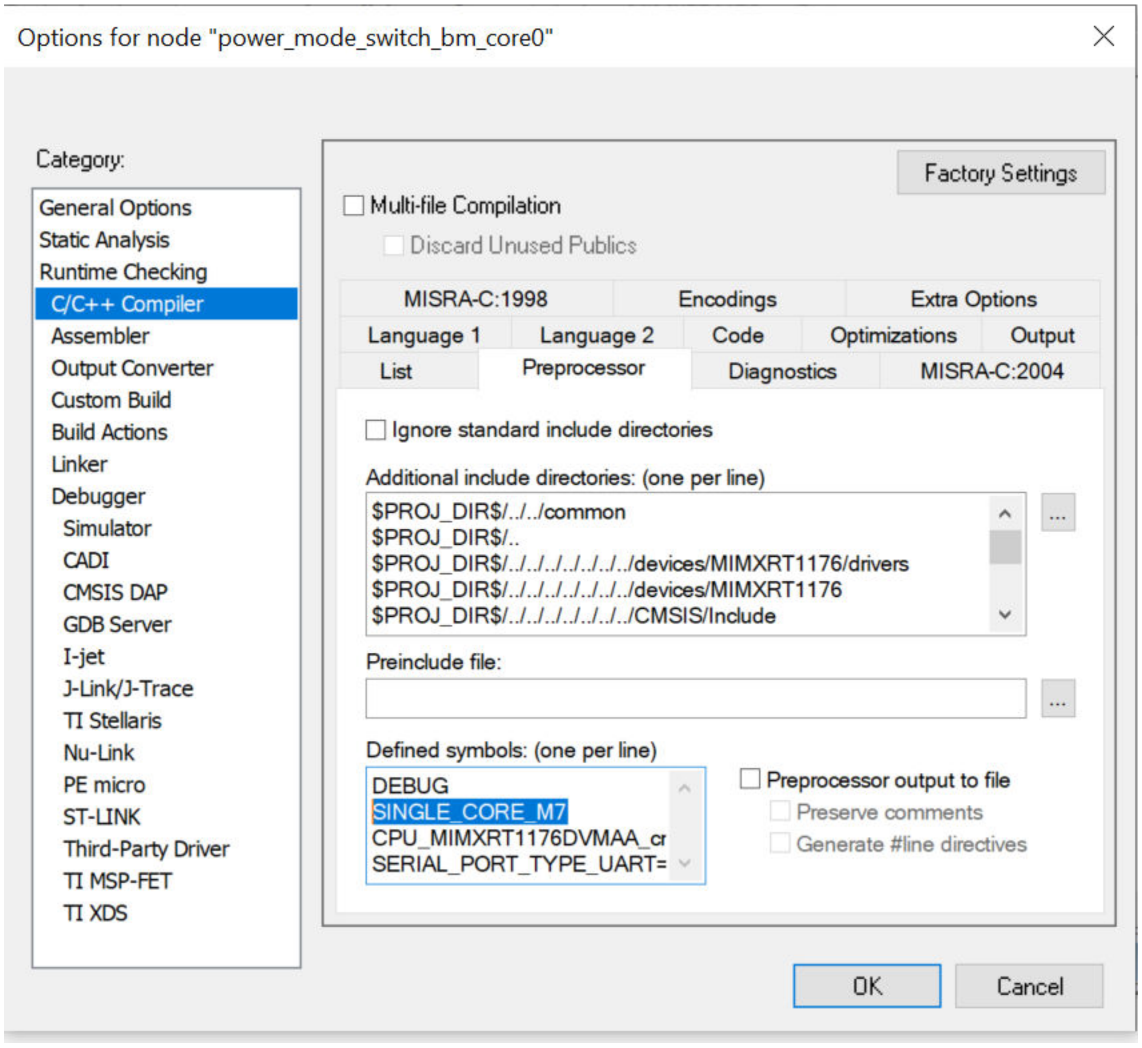


图 7. Options

a. 添加宏 SINGLE_CORE_M7 到 Preprocessor of C/C++ Compiler。



Additional include directories: (one per line)

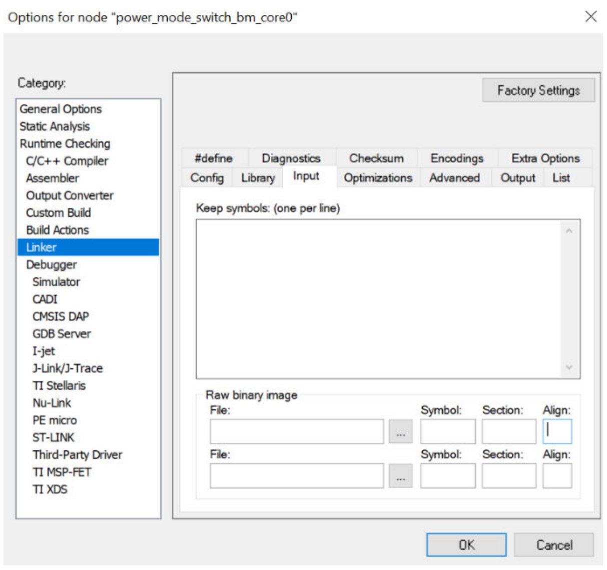
```
$PROJ_DIR$/../../common
$PROJ_DIR$/..
$PROJ_DIR$/../../../../../../../devices/MIMXRT1176/drivers
$PROJ_DIR$/../../../../../../../devices/MIMXRT1176
$PROJ_DIR$/../../../../../../../CMSIS/Include
```

Defined symbols: (one per line)

```
DEBUG
SINGLE_CORE_M7
CPU_MIMXRT1176DVMAA_cr
SERIAL_PORT_TYPE_UART=
```

图 8. 添加宏 preprocessor

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



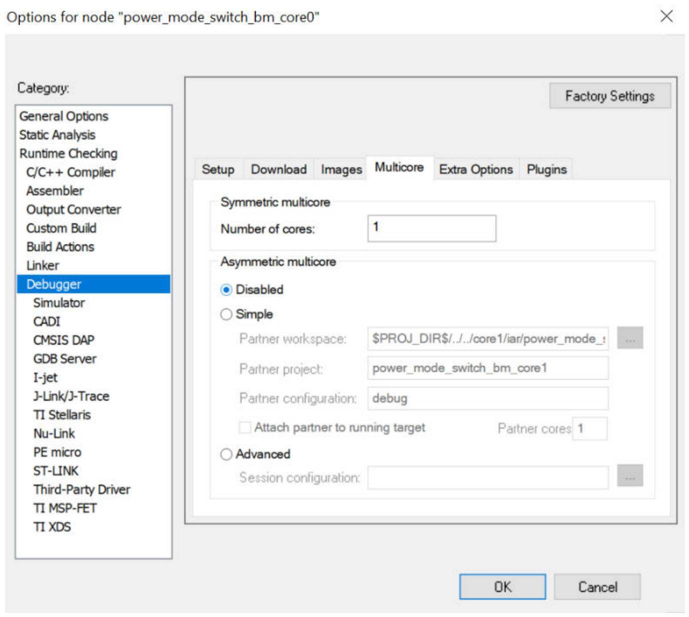
Options for node "power_mode_switch_bm_core0"

Keep symbols: (one per line)

Raw binary image
File: Symbol: Section: Align:
File: Symbol: Section: Align:

图 9. 清除 input

c. 在 Debugger 部分，选择 Multicore 里的 Disabled：



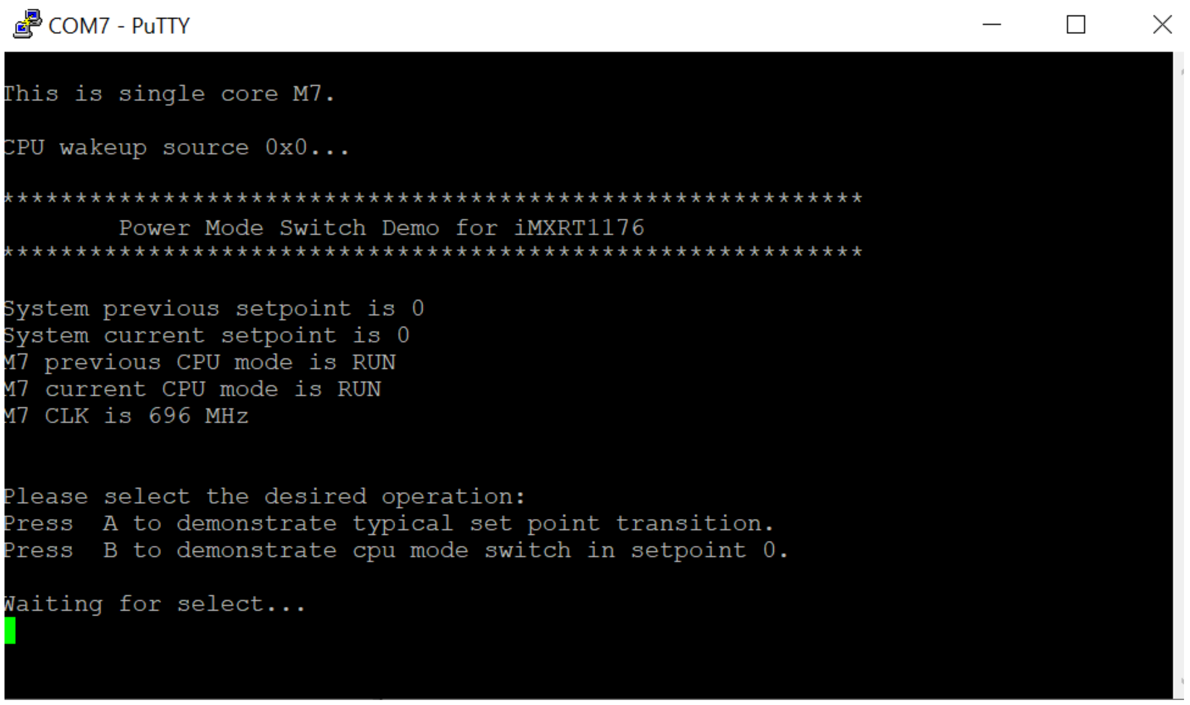
Options for node "power_mode_switch_bm_core0"

Symmetric multicore
Number of cores: 1

Asymmetric multicore
[✓ Disabled]
Simple
Partner workspace: $PROJ_DIR$/../../core1/iar/power_mode_s
Partner project: power_mode_switch_bm_core1
Partner configuration: debug
Attach partner to running target Partner cores: 1
Advanced
Session configuration:

图 10. Disabled multicore

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：

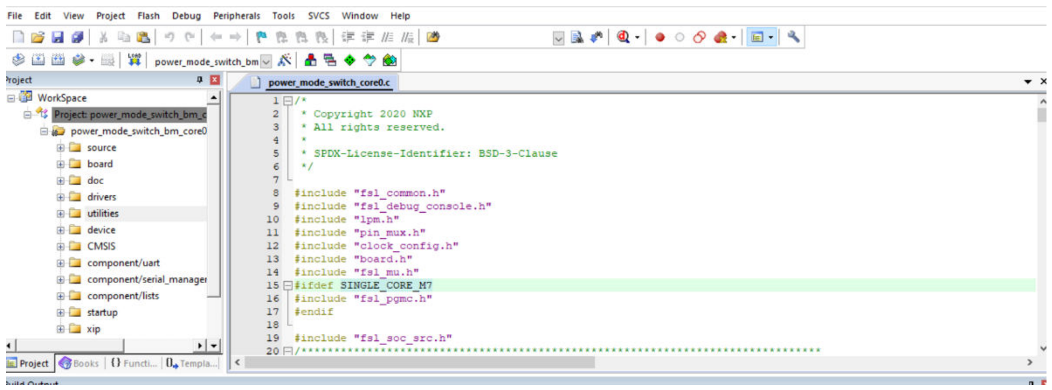


```
This is single core M7.

CPU wakeup source 0x0...

***********************************************************
        Power Mode Switch Demo for iMXRT1176
***********************************************************

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 11. 程序输出

6.2 用 Keil® MDK/µVision 运行程序

1. 用 keil 打开 power_mode_switch 程序。程序的地址在 *<install_dir>\boards\evkmimxrt1170\demo_apps\power_mode_switch\bm\core0\mdk\power_mode_switch_bm_core0.uvmpw*

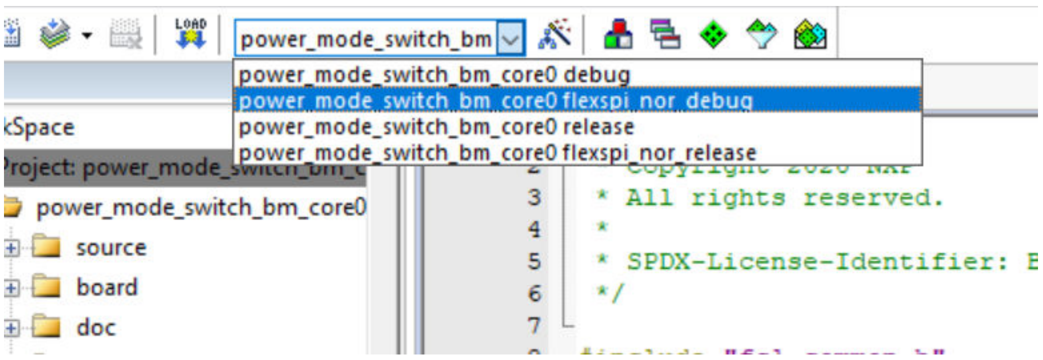


File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
power_mode_switch_bm
Project
WorkSpace
Project: power_mode_switch_bm_c
power_mode_switch_bm_core0
source
board
doc
drivers
utilities
device
CMSIS
component/uart
component/serial_manager
component/lists
startup
xip
Project Books Functi... Templa...
power_mode_switch_core0.c

```
/*
 * Copyright 2020 NXP
 * All rights reserved.
 *
 * SPDX-License-Identifier: BSD-3-Clause
 */

#include "fsl_common.h"
#include "fsl_debug_console.h"
#include "lpm.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "board.h"
#include "fsl_mu.h"
#ifdef SINGLE_CORE_M7
#include "fsl_pgmc.h"
#endif

#include "fsl_soc_src.h"
/*******************************************************************************
```

图 12. 打开 demo

2. 把目标改为 flexsip_nor_debug：



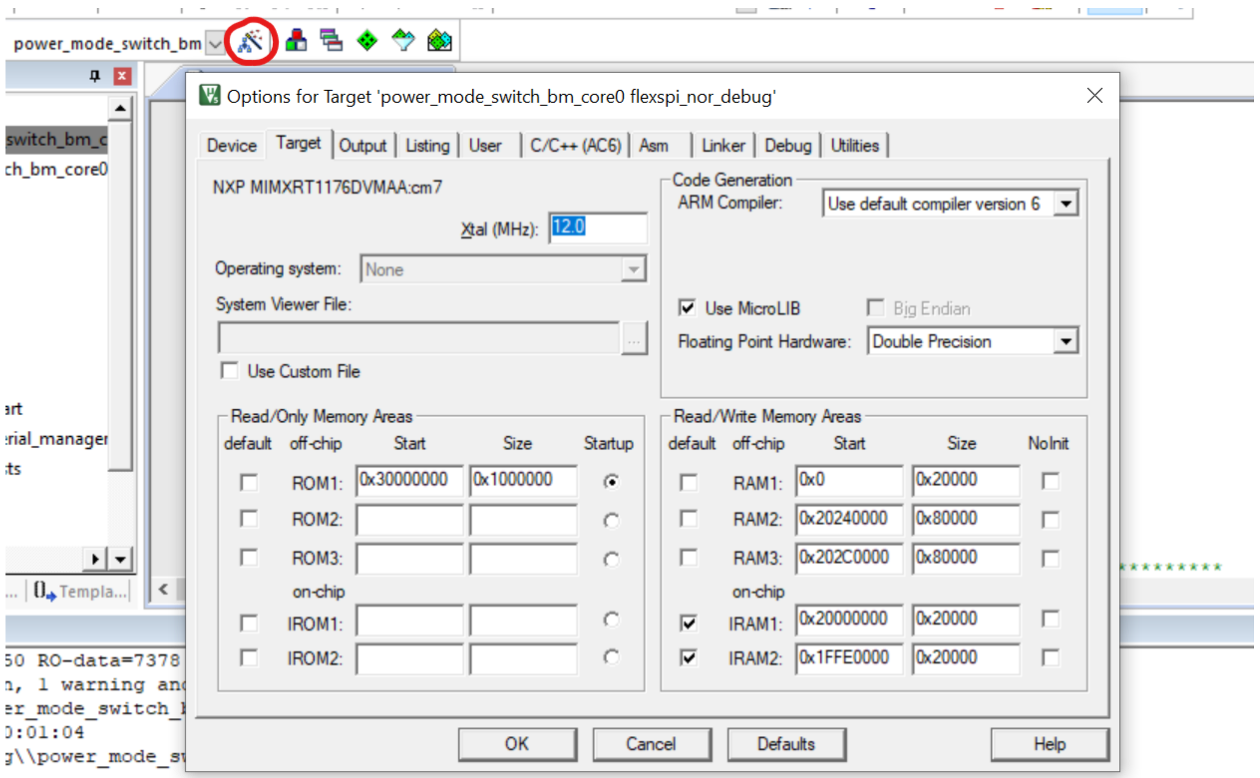
power_mode_switch_bm

power_mode_switch_bm_core0 debug
power_mode_switch_bm_core0 flexspi_nor_debug
power_mode_switch_bm_core0 release
power_mode_switch_bm_core0 flexspi_nor_release

```
 * All rights reserved.
 *
 * SPDX-License-Identifier: B
 */
```

图 13. 修改目标

3. 打开目标的 Options：



Options for Target 'power_mode_switch_bm_core0 flexspi_nor_debug'

NXP MIMXRT1176DVMAA:cm7

Xtal (MHz): 12.0

Operating system: None

System Viewer File:

Use Custom File

Code Generation

ARM Compiler: Use default compiler version 6

☑ Use MicroLIB ☐ Big Endian

Floating Point Hardware: Double Precision

Read/Only Memory Areas

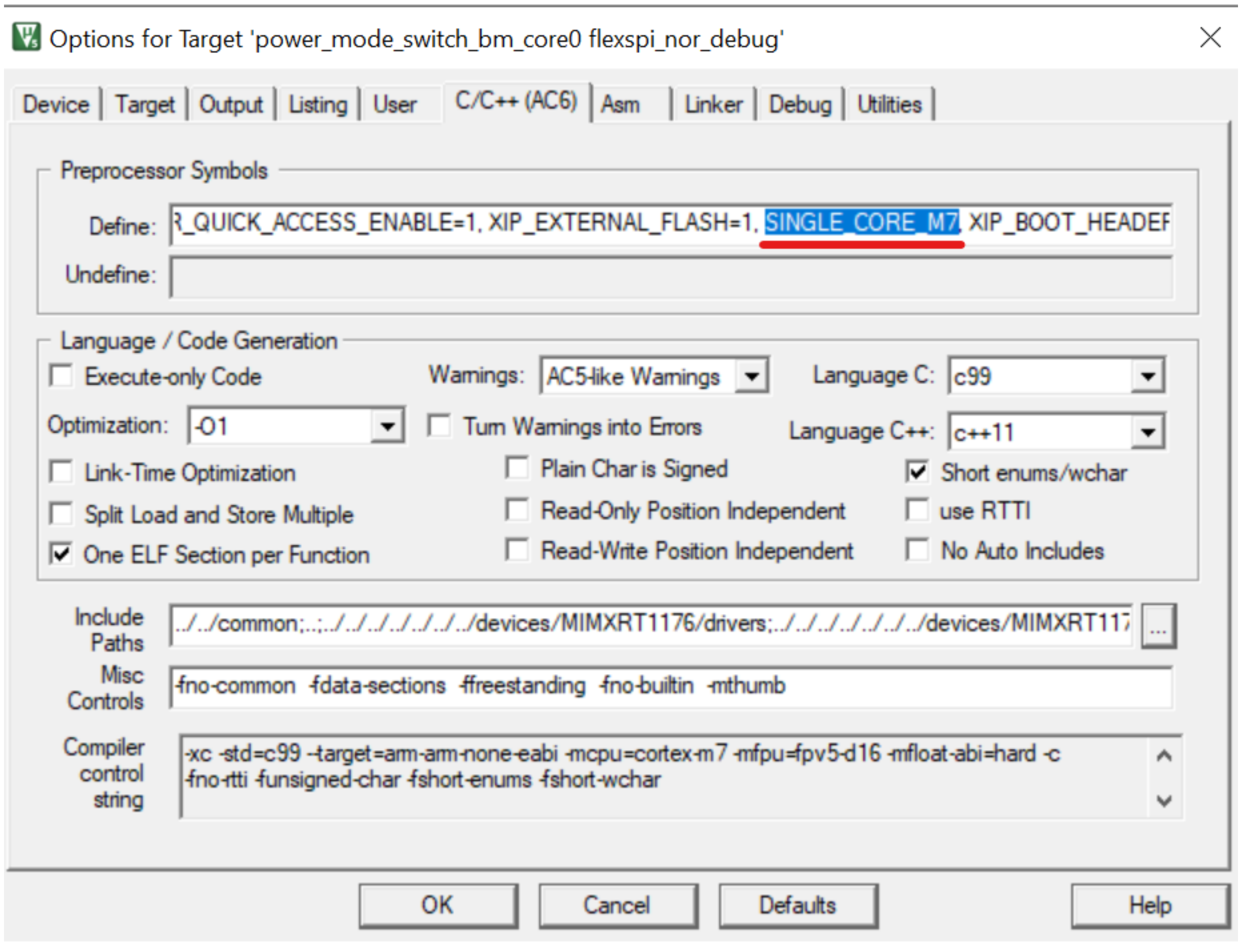
default	off-chip	Start	Size	Startup
☐	ROM1:	0x30000000	0x1000000	●
☐	ROM2:			○
☐	ROM3:			○
	on-chip			
☐	IROM1:			○
☐	IROM2:			○

Read/Write Memory Areas

default	off-chip	Start	Size	NoInit
☐	RAM1:	0x0	0x20000	☐
☐	RAM2:	0x20240000	0x80000	☐
☐	RAM3:	0x202C0000	0x80000	☐
	on-chip			
☑	IRAM1:	0x20000000	0x20000	☐
☑	IRAM2:	0x1FFE0000	0x20000	☐

图 14. Open options

4. 在 C/C++ 里, 添加 SINGLE_CORE_M7 到 Define：



Preprocessor Symbols

Define: R_QUICK_ACCESS_ENABLE=1, XIP_EXTERNAL_FLASH=1, SINGLE_CORE_M7, XIP_BOOT_HEADER

Undefine:

Language / Code Generation

- [ ] Execute-only Code
- Warnings: AC5-like Warnings
- Language C: c99
- Optimization: -O1
- [ ] Turn Warnings into Errors
- Language C++: c++11
- [ ] Link-Time Optimization
- [ ] Plain Char is Signed
- [✓] 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

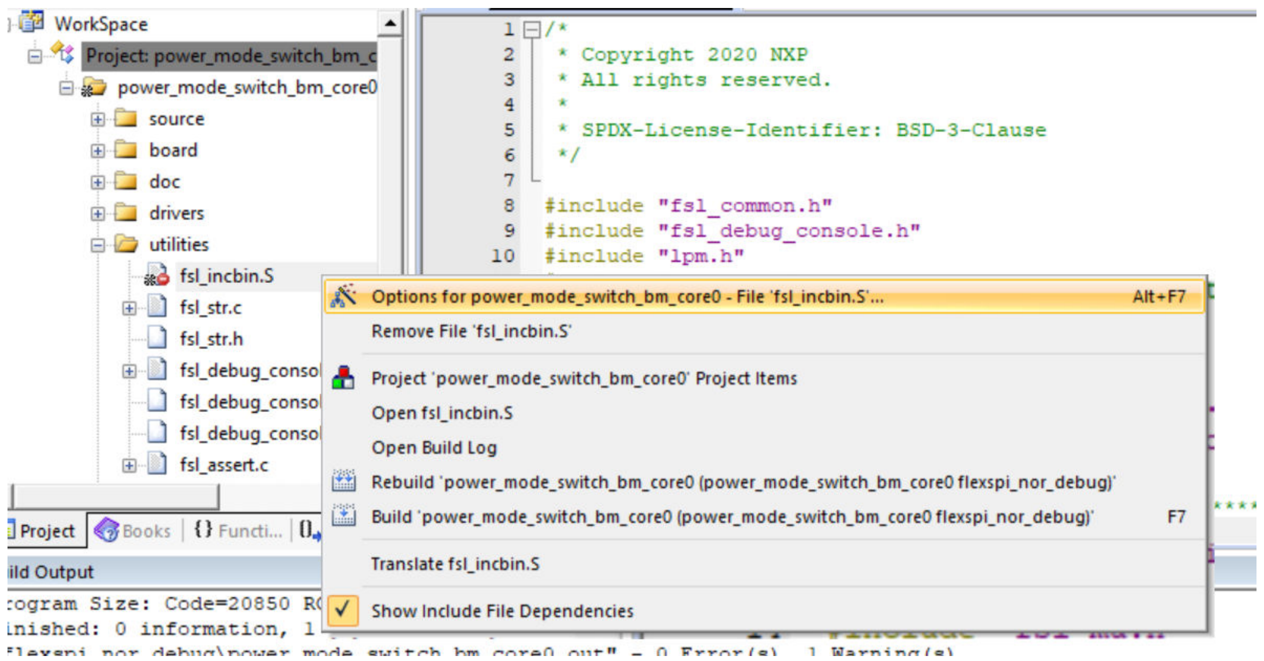
Include Paths: ../../common;..;../../../../../../../devices/MIMXRT1176/drivers;../../../../../../../devices/MIMXRT117

Misc Controls: -fno-common -fdata-sections -ffreestanding -fno-builtin -mthumb

Compiler control string:
```
-xc -std=c99 -target=arm-arm-none-eabi -mcpu=cortex-m7 -mfpu=fpv5-d16 -mfloat-abi=hard -c
-fno-rtti -funsigned-char -fshort-enums -fshort-wchar
```

图 15. 添加宏到 define

5. 打开 fsl_incbin.S 的 Options：

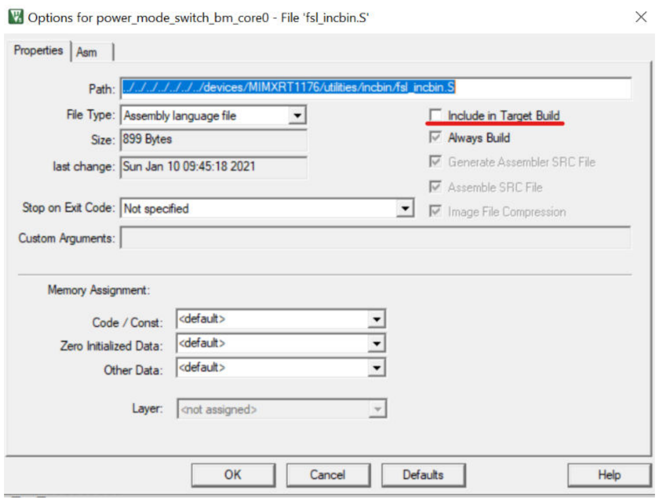

WorkSpace
Project: power_mode_switch_bm_c
power_mode_switch_bm_core0
source
board
doc
drivers
utilities
fsl_incbin.S
fsl_str.c
fsl_str.h
fsl_debug_consol
fsl_debug_consol
fsl_debug_consol
fsl_assert.c
Project
Books
{} Functi...
Options for power_mode_switch_bm_core0 - File 'fsl_incbin.S'... Alt+F7
Remove File 'fsl_incbin.S'
Project 'power_mode_switch_bm_core0' Project Items
Open fsl_incbin.S
Open Build Log
Rebuild 'power_mode_switch_bm_core0 (power_mode_switch_bm_core0 flexspi_nor_debug)'
Build 'power_mode_switch_bm_core0 (power_mode_switch_bm_core0 flexspi_nor_debug)' F7
Translate fsl_incbin.S
Show Include File Dependencies
ild Output

```
/*
 * Copyright 2020 NXP
 * All rights reserved.
 *
 * SPDX-License-Identifier: BSD-3-Clause
 */

#include "fsl_common.h"
#include "fsl_debug_console.h"
#include "lpm.h"
```

```
rogram Size: Code=20850 R
inished: 0 information, 1
flexspi_nor_debug\power_mode_switch_bm_core0.out" - 0 Error(s), 1 Warning(s).
```

图 16. 打开 Options

6. 去掉选项 Include in Target Build：


Options for power_mode_switch_bm_core0 - File 'fsl_incbin.S'
Properties
Asm
OK
Cancel
Defaults
Help

Path: ../../../../../../devices/MIMXRT1176/utilities/incbin/fsl_incbin.S

File Type: Assembly language file

Size: 899 Bytes

last change: Sun Jan 10 09:45:18 2021

Stop on Exit Code: Not specified

Custom Arguments:

☐ Include in Target Build
☑ Always Build
☑ Generate Assembler SRC File
☑ Assemble SRC File
☑ Image File Compression

Memory Assignment:

Code / Const: <default>

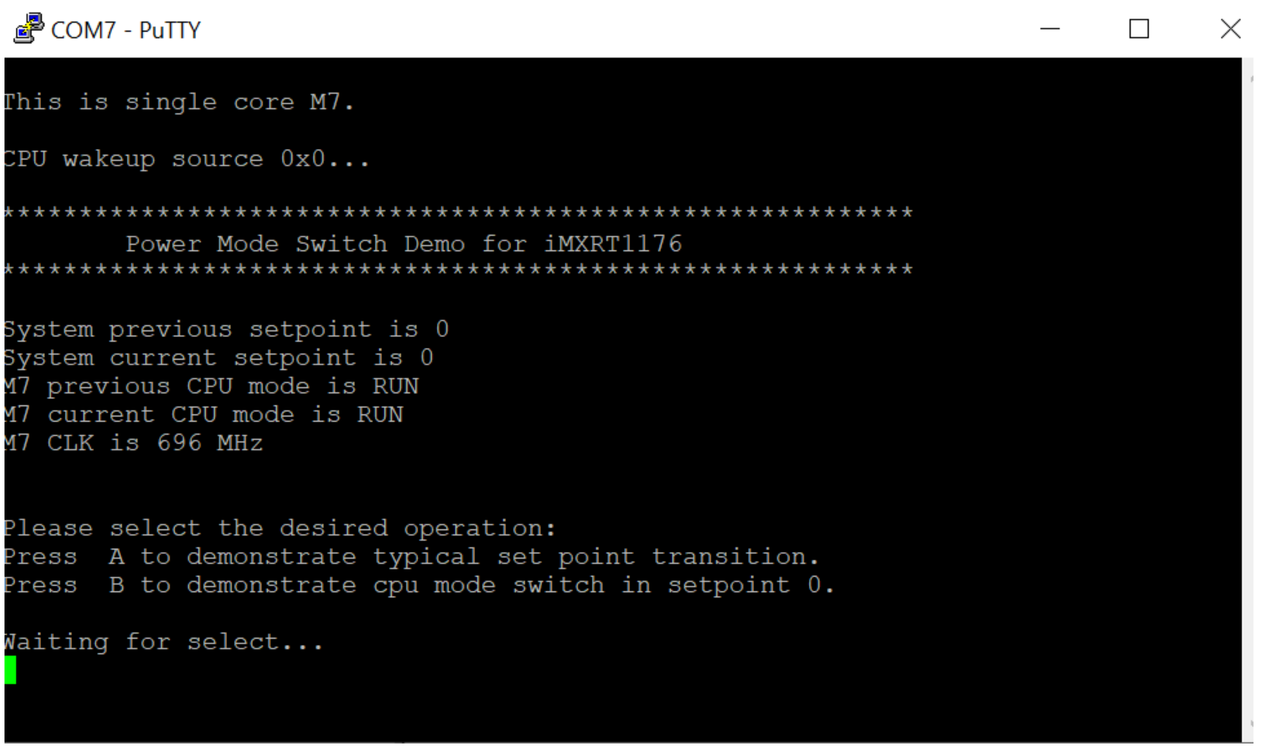
Zero Initialized Data: <default>

Other Data: <default>

Layer: <not assigned>

图 17. 去掉 include in Target Build

7. Debug 然后运行程序。



```
This is single core M7.

CPU wakeup source 0x0...

***********************************************************
        Power Mode Switch Demo for iMXRT1176
***********************************************************

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

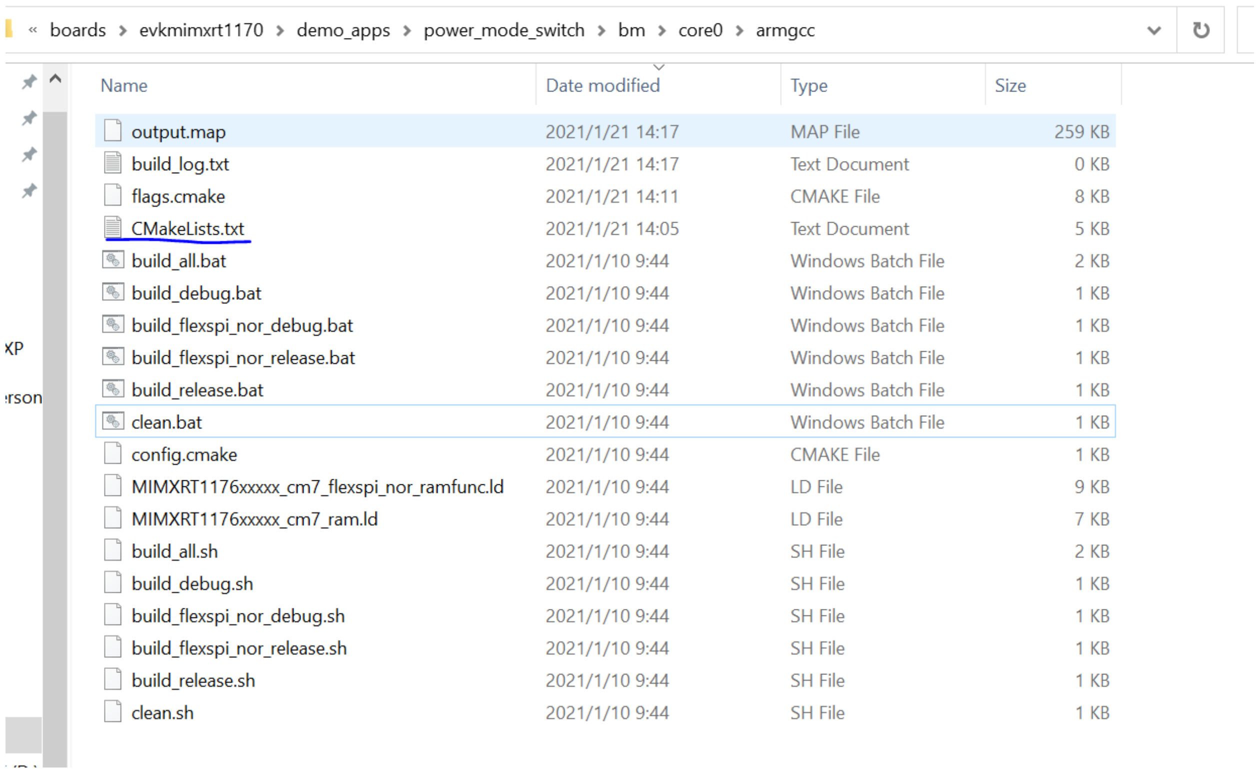
6.3 在 Arm® GCC 上运行 demo

本节展示了如何用 GCC IDE 运行 demo。

注意

在此之前，请确保 CMake 和 Segger J-Link 已经安装。

1. 在项目的文件夹里打开 cmakelist.txt：

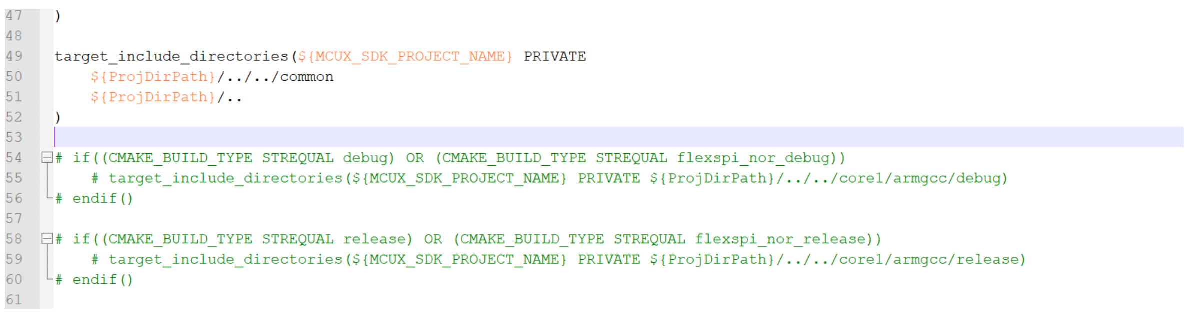


« boards › evkmimxrt1170 › demo_apps › power_mode_switch › bm › core0 › armgcc

Name	Date modified	Type	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
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

图 19. 打开 CMakeLists.txt

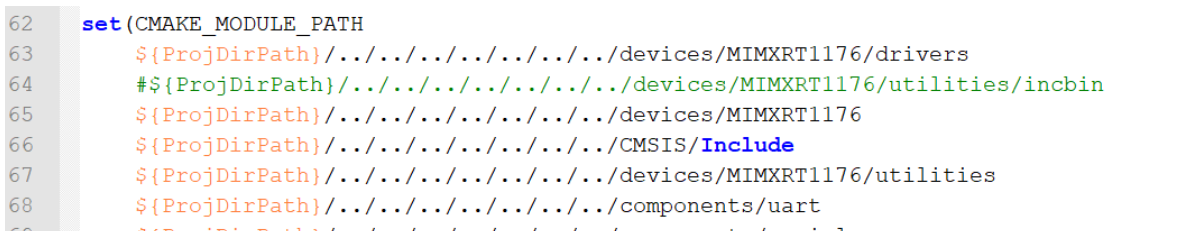
2. 有三部分的程序需要被删除或者注释掉：
   a. 包含 core1 的代码。



```
47  )
48
49  target_include_directories(${MCUX_SDK_PROJECT_NAME} PRIVATE
50      ${ProjDirPath}/../../common
51      ${ProjDirPath}/..
52  )
53
54  # if((CMAKE_BUILD_TYPE STREQUAL debug) OR (CMAKE_BUILD_TYPE STREQUAL flexspi_nor_debug))
55      # target_include_directories(${MCUX_SDK_PROJECT_NAME} PRIVATE ${ProjDirPath}/../../core1/armgcc/debug)
56  # endif()
57
58  # if((CMAKE_BUILD_TYPE STREQUAL release) OR (CMAKE_BUILD_TYPE STREQUAL flexspi_nor_release))
59      # target_include_directories(${MCUX_SDK_PROJECT_NAME} PRIVATE ${ProjDirPath}/../../core1/armgcc/release)
60  # endif()
61
```

图 20. 删除或注释 54-60 行

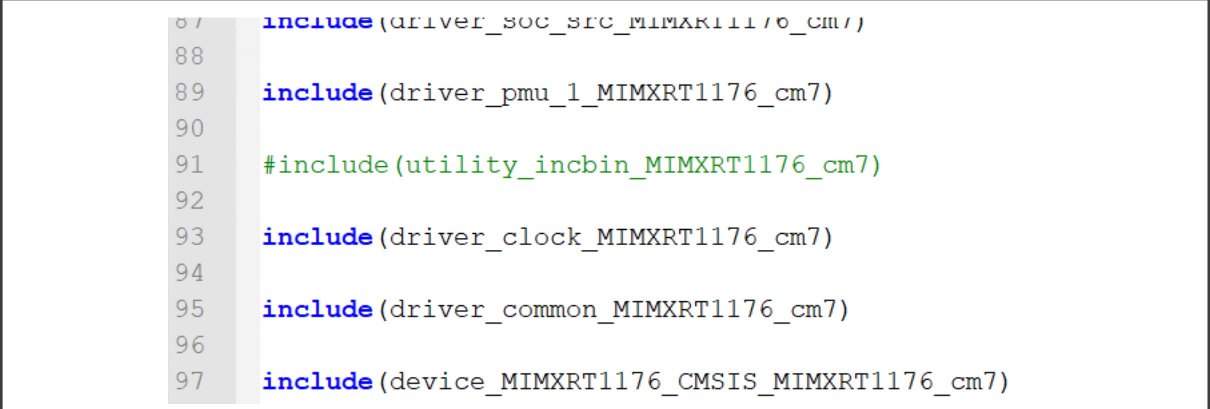
   b. Incbin 的路径。



```
62  set(CMAKE_MODULE_PATH
63      ${ProjDirPath}/../../../../../../../../devices/MIMXRT1176/drivers
64      #${ProjDirPath}/../../../../../../../../devices/MIMXRT1176/utilities/incbin
65      ${ProjDirPath}/../../../../../../../../devices/MIMXRT1176
66      ${ProjDirPath}/../../../../../../../../CMSIS/Include
67      ${ProjDirPath}/../../../../../../../../devices/MIMXRT1176/utilities
68      ${ProjDirPath}/../../../../../../../../components/uart
```

图 21. 删除或注释 64 行

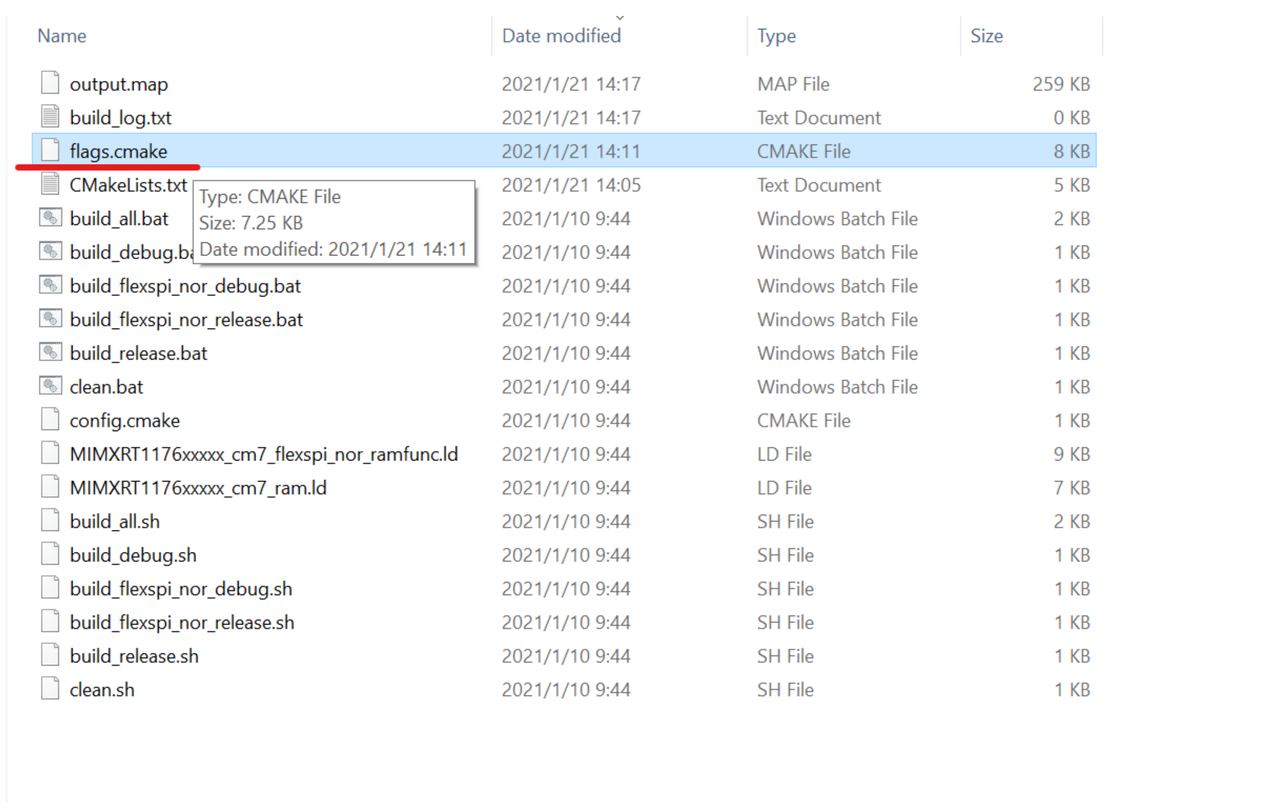
c. include incbin。



```
87  include(driver_soc_src_MIMXRT1176_cm7)
88
89  include(driver_pmu_1_MIMXRT1176_cm7)
90
91  #include(utility_incbin_MIMXRT1176_cm7)
92
93  include(driver_clock_MIMXRT1176_cm7)
94
95  include(driver_common_MIMXRT1176_cm7)
96
97  include(device_MIMXRT1176_CMSIS_MIMXRT1176_cm7)
```

图 22. 删除或注释 91 行

3. 打开 flags.cmake 文件：

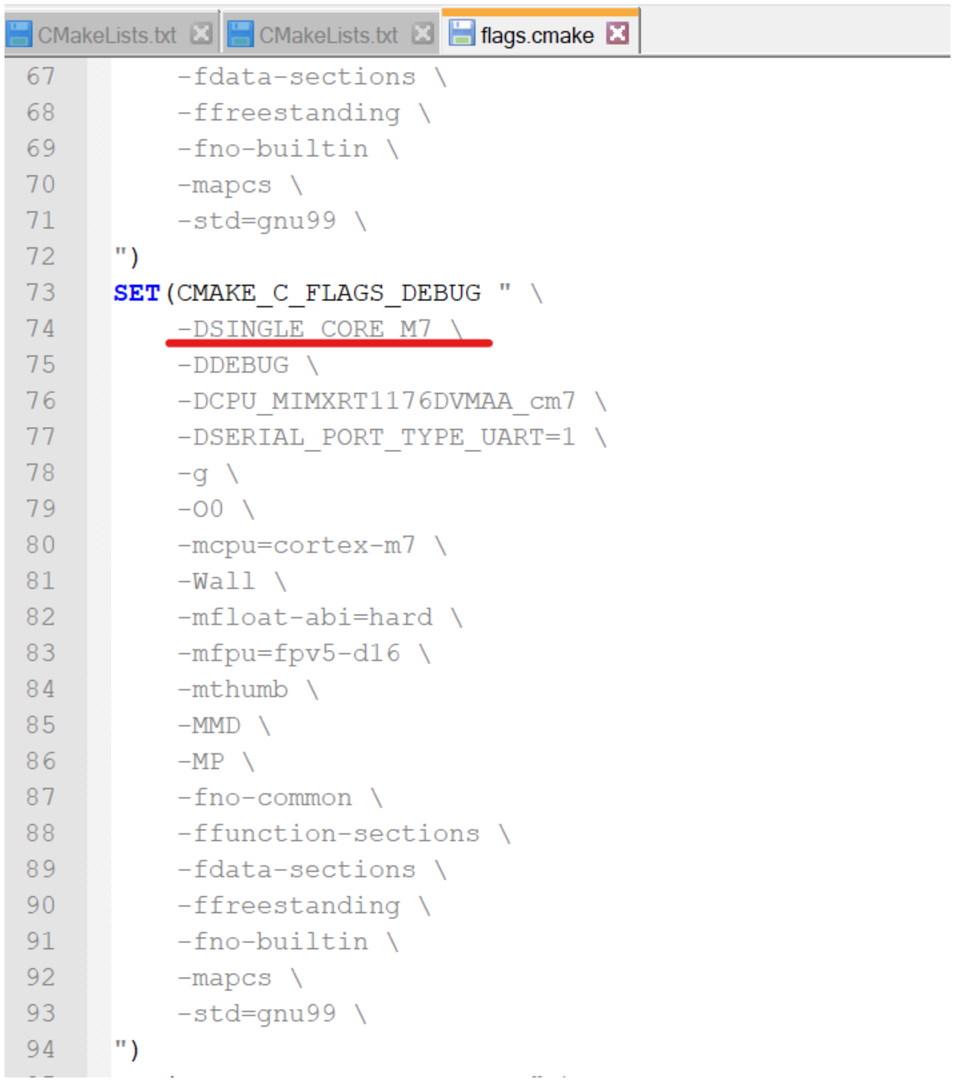


Name	Date modified	Type	Size
output.map	2021/1/21 14:17	MAP File	259 KB
build_log.txt	2021/1/21 14:17	Text Document	0 KB
flags.cmake	2021/1/21 14:11	CMAKE File	8 KB
CMakeLists.txt	2021/1/21 14:05	Text Document	5 KB
build_all.bat	2021/1/10 9:44	Windows Batch File	2 KB
build_debug.ba	2021/1/10 9:44	Windows 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

Type: CMAKE File
Size: 7.25 KB
Date modified: 2021/1/21 14:11

图 23. 打开 flags.cmake

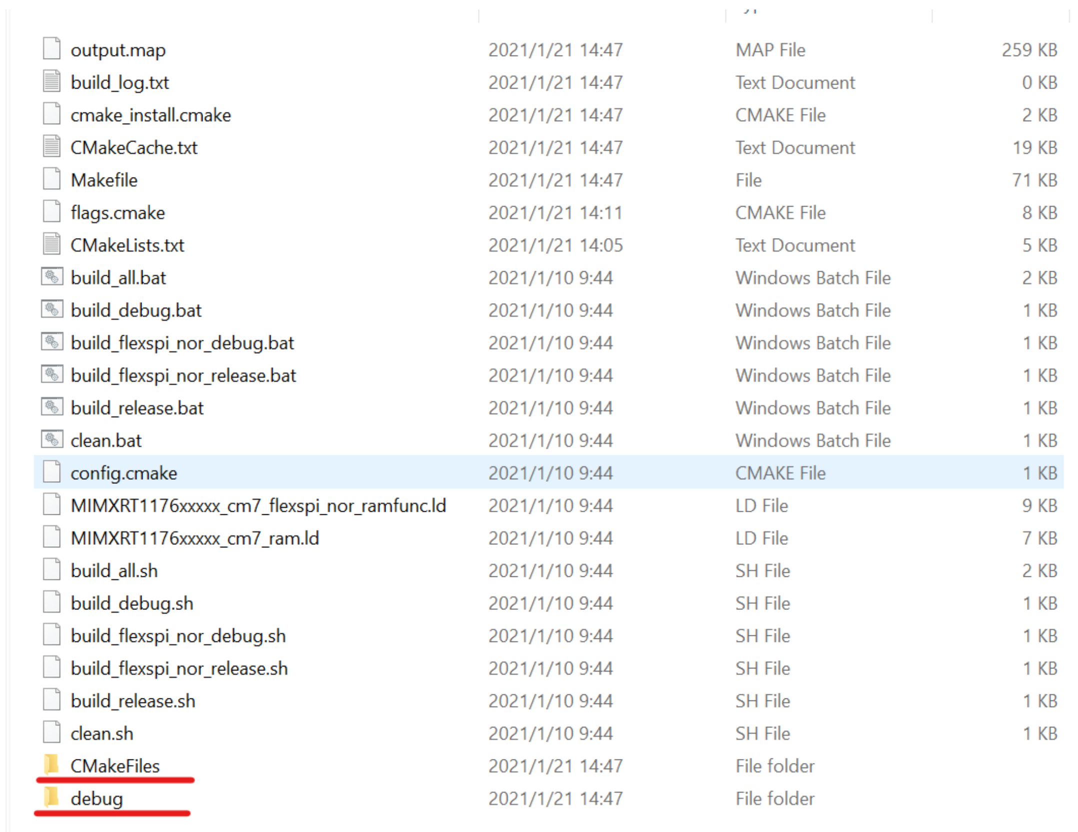
4. 添加宏 SINGLE_CORE_M7 到目标：



```
        -fdata-sections \
        -ffreestanding \
        -fno-builtin \
        -mapcs \
        -std=gnu99 \
")
SET(CMAKE_C_FLAGS_DEBUG " \
        -DSINGLE_CORE_M7 \
        -DDEBUG \
        -DCPU_MIMXRT1176DVMAA_cm7 \
        -DSERIAL_PORT_TYPE_UART=1 \
        -g \
        -O0 \
        -mcpu=cortex-m7 \
        -Wall \
        -mfloat-abi=hard \
        -mfpu=fpv5-d16 \
        -mthumb \
        -MMD \
        -MP \
        -fno-common \
        -ffunction-sections \
        -fdata-sections \
        -ffreestanding \
        -fno-builtin \
        -mapcs \
        -std=gnu99 \
")
```

图 24. 添加宏到目标

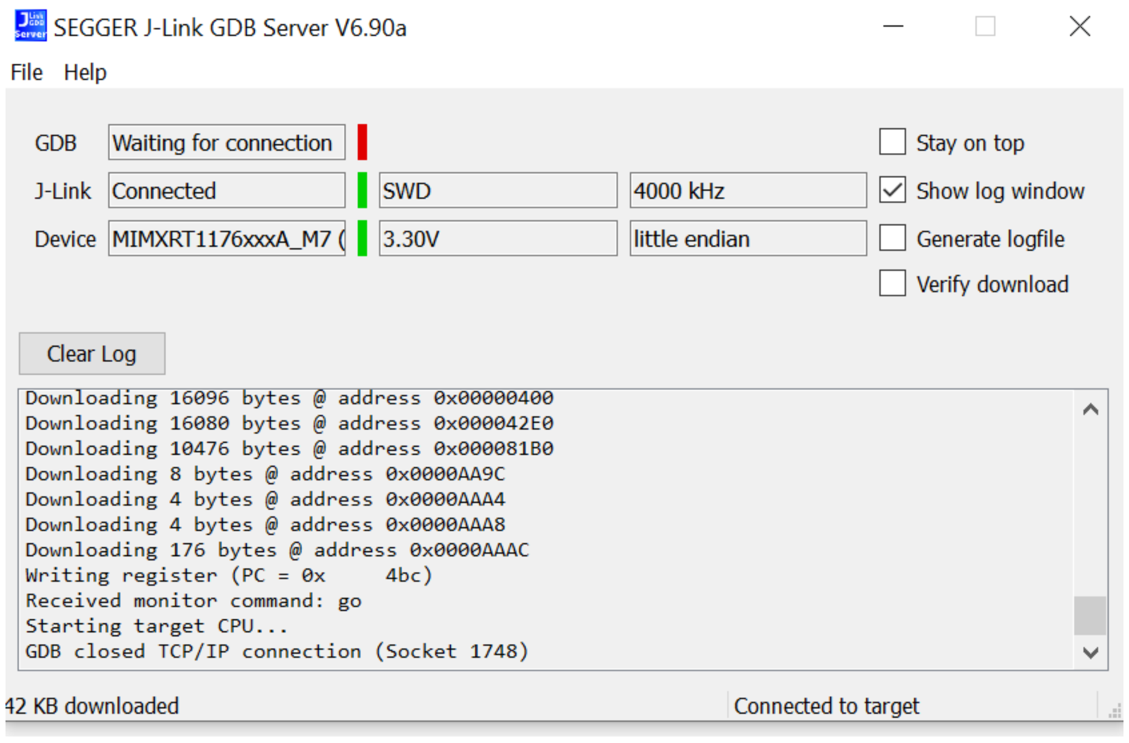
5. 修改并保存之后, 双击 build_debug.bat 来建立目标。
6. 如果成功的话, 有两个新的文件夹（CmakeFiles, debug）被建立。



output.map	2021/1/21 14:47	MAP File	259 KB
build_log.txt	2021/1/21 14:47	Text Document	0 KB
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
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
CMakeFiles	2021/1/21 14:47	File folder	
debug	2021/1/21 14:47	File folder	

图 25. Demo 建立成功

7. 打开 J-Link GDB server 然后连接设备。



GDB	Waiting for connection		
J-Link	Connected	SWD	4000 kHz
Device	MIMXRT1176xxxA_M7 (	3.30V	little endian

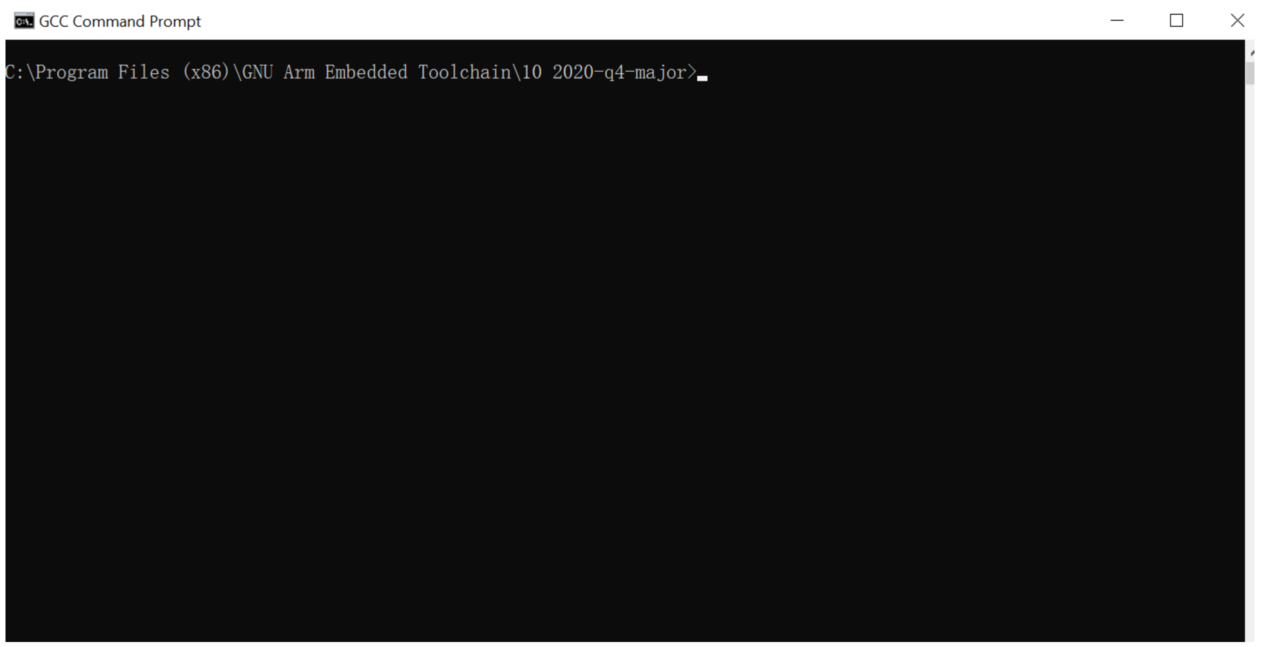
☐ Stay on top
✓ Show log window
☐ Generate logfile
☐ Verify download

```
Downloading 16096 bytes @ address 0x00000400
Downloading 16080 bytes @ address 0x000042E0
Downloading 10476 bytes @ address 0x000081B0
Downloading 8 bytes @ address 0x0000AA9C
Downloading 4 bytes @ address 0x0000AAA4
Downloading 4 bytes @ address 0x0000AAA8
Downloading 176 bytes @ address 0x0000AAAC
Writing register (PC = 0x      4bc)
Received monitor command: go
Starting target CPU...
GDB closed TCP/IP connection (Socket 1748)
```

42 KB downloaded   Connected to target

图 26. Segger J-link GDB server 连接成功

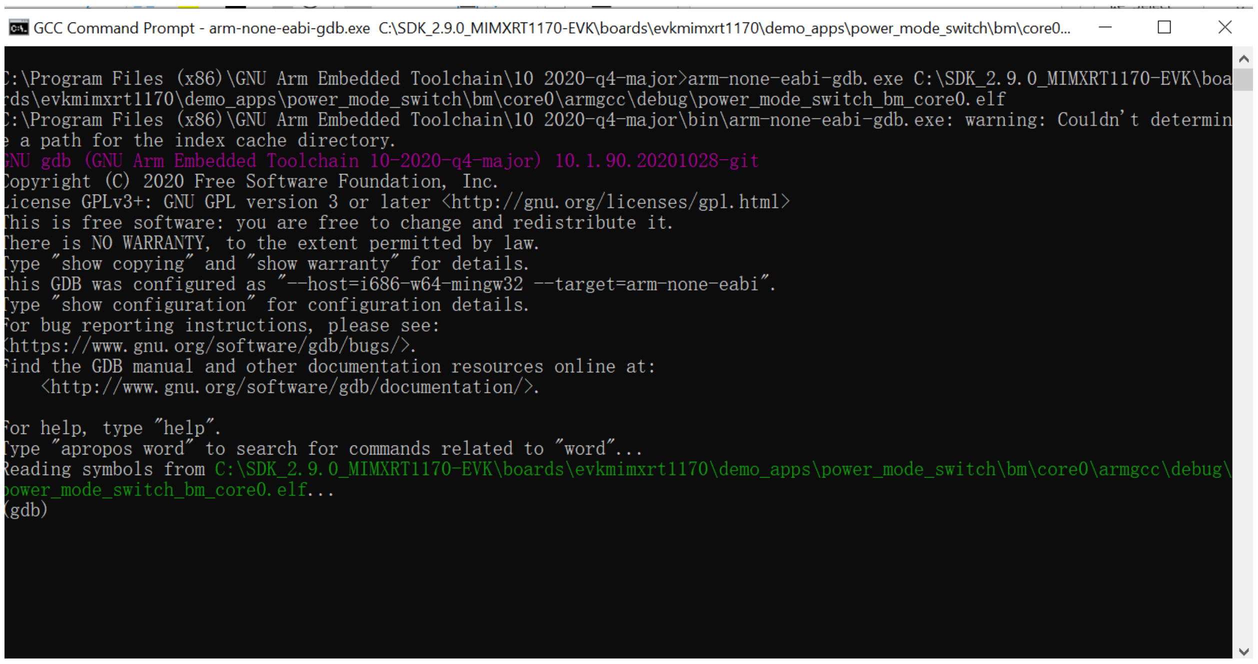
8. 打开 GCC ARM Embedded tool chain command window。打开的方式是从 **Start menu** 里， 找到 **GNU Tools ARM Embedded <version>** 然后选择 **GCC Command Prompt**。



```
C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-q4-major>
```

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-eabi-gdb.exe power_mode_switch_bm_core0.elf*。



```
GCC Command Prompt - arm-none-eabi-gdb.exe  C:\SDK_2.9.0_MIMXRT1170-EVK\boards\evkmimxrt1170\demo_apps\power_mode_switch\bm\core0...

:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-q4-major>arm-none-eabi-gdb.exe C:\SDK_2.9.0_MIMXRT1170-EVK\boa
rds\evkmimxrt1170\demo_apps\power_mode_switch\bm\core0\armgcc\debug\power_mode_switch_bm_core0.elf
:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-q4-major\bin\arm-none-eabi-gdb.exe: warning: Couldn't determin
e a path for the index cache directory.
NU gdb (GNU Arm Embedded Toolchain 10-2020-q4-major) 10.1.90.20201028-git
opyright (C) 2020 Free Software Foundation, Inc.
icense GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
his is free software: you are free to change and redistribute it.
here is NO WARRANTY, to the extent permitted by law.
ype "show copying" and "show warranty" for details.
his GDB was configured as "--host=i686-w64-mingw32 --target=arm-none-eabi".
ype "show configuration" for configuration details.
or bug reporting instructions, please see:
https://www.gnu.org/software/gdb/bugs/>.
ind the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

or help, type "help".
ype "apropos word" to search for commands related to "word"...
eading symbols from C:\SDK_2.9.0_MIMXRT1170-EVK\boards\evkmimxrt1170\demo_apps\power_mode_switch\bm\core0\armgcc\debug\
ower_mode_switch_bm_core0.elf...
(gdb)
```

图 28. 运行 arm-none-eabi-gdb

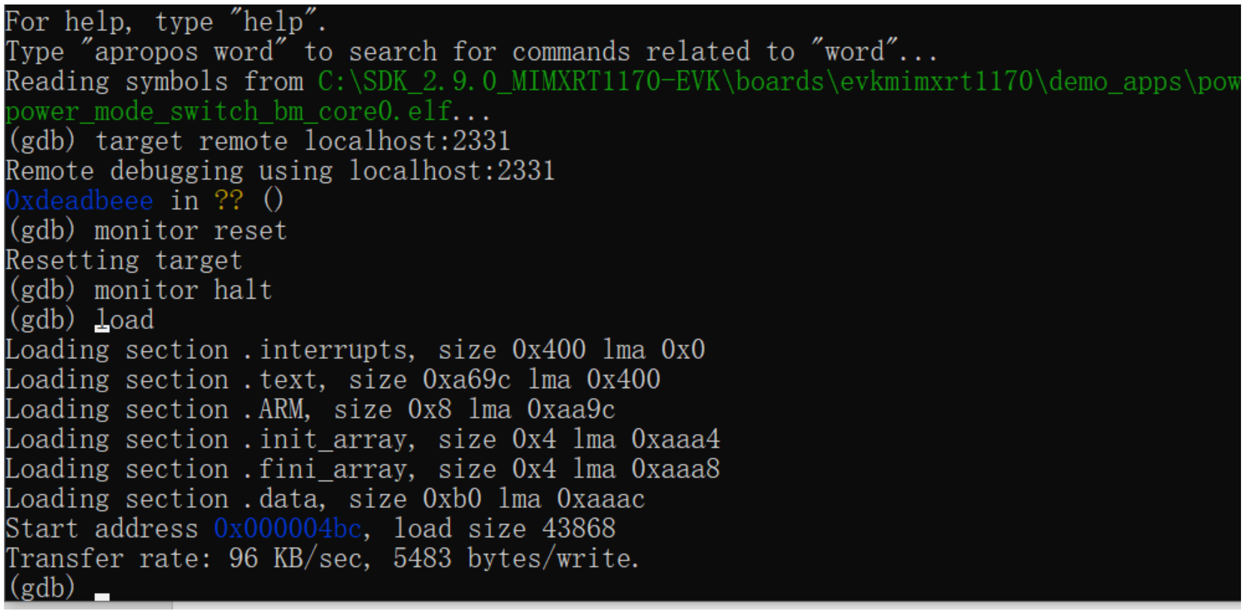
10. 运行以下命令：

    a. `Target remote localhost:2331`

    b. `Monitor reset`

    c. `Monitor halt`

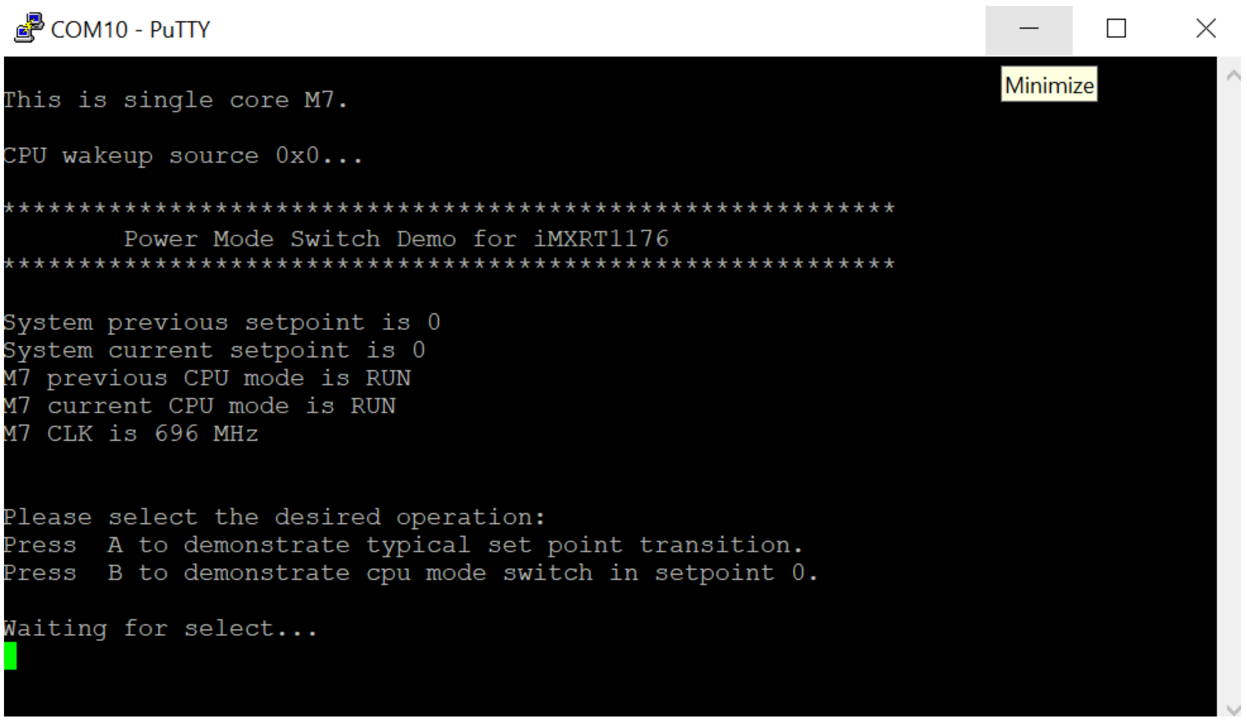
    d. `Load`



```
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
power_mode_switch_bm_core0.elf...
(gdb) target remote localhost:2331
Remote debugging using localhost:2331
0xdeadbeee in ?? ()
(gdb) monitor reset
Resetting target
(gdb) monitor halt
(gdb) load
Loading section .interrupts, size 0x400 lma 0x0
Loading section .text, size 0xa69c lma 0x400
Loading section .ARM, size 0x8 lma 0xaa9c
Loading section .init_array, size 0x4 lma 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。



```
This is single core M7.

CPU wakeup source 0x0...

***********************************************************
        Power Mode Switch Demo for iMXRT1176
***********************************************************

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 30. 程序输出

Date of release: February 2, 2021

Document identifier: AN13116