

Sri Vidya Gunturi, Sanjeev Karaiyan

## 摘要

AM263x Sitara™ Arm® 微控制器旨在满足下一代工业和汽车嵌入式产品复杂的实时处理需求。AM263x MCU 系列包含多个具有多达四个 400MHz Arm Cortex®-R5F 内核的引脚对引脚兼容器件。该系列设计用于先进的电机控制和带有先进模拟模块的数字电源控制应用。本文档根据器件型号详细介绍了这些系列器件之间的差异，并与超集器件 AM2634 进行了比较。

## 内容

1 引言.....	2
2 器件命名规则.....	3
3 AM263x 中的 R5 内核和 TCM.....	4
3.1 Am263x 中的 R5 内核命名规则.....	4
4 AM263x 系列的示例支持.....	5
5 双核器件 (AM2632) 的 IPC 示例支持.....	7
5.1 选项 1 (使用 MulticoreImageGen.js).....	7
5.2 选项 2 (四核系统工程到双核系统工程迁移指南).....	8
6 双核器件 (AM2632) 的系统工程示例支持.....	12
7 CCS 中的目标配置.....	13
7.1 先决条件.....	13
7.2 创建一个目标配置.....	13
8 连接到目标核心.....	16
9 LaunchPad 和 ControlCARD 的硬件说明.....	17
9.1 标准模拟器件的 LaunchPad 引脚排列.....	19
9.2 标准模拟器件 LaunchPad 中的 ADC 和 DAC 映射.....	19
9.3 引脚多路复用映射 - 标准模拟器件 - LaunchPad.....	20
9.4 标准模拟器件 ControlCARD 中的 ADC 和 DAC 映射.....	25
10 总结.....	26
11 参考文献.....	27

## 商标

Sitara™ and LaunchPad™ are trademarks of Texas Instruments.

Arm® and Cortex® are registered trademarks of Arm Ltd.

所有商标均为其各自所有者的财产。

## 1 引言

适用于 AM263x 的当前 MCU\_PLUS\_SDK 旨在支持整个系列的 AM263x 器件。本文档介绍了为 AM263x 的不同子集器件而设计的 SDK 示例，并介绍了修改现有示例以支持子集器件 AM2632 和 AM2631 的步骤。

AM263x LaunchPad™ 开发套件和 controlCARD 专为超集器件 AM2634 而开发。使用其他 AM263x SOC 的用户可以开始在 AM2634 EVM 上使用，因为这些 SOC 与 AM2632 和 AM2631 引脚对引脚兼容。本文档™ 说明了如何调整这些 EVM 以适用于其他系列的器件。使用不同 AM263x 系列器件的用户可将相同的 SDK 库和 EVM 无缝用于其不同的 SOC 工程。

## 2 器件命名规则

任何 AM263x IP 的特性和特征均由器件型号标识。表 2-1 介绍了如何解密功能。有关速度、存储器、控制 IP 数量、网络设置等的详细信息，请参阅数据表。表 2-2 根据表 2-1 中说明的零件号进行分类。

表 2-1. 基于器件型号的器件命名规则

AM263	4	C	O	K	F	H	M	ZCZ	R	Q1
通用 IP 名称	R5 内核数	器件修订版本	器件运行性能点	特性	功能安全	安全性	结温	封装标识符	包装	汽车标识符

表 2-2 指定了六个器件型号之间的差异。有关这些差异的更多信息，请参阅 [AM263x Sitara™ 微控制器](#) 数据表的“器件比较”部分。

表 2-2. 六个 AM263x 器件型号的器件比较

器件型号	AM2634	AM2632	AM2631	AM2634-Q1	AM2632-Q1	AM2631-Q1
	AM2634COMFHA ZCZR	AM2632COLFHA ZCZR	AM2631CNDGHA ZCZR	AM2634COKFHM ZCZRQ1	AM2632COKFHM ZCZRQ1	AM2631CODGHM ZCZRQ1
内核特征						
R5 内核	4	2	1	4	2	1
时钟速度	400MHz	400MHz	400MHz	400MHz	400MHz	400MHz
存储器	2MB	2MB	1MB	2MB	2MB	2MB
安全与安防						
功能安全	是	是	否	是	是	否
安全性	是	是	是	是	是	是
汽车规范	否	否	否	是	是	是
温度	-40 至 105°C	-40 至 105°C	-40 至 105°C	-40°C 至 150°C	-40°C 至 150°C	-40°C 至 150°C
控制系统实例						
模拟	增强模拟器件	增强模拟器件	标准模拟器件	增强模拟器件	增强模拟器件	标准模拟器件
ADC	5	5	3	5	5	3
PWM	32	32	16	32	32	16
QEP	3	3	2	3	3	2
SDFM	2	2	1	2	2	1
CMP	20	20	12	20	20	12
网络协议						
特性	M	L	D	K	K	D
Bosch CAN-FD	是	是	是	是	是	是
EtherCAT	是	是	否	否	否	否
Kunbus 堆栈 (集成堆栈)	是	否	否	否	否	否
ICSS-PRU	是	是	是	是	是	是
一般特性						
封装	15x15	15x15	15x15	15x15	15x15	15x15
汽车规范	否	否	否	是	是	是

### 3 AM263x 中的 R5 内核和 TCM

在 AM263x 中，多个 R5F 内核排列在集群中，具有 256KB 的共享紧密耦合内存 (TCM)，如图 3-1 所示。

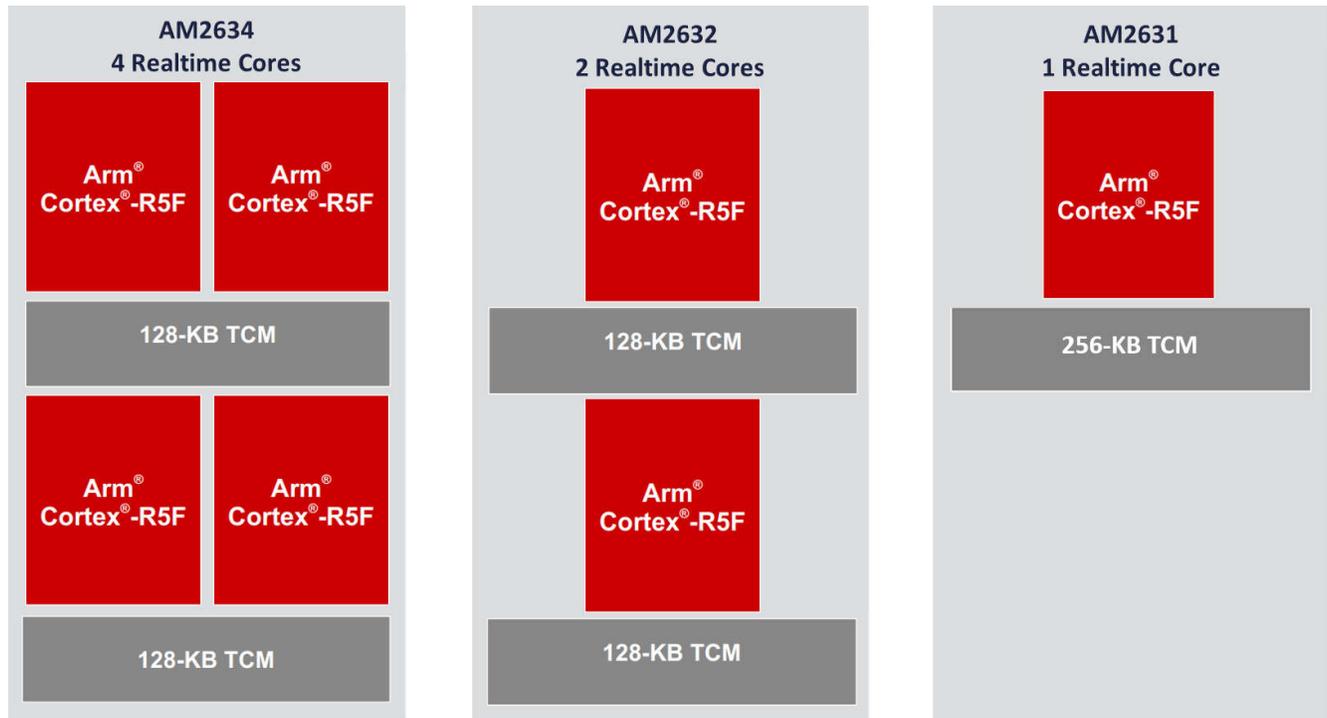


图 3-1. AM263x 系列中的 R5 内核和 TCM

#### 3.1 Am263x 中的 R5 内核命名规则

在本文档的后面，根据内核在集群中的位置，R5 内核的定义如下所述：

AM2634 有两个 R5 集群：

- 第一个集群称为 R50。
  - 第一个集群中的第一个内核称为内核 R50-0。
  - 第一个集群中的第二个内核称为内核 R50-1。
- 第二个集群称为 R51。
  - 第二个集群中的第一个内核称为 R51-0。
  - 第二个集群中的第二个内核称为 R51-1。

AM2632 有两个 R5 集群：

- 第一个集群称为 R50。
  - 第一个集群中的第一个内核称为内核 R50-0。
- 第二个集群称为 R51。
  - 第二个集群中的第一个内核称为 R51-0。

Am2631 有一个 R5 集群：

- 单个集群称为 R50。
  - 此集群中的第一个内核称为内核 R50-0

## 4 AM263x 系列的示例支持

AM2634 器件支持 SDK 中的所有示例。目前，某些器件不支持 IPC 和网络。节 2 中器件型号的第 8 个字符代表器件的**运行性能点**。同样，第 6 个字符表示器件中的**R5 内核数**，请参阅节 2 了解更多信息。例如，在 AM2632COLFHAZCZR 中，O 是速度和存储器设置，2 代表两个 R5 内核。节 4 根据存储器设置和内核数，列出了适用于 AM263x 系列器件的 MCU PLUS SDK 中支持的示例。

**表 4-1. 适用于 AM263x 系列的 MCU\_PLUS\_SDK 中的示例**

SDK 中的示例	子模块	存储器分级		基于内核数的分级		
		K、O、P	N	4 个内核	2 个内核	1 个内核
示例工程		支持	支持	支持	支持	支持
软件诊断库 (SDL)		支持	支持	支持	支持	支持
空工程 (多核示例)		支持	支持	支持	支持，另请参阅节 6	仅支持 r5ss0_0 示例。 不支持多核示例。
Hello World 工程		支持	支持	支持	支持	支持
Hello World C++ 工程		支持	支持	支持	支持	支持
OS 内核和驱动程序移植层 (DPL)		支持	支持	支持	支持	支持
SOC 和电路板外设驱动程序	控制 IP 示例	支持	支持	支持	支持	支持
	GPIO	支持	支持	支持	支持	支持
	IPC (多核示例)	支持	支持	支持	支持，另请参阅节 5	不支持。
	HSM	支持	支持	支持	支持	支持
	串行通信示例	支持	支持	支持	支持	支持
	EDMA	支持	支持	支持	支持	支持
	引导	支持	支持	支持	支持	支持
	看门狗	支持	支持	支持	支持	支持
次级引导加载程序 (SBL)		支持	支持	支持	支持	支持
实时调试		支持	支持	支持	支持	支持
工业通信工具套件		支持	支持	支持	支持	支持

表 4-1. 适用于 AM263x 系列的 MCU\_PLUS\_SDK 中的示例 (continued)

SDK 中的示例	子模块	存储器分级		基于内核数的分级		
		K、O、P	N	4 个内核	2 个内核	1 个内核
网络	Enet CPSW EST 示例	支持	支持	支持	支持	支持
	Enet 第 2 层 CPSW 示例	支持	支持	支持	支持	支持
	Enet 第 2 层 CPSW 开关示例	支持	支持	支持	支持	支持
	Enet 第 2 层多通道示例	支持	支持	支持	支持	支持
	Enet 环回示例	支持	支持	支持	支持	支持
	Enet Lwip CPSW 原始 HTTP 服务器	支持	支持	支持	支持	支持
	Enet Lwip 套接字示例	支持	支持	支持	支持	支持
	Enet Tx 散聚示例	支持	不支持	支持	支持, 如果存储器设置为 N, 则不支持	支持, 如果存储器设置为 N, 则不支持
	Enet Lwip TCP 客户端示例	支持	不支持	支持	支持, 如果存储器设置为 N, 则不支持	支持, 如果存储器设置为 N, 则不支持
	Enet Lwip TCP 服务器示例	支持	不支持	支持	支持, 如果存储器设置为 N, 则不支持	支持, 如果存储器设置为 N, 则不支持
	Enet Lwip UDP IGMP 服务器示例	支持	不支持	支持	支持, 如果存储器设置为 N, 则不支持	支持, 如果存储器设置为 N, 则不支持
	Enet Lwip UDP 服务器示例	支持	不支持	支持	支持, 如果存储器设置为 N, 则不支持	支持, 如果存储器设置为 N, 则不支持
	Enet Lwip CPSW 示例	支持	不支持	支持	支持, 如果存储器设置为 N, 则不支持	支持, 如果存储器设置为 N, 则不支持
Enet CPSW Timesync PTP 演示	支持	不支持	支持	支持, 如果存储器设置为 N, 则不支持	支持, 如果存储器设置为 N, 则不支持	
Enet CPSW 工作模式演示	支持	不支持	支持	支持, 如果存储器设置为 N, 则不支持	支持, 如果存储器设置为 N, 则不支持	
MATHLIB 基准测试		支持	支持	支持	支持	支持
安全		支持	支持	支持	支持	支持
软件诊断库 (SDL)		支持	支持	支持	支持	支持

## 5 双核器件 (AM2632) 的 IPC 示例支持

IPC Notify 示例中的更改，以通过 CCS 构建来创建 AM2632 的双核 .appimage :

本节包含在 IPC Notify 示例中所做的更改，以根据现有四核 IPC Notify 示例创建双锁步内核 (两个内核) 示例。

### 5.1 选项 1 (使用 MulticoreImageGen.js)

本节提供了一种从现有四核 IPC Notify 示例创建双锁步内核 (两个内核) 示例的简单方法。此示例不涉及任何系统工程，因此，用户必须使用命令行来创建最终的组合应用映像。下面介绍了创建双核应用映像的步骤。

- 按如下所示从 SDK 导入 R50-0 和 R51-0 的 IPC Notify 示例 (另请参阅节 3.1)。忽略系统工程。

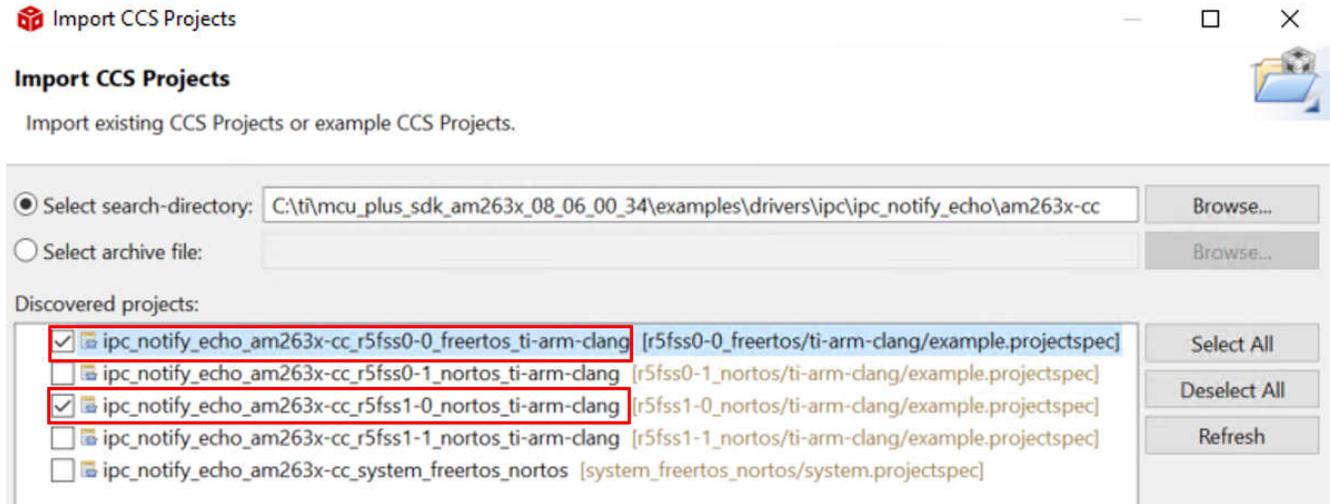


图 5-1. R50-0 和 R51-0 的 IPC Notify 示例

- 修改 example.syscfg 以支持对内核 0-0 和 1-0 的 IPC Notify 功能。然后，禁用内核 1-0 和 1-1 的 IPC Notify 功能。



图 5-2. IPC Notify R50\_0 Syscfg

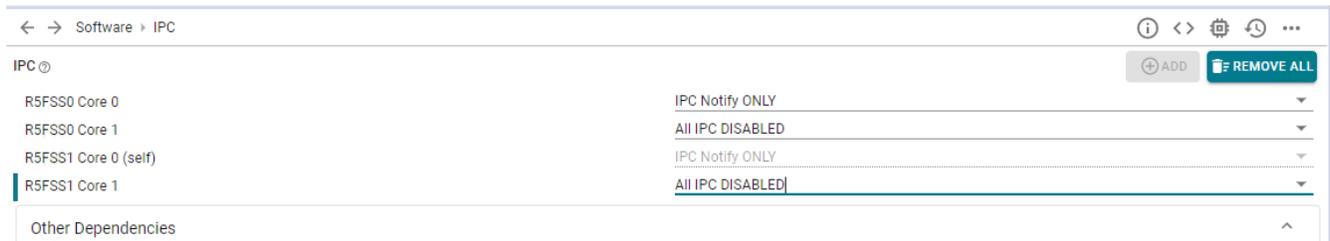


图 5-3. IPC Notify R51\_0 Syscfg

- 按如下所示更改远程内核 ID 列表以仅支持内核 1-0，并删除其余内核。在 `ipc_notify_echo_am263x-cc_r5fss0-0_nortos_ti-arm-clang` 和 `ipc_notify_echo_am263x-cc_r5fss1-0_nortos_ti-arm-clang` 工程的 `ipc_notify_echo.c` 文件中进行此更改。

```
uint32_t gRemoteCoreId[] = {
    CSL_CORE_ID_R5FSS1_0,
    CSL_CORE_ID_MAX
};
```

- 单独构建这些工程。构建后，`.rprc` 文件可用于这两个工程。
- 使用以下命令组合这些 `.rprc` 以形成一个双核 `.appimage`：

```
C:\ti\{sysconfig}\nodejs\node C:\ti\{mcu_plus_sdk}\tools\boot\multicoreImageGen/
multicoreImageGen.js --devID 55 --out Debug/Combined.debug.appimage ./ipc_notify_echo_am263x-
cc_r5fss0-0_freertos_ti-arm-clang/Debug/ipc_notify_echo_am263x-cc_r5fss0-0_freertos_ti-arm-
clang.rprc@0 ./ipc_notify_echo_am263x-cc_r5fss1-0_nortos_ti-arm-clang/Debug/ipc_notify_echo_am263x-
cc_r5fss1-0_nortos_ti-arm-clang.rprc@2
```

MulticoreImage Generation 的命令格式如下所示。

```
cd ${SDK_INSTALL_PATH}/tools/boot/multicoreImageGen${NODE} multicoreImageGen.js --devID {DEV_ID} --
out {Output image file (.appimage)} {core 1 rprc file}@{core 1 id} [ {core n rprc file}@{core n
id} ... ]
```

## 5.2 选项 2 ( 四核系统工程到双核系统工程迁移指南 )

此过程有助于将现有 IPC Notify 示例导入为双核系统工程，其中 R50\_0 作为主内核，R51\_0 作为远程内核。需要在用于 IPC Notify 的 SDK 中的 `makefile` 中完成以下更改。根据器件类型 CC 或 LP 选择文件夹。

打开 `examples\drivers\ipc\ipc_notify_echo\am263x-cc\system_freertos_nortos\makefile` 处的 `system_freertos_nortos` `makefile`。

- 保留 `CORE_0` 定义。从第 18 和 19 行中移除 `CORE_1` 和 `CORE_2` 定义，因为这些定义不是必需的。
- 按如下所示定义 `CORE_1`：

```
CORE_1=--script ../r5fss1-0_nortos/example.syscfg --context r5fss1-0 --output ../r5fss1-0_nort
os/ti-arm-clang/generated
```

- 从 `CORES` 中移除 `$(CORE_3)` 和 `$(CORE_2)`。按如下所示定义 `CORES`。

```
CORES = \
$(CORE_1) \
$(CORE_0) \
```

- 从 `all: syscfg` 部分删除 `r5fss0-1` 和 `r5fss1-1` 实例。

```
all:
syscfg $(MAKE) -C ../r5fss0-0_freertos/ti-arm-clang/ all
$(MAKE) -C ../r5fss1-0_nortos/ti-arm-clang/ all
$(MAKE) $(MULTI_CORE_BOOTIMAGE_NAME)
```

- 从 `clean` 部分删除 `r5fss0-1` 和 `r5fss1-1` 实例。

```
clean:
$(MAKE) -C ../r5fss0-0_freertos/ti-arm-clang/ clean
$(MAKE) -C ../r5fss1-0_nortos/ti-arm-clang/ clean
$(RM) $(MULTI_CORE_BOOTIMAGE_NAME)
$(RM) $(MULTI_CORE_BOOTIMAGE_NAME_SIGNED)
$(RM) $(MULTI_CORE_BOOTIMAGE_NAME_XIP)
```

- 从 scrub 部分删除 r5fss0-1 和 r5fss1-1 实例。

```
scrub:
$(MAKE) -C ../r5fss0-0_freertos/ti-arm-clang/ scrub
$(MAKE) -C ../r5fss1-0_nortos/ti-arm-clang/ scrub
```

- 从 MULTI\_CORE\_APP\_PARAMS 部分删除 r5fss0-1 和 r5fss1-1 实例。

```
MULTI_CORE_APP_PARAMS = \
../r5fss0-0_freertos/ti-arm-clang/ipc_notify_echo.$(PROFILE).rprc@$(BOOTIMAGE_CORE_ID_r5fss0-0) \
../r5fss1-0_nortos/ti-arm-clang/ipc_notify_echo.$(PROFILE).rprc@$(BOOTIMAGE_CORE_ID_r5fss1-0) \
```

- 从 MULTI\_CORE\_APP\_PARAMS\_XIP 部分删除 r5fss0-1 和 r5fss1-1 实例。

```
MULTI_CORE_APP_PARAMS_XIP = \
../r5fss0-0_freertos/ti-arm-clang/ipc_notify_echo.$(PROFILE).rprc_xip@$(
BOOTIMAGE_CORE_ID_r5fss0-0) \
../r5fss1-0_nortos/ti-arm-clang/ipc_notify_echo.$(PROFILE).rprc_xip@$(BOOTIMAGE_CORE_ID_r5fss1-0) \
```

- 从 MULTI\_CORE\_BOOTIMAGE\_DEPENDENCY 部分删除 r5fss0-1 和 r5fss1-1 实例。

```
MULTI_CORE_BOOTIMAGE_DEPENDENCY = \
../r5fss0-0_freertos/ti-arm-clang/ipc_notify_echo.$(PROFILE).rprc \
../r5fss1-0_nortos/ti-arm-clang/ipc_notify_echo.$(PROFILE).rprc \
```

打开以下位置的 projectspec makefile : examples\drivers\ipc\ipc\_notify\_echo\am263x-cc\system\_freertos\_nortos\ makefile\_projectspec。

在此 makefile 文件中进行以下更改 :

- 从 clean 部分删除 r5fss0-1 和 r5fss1-1 实例 , 以便仅支持 r5fss0-0 和 r5fss1-0。

```
clean:
$(CCS_ECLIPSE) -noSplash -data $(MCU_PLUS_SDK_PATH)/ccs_projects -application
com.ti.ccstudio.apps.projectBuild -ccs.projects
$(PROJECT_NAME) -ccs.configuration
$(PROFILE) -ccs.clean
$(MAKE) -C ../r5fss0-0_freertos/ti-arm-clang/ -f makefile_projectspec clean
$(MAKE) -C ../r5fss1-0_nortos/ti-arm-clang/ -f makefile_projectspec clean
```

打开以下位置的 bootimage\_gen makefile : examples\drivers\ipc\ipc\_notify\_echo\am263x-cc\system\_freertos\_nortos\ makefile\_system\_ccs\_bootimage\_gen。

在此 makefile 文件中进行以下更改 :

- 从 MULTI\_CORE\_APP\_PARAMS 部分删除 r5fss0-1 和 r5fss1-1 实例。

```
MULTI_CORE_APP_PARAMS = \
../ipc_notify_echo_am263x-cc_r5fss0-0_freertos_ti-arm-clang/$(PROFILE)/ipc_notify_echo_am263x-
cc_r5fss0-0_freertos_ti-arm-clang.rprc@$(BOOTIMAGE_CORE_ID_r5fss0-0) \
../ipc_notify_echo_am263x-cc_r5fss1-0_nortos_ti-arm-clang/$(PROFILE)/ipc_notify_echo_am263x-
cc_r5fss1-0_nortos_ti-arm-clang.rprc@$(BOOTIMAGE_CORE_ID_r5fss1-0) \
```

- 从 MULTI\_CORE\_APP\_PARAMS\_XIP 部分删除 r5fss0-1 和 r5fss1-1 实例。

```
MULTI_CORE_APP_PARAMS_XIP = \
../ipc_notify_echo_am263x-cc_r5fss0-0_freertos_ti-arm-clang/$(PROFILE)/ipc_notify_echo_am263x-
cc_r5fss0-0_freertos_ti-arm-clang.rprc_xip@$(BOOTIMAGE_CORE_ID_r5fss0-0) \
../ipc_notify_echo_am263x-cc_r5fss1-0_nortos_ti-arm-clang/$(PROFILE)/ipc_notify_echo_am263x-
cc_r5fss1-0_nortos_ti-arm-clang.rprc_xip@$(BOOTIMAGE_CORE_ID_r5fss1-0) \
```

打开以下位置的 system projectspec 文件 : examples\drivers\ipc\ipc\_notify\_echo\am263x-cc\system\_freertos\_nortos\ system.projectspec。

在此 projectspec 文件中进行以下更改：

- 删除 r5fss0-1 和 r5fss1-1 projectspec 文件，使其不再随 system\_freertos\_nortos 工程一起导入（第 4 行和第 6 行）。

```
<projectSpec>
<import spec="../r5fss0-0_freertos/ti-arm-clang/example.projectspec"/>
<import spec="../r5fss1-0_nortos/ti-arm-clang/example.projectspec"/>
```

打开以下位置的系统 projectspec 文件：examples\drivers\ipc\ipc\_notify\_echo\am263x-cc\system\_freertos\_nortos\system.xml。

在此 projectspec 文件中进行以下更改：

- 从 xml 文件中删除 r5fss0-1 和 r5fss1-1 内核工程配置，并仅保留 0-0 和 1-0 内核。按如下所示将内核 1 修改为 r5fss1-0 内核。

```
<project configuration="@match" id="project_0" name="ipc_notify_echo_am263x-cc_r5fss0-0_freertos_ti-arm-clang">
</project>
<core id="MAIN_PULSAR_Cortex_R5_0_0" project="project_0"/>
<project configuration="@match" id="project_1" name="ipc_notify_echo_am263x-cc_r5fss1-0_nortos_ti-arm-clang">
</project>
<core id="MAIN_PULSAR_Cortex_R5_1_0" project="project_1"/>
```

修改系统工程 makefile 后，将此示例导入 CCS。此步骤与其他系统工程相同。

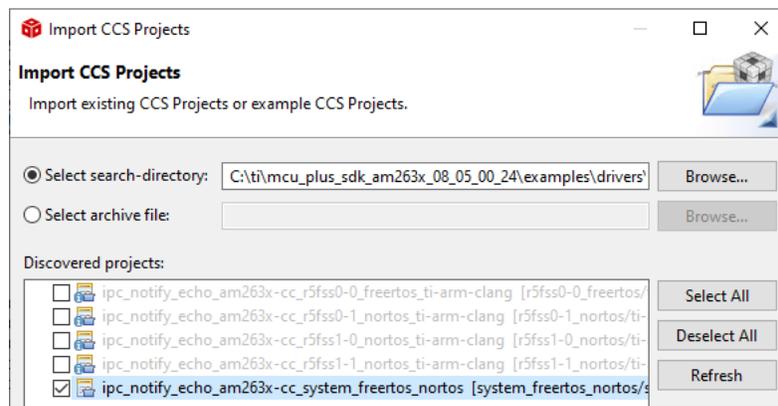


图 5-4. 导入 IPC Notify 的双核系统工程

修改 example.syscfg 以支持对内核 0-0 和 1-0 的 IPC Notify 功能。然后，禁用内核 1-0 和 1-1 的 IPC Notify 功能。

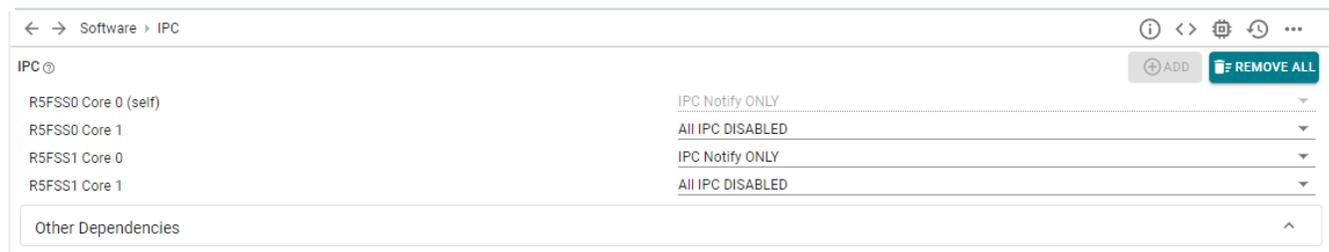


图 5-5. IPC Notify R50\_0 Syscfg



图 5-6. IPC Notify R50\_0 Syscfg

按如下所示更改远程内核 ID 列表以仅支持内核 1-0，并删除其余内核。在 `ipc_notify_echo_am263x-cc_r5fss0-0_nortos_ti-arm-clang` 和 `ipc_notify_echo_am263x-cc_r5fss1-0_nortos_ti-arm-clang` 工程的 `ipc_notify_echo.c` 文件中进行此更改。

```
uint32_t gRemoteCoreId[] = {
    CSL_CORE_ID_R5FSS1_0,
    CSL_CORE_ID_MAX
};
```

现在编译系统工程来为内核 0-0 和 1-0 生成组合的应用映像。此应用映像可以刷写到 AM2632 器件上。

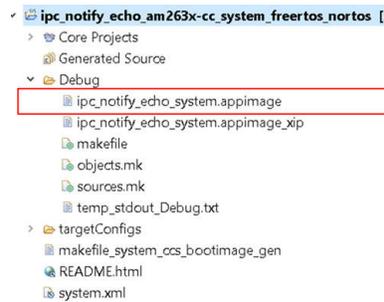


图 5-7. 为双核工程生成应用映像

#### 备注

AM2631 是单核器件，单核器件不支持 IPC 示例。

## 6 双核器件 (AM2632) 的系统工程示例支持

在 MCU-PLUS-SDK\_AM263x 中，默认为双核器件配置系统工程。但内核 R50\_0 和 R50\_1 有相关示例。可以修改这些示例以启用内核 R50\_0 和 R51\_0 的系统工程。

若要查找系统工程，请转至 → C:\ti\mcu\_plus\_sdk\_am263x\_08\_05\_00\_24\examples\empty。在这里，SDK 有 2 个系统工程：freeRtos 系统工程和 Nortos 系统工程。工程师可为应用选择系统工程。

需要修改以下文件：

1. examples\empty\am263x-cc\system\_nortos\makefile
2. examples\empty\am263x-cc\system\_nortos\makefile\_projectspec
3. examples\empty\am263x-cc\system\_nortos\makefile\_system\_ccs\_bootimage\_gen
4. examples\empty\am263x-cc\system\_nortos\system.projectspec
5. examples\empty\am263x-cc\system\_nortos\system.xml

在这些文件中，将 r5fss0-1 内核替换为 r5fss1-0。节 5 介绍了这些更改。

---

### 备注

AM2631 是单核器件，单核器件不支持多核系统工程。

---

## 7 CCS 中的目标配置

### 7.1 先决条件

- 确保已按照 [下载、安装和设置 CCS](#) 中所述安装 CCS。
- 确保按照 [设置 UART 终端](#) 中所述标识用于控制台的 UART 端口。
- 确保已按照 [电缆连接](#) 中所示连接 EVM 电源线、JTAG 电缆和 UART 电缆。
- 确保已完成 SOC 初始化方法的步骤。
  - 推荐的方法是 [使用 QSPI 存储器中刷写的二进制文件进行 SOC 初始化](#)。
  - 如果不能使用推荐的方法，[SOC 初始化](#) 中提到了其他选项。
- 根据 SOC 初始化方法正确设置某些 EVM 引导模式开关。
  - 对于 [推荐](#) 的方法，[使用 QSPI 存储器中刷写的二进制文件进行 SOC 初始化](#)，引导模式为 [QSPI 引导模式](#)。
- 确保在进行 EVM 上电时 UART 或 CCS 控制台日志指示 SOC 初始化成功。
- 确保已按照 [构建 Hello World 示例](#) 中所述构建了相关示例。

### 7.2 创建一个目标配置

- 转到 `View > Target Configuration`。

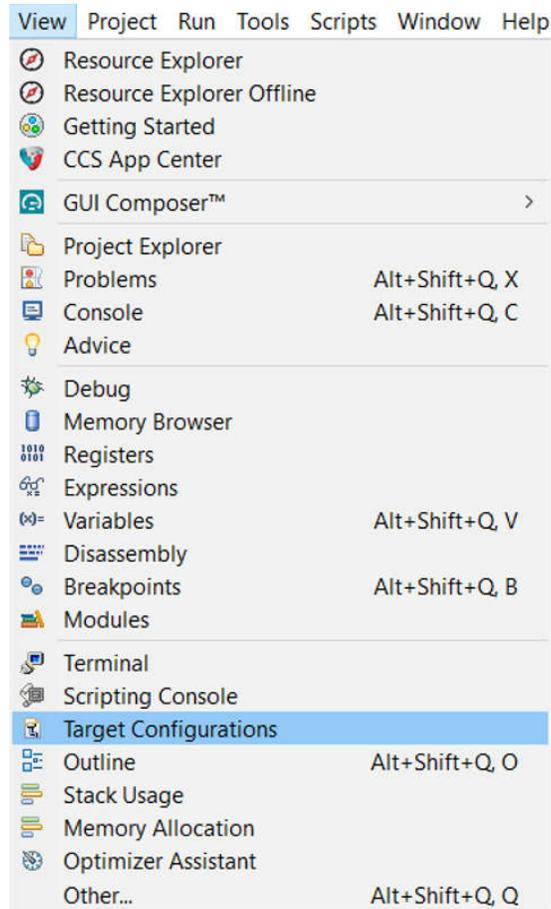


图 7-1. Target Configurations 视图

- 创建新的目标配置。

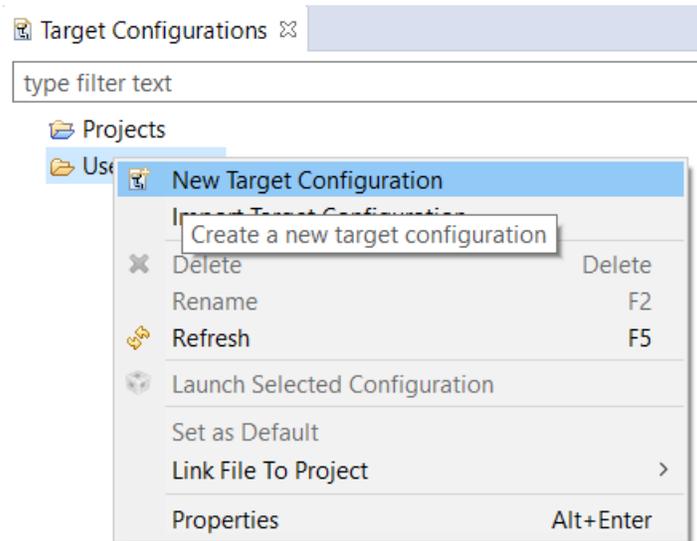


图 7-2. 新目标配置

- 为新目标配置命名，通常为 {soc name}\_{JTAG type}。

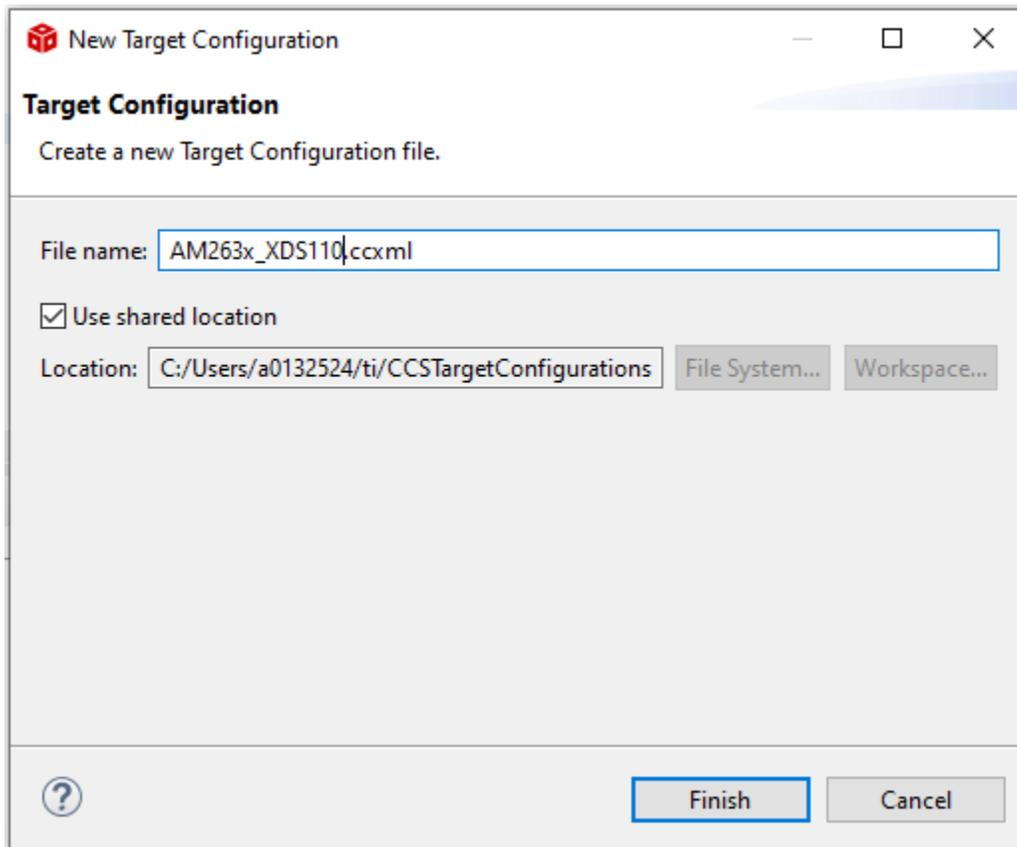


图 7-3. 新建目标配置窗口

- 选择 XDS110 USB Debug Probe 作为连接。

## Basic

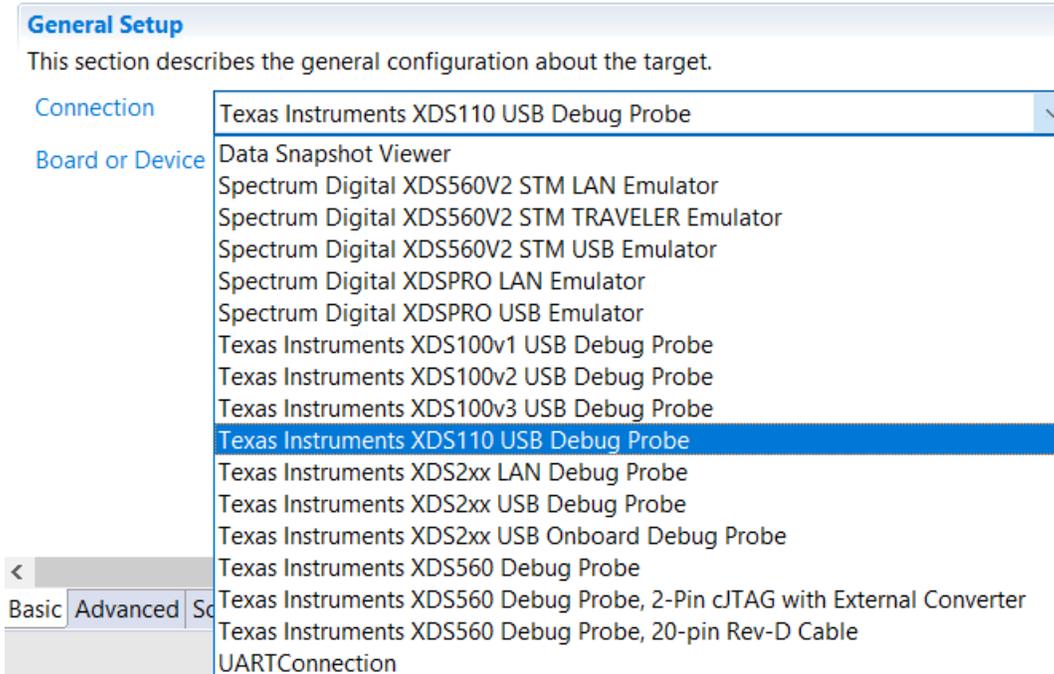


图 7-4. 配置器配置

- 在 *Board or Device Setting* 中，根据器件选择 AM2632 或 AM2631。

### Basic

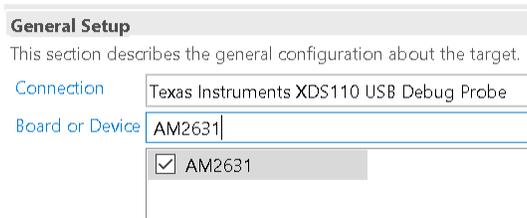


图 7-5. 选择器件或器件型号

- 在 *Advanced* 选项卡下，确保已配置所有设置。

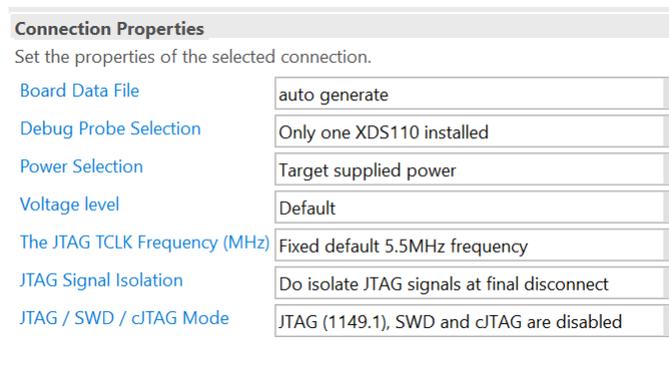


图 7-6. JTAG 频率配置

- 在 *Advanced* 选项卡下，选择 `Cortex_R5_0`。
- 确保器件 `csp gel` 文件路径位于 `initialization script` 字段中。

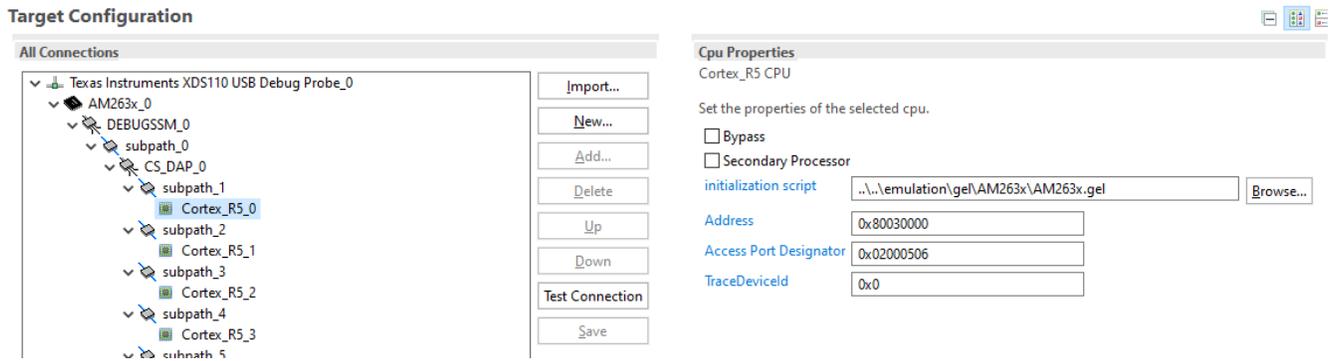


图 7-7. GEL 脚本

- 点击 **Save** 以保存新创建的目标配置。
- 现在，完成 [EVM 设置](#)，为运行程序准备 EVM。

## 8 连接到目标核心

在连接到内核以刷写映像时，您可能会遇到所有 AM263x 器件支持的多个内核。请仅根据您的器件选择以下内核。

表 8-1. 可用目标内核

器件	工程	可连接到的内核
AM2631	单核工程	Cortex_R5_0_0
AM2632	单核工程	Cortex_R5_0_0 或 Cortex_R5_1_0
	双锁步内核工程	Cortex_R5_0_0 Cortex_R5_1_0
AM2634	任何组合工程都是可能的	Cortex_R5_0_0 Cortex_R5_0_1 Cortex_R5_1_0 Cortex_R5_1_1

### 备注

子集器件中的锁步和双核配置处理：AM2632 和 AM2631 的各个集群配置为锁步模式。这些器件中不存在集群的双核模式。

## 9 LaunchPad 和 ControlCARD 的硬件说明

LaunchPad 和 controlCARD 仅使用超集器件 (AM2634) 构建。然后，客户可以通过购买满足其系统要求的精选 OPN 器件 (AM2642 或 AM2631) 来缩减规模。

用于增强模拟器件的 AM2634、AM2632 和 AM2631 LaunchPad 和 ControlCARD 具有相似的硬件说明和引脚多路复用器。这些硬件详细信息和引脚映射在 [AM263x LaunchPad 用户指南](#) 和 [ControlCARD 用户指南](#) 中列出。请参阅 [节 2](#) 器件型号中的 *特性字符* (第 9 个字符)。表 9-1 展示了标准模拟器件和增强模拟器件之间的差异。唯一的差异是集成的控制外设数。

**表 9-1. 增强与标准模拟器件**

模拟	增强模拟器件		标准模拟器件	
特性说明	J、K、L、M		C、D、E、F	
	编号	实例	编号	实例
ADC	5	ADC0 ADC1 ADC2 ADC3 ADC4	3	ADC0 ADC1 ADC2 ADC3
PWM	32	EPWM0 EPWM1 EPWM2 EPWM3 EPWM4 EPWM5 EPWM6 EPWM7 EPWM8 EPWM9 EPWM10 EPWM11 EPWM12 EPWM13 EPWM14 EPWM15 EPWM16 EPWM17 EPWM18 EPWM19 EPWM20 EPWM21 EPWM22 EPWM23 EPWM24 EPWM25 EPWM26 EPWM27 EPWM28 EPWM29 EPWM30 EPWM31	16	EPWM0 EPWM1 EPWM2 EPWM3 EPWM4 EPWM5 EPWM6 EPWM7 EPWM8 EPWM9 EPWM10 EPWM11 EPWM12 EPWM13 EPWM14 EPWM15
QEP	3	EQEP0 EQEP1 EQEP2	2	EQEP0 EQEP1

**表 9-1. 增强与标准模拟器件 (continued)**

模拟	增强模拟器件		标准模拟器件	
特性说明	J、K、L、M		C、D、E、F	
	编号	实例	编号	实例
SDFM	2	SDFM0 SDFM1	1	SDFM0
CMPSS	20	CMPSSA0:inH CMPSSA0:inL CMPSSA1:inH CMPSSA1:inL CMPSSB0:inH/inL CMPSSB1:inH/inL CMPSSA2:inH CMPSSA2:inL CMPSSA3:inH CMPSSA3:inL CMPSSB2:inH/inL CMPSSB3:inH/inL CMPSSA4:inH CMPSSA4:inL CMPSSA5:inH CMPSSA5:inL CMPSSB4:inH/inL CMPSSB5:inH/inL CMPSSA6:inH CMPSSA6:inL CMPSSA7:inH CMPSSA7:inL CMPSSB6:inH/inL CMPSSB7:inH/inL CMPSSA8:inH CMPSSA8:inL CMPSSA9:inH CMPSSA9:inL CMPSSB8:inH/inL CMPSSB9:inH/inL	12	CMPSSA0:inH CMPSSA0:inL CMPSSA1:inH CMPSSA1:inL CMPSSB0:inH/inL CMPSSB1:inH/inL CMPSSA2:inH CMPSSA2:inL CMPSSA3:inH CMPSSA3:inL CMPSSB2:inH/inL CMPSSB3:inH/inL CMPSSA4:inH CMPSSA4:inL CMPSSA5:inH CMPSSA5:inL CMPSSB4:inH/inL CMPSSB5:inH/inL

## 9.1 标准模拟器件的 LaunchPad 引脚排列

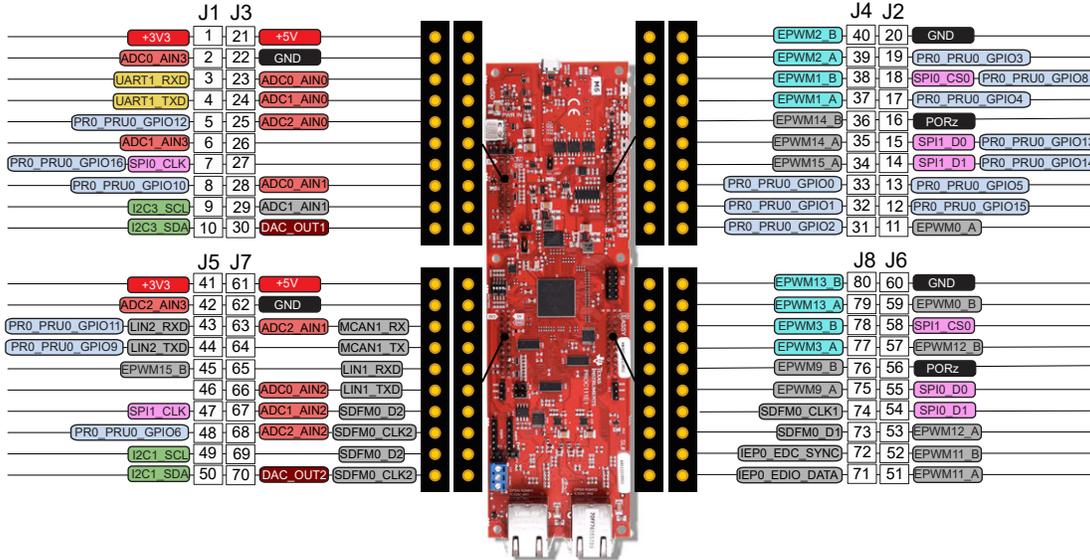


图 9-1. AM263x 标准模拟器件 LaunchPad 引脚排列

上图显示了标准模拟器件的 LaunchPad 引脚排列。在该图中，以彩色显示的信号是符合 LaunchPad BoosterPack 引脚排列标准的信号。以灰色显示的信号是 AM263x 支持的信号，但这些信号不符合 LP/BP 引脚排列标准。AM263x LaunchPad 支持两个完全独立的 BoosterPack XL 连接器。BoosterPack 站点 #1 ( J1/J3、J2/J4 ) 位于 SoC 和 Micro-B USB 连接器之间。BoosterPack 站点 #2 ( J5/J7、J6/J8 ) 位于 SoC 和 RJ45 连接器之间。每个 GPIO 都通过 GPIO 多路复用器提供多项功能。从 SoC 连接到 BoosterPack 接头的信号包括：

- 各种 ADC 输入
- DAC 输出
- UART1
- 各种 GPIO 信号
- SPI0 和 SPI1
- I2C1 和 I2C3
- 各种 EPWM 通道
- LIN1 和 LIN2
- MCAN1
- SDFM0

## 9.2 标准模拟器件 LaunchPad 中的 ADC 和 DAC 映射

在标准模拟器件中，AM263x LaunchPad 将 16 个 ADC 输入映射到 BoosterPack 接头。LaunchPad 中使用的所有 ADC 输入都受到 ESD 保护。增强模拟器件中有但标准模拟器件中没有的 ADC 标有 X，如图 9-2 所示。

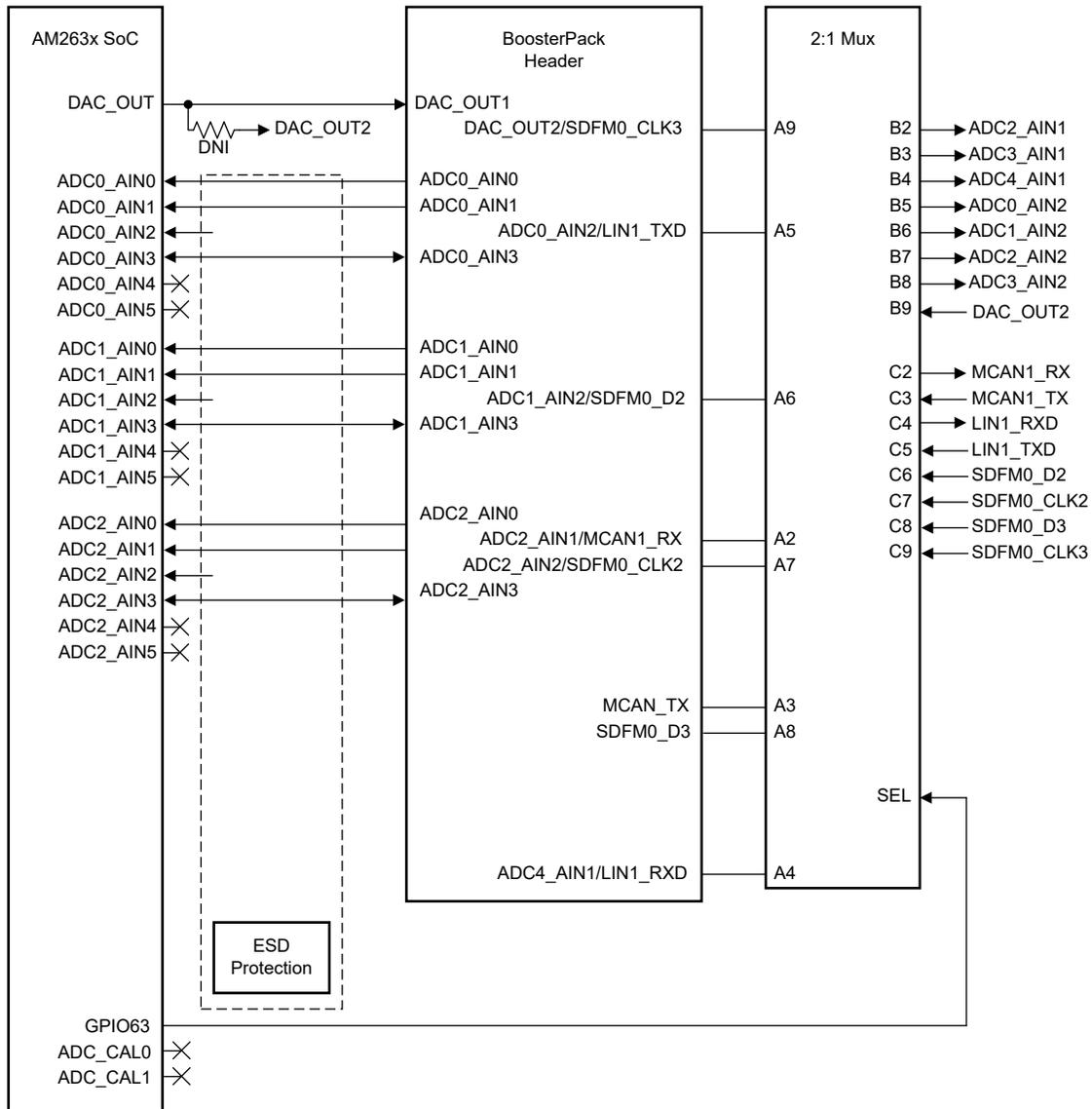


图 9-2. 标准模拟器件的 ADC 和 DAC 映射

### 9.3 引脚多路复用映射 - 标准模拟器件 - LaunchPad

表 9-2 中列出了 BoosterPack 连接器引脚的各种引脚多路复用选项。

表 9-2. J1 的引脚多路复用选项

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J1.1	3V3									
J1.2	ADC0_AIN3									
J1.3	UART1_RXD	LIN1_RXD				EPWM16_A	GPMC0_A D6	GPIO75		
J1.4	UART1_TXD	LIN1_TXD				EPWM16_B	GPMC0_A D7	GPIO76		
J1.5	PR0_PRU0_GPIO12		RMII2_TX D1	RGMII2_T D1	MII2_TXD1	EPWM28_B	GPMC0_A 8	GPIO100		

**表 9-2. J1 的引脚多路复用选项 (continued)**

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J1.6	ADC1_AIN3									
J1.7	SPI0_CLK	UART3_TXD	LIN3_TXD				FSITX0_CLK	GPIO12		
	PR0_PRU0_GPIO16			RGMI2_TXC	MII2_TXCLK	EPWM27_A	GPMC0_A5	GPIO97		
J1.8	PR0_PRU0_GPIO10		RMII2_CRS_DV	PR0_UART0_RTSn	MII2_CRS	EPWM23_A	GPMC0_WAIO	GPIO89		
J1.9	EPWM8_B	UART4_RXD	I2C3_SCL				FSITX2_DATA0	GPIO60		
J1.10	EPWM8_A	UART4_TXD	I2C3_SDA				FSITX2_CLK	GPIO59		

**表 9-3. J2 的引脚多路复用选项**

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J2.11	EPWM0_A							GPIO43		
J2.12	PR0_PRU0_GPIO15		RMII2_TXEN	RGMI2_TX_CTL	MII2_TX_EN	EPWM27_B	GPMC0_A6	GPIO98		
J2.13	PR0_PRU0_GPIO5		RMII2_RX_ER		MII2_RX_ER	EPWM22_A	GPMC0_DIR	GPIO87		
J2.14	SPI0_D1						FSITX0_DATA1	GPIO14		
	PR0_PRU0_GPIO14			RGMI2_TD3	MII2_TXD3	EPWM29_B	GPMC0_A10	GPIO102		
J2.15	SPI0_D0						FSITX0_DATA0	GPIO13		
	PR0_PRU0_GPIO13			RGMI2_TD2	MII2_TXD2	EPWM29_A	GPMC0_A9	GPIO101		
J2.16	PORz									
J2.17	PR0_PRU0_GPIO4			RGMI2_RX_CTL	MII2_RXDV	EPWM24_B	GPMC0_A0	GPIO92		
J2.18	SPI0_CS0	UART3_RXD	LIN3_RXD					GPIO11		
	PR0_PRU0_GPIO8					EPWM23_B	GPMC0_WPn	GPIO90		
J2.19	PR0_PRU0_GPIO3			RGMI2_RD3	MII2_RXD3	EPWM26_B	GPMC0_A4	GPIO96		
J2.20	GND									

**表 9-4. J3 的引脚多路复用选项**

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J3.21	5V									
J3.22	GND									
J3.23	ADC0_AIN0									

**表 9-4. J3 的引脚多路复用选项 (continued)**

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J3.24	ADC1_AIN0									
J3.25	ADC2_AIN0									
J3.26	ADC3_AIN0									
J3.27	ADC4_AIN0									
J3.28	ADC0_AIN1									
J3.29	ADC1_AIN1									
J3.30	DAC_OUT									

**表 9-5. J4 的引脚多路复用选项**

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J4.31	PR0_PRU0_GPIO2			RGMI2_R D2	MII2_RXD 2	EPWM26_A	GPMC0_A 3	GPIO95		
J4.32	PR0_PRU0_GPIO1		RMII2_RX D1	RGMI2_R D1	MII2_RXD 1	EPWM25_B	GPMC0_A 2	GPIO94		
J4.33	PR0_PRU0_GPIO0		RMII2_RX D0	RGMI2_R D0	MII2_RXD 0	EPWM25_A	GPMC0_A 1	GPIO93		
J4.34	EPWM15_A	UART5_TX D	MII1_COL				GPMC0_A D4	GPIO73		
J4.35	EPWM14_A	UART1_D SRn					GPMC0_A D2	GPIO71		
J4.36	EPWM14_B		MII1_RX_E R				GPMC0_A D3	GPIO72		
J4.37	EPWM1_A							GPIO45		
J4.38	EPWM1_B							GPIO46		
J4.39	EPWM2_A							GPIO47		
J4.40	EPWM2_B							GPIO48		

**表 9-6. J5 的引脚多路复用选项**

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J5.41	3V									
J5.42	ADC2_AIN3									
J5.43	LIN2_RXD	UART2_RX D	SPI2_D0					GPIO21		
	PR0_PRU0_GPIO11		RMII2_TX D0	RGMI2_T D0	MII2_TXD0	EPWM28_A	GPMC0_A 7	GPIO99		

**表 9-6. J5 的引脚多路复用选项 (continued)**

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J5.44	LIN2_TXD	UART2_TXD	SPI2_D1					GPIO22		
	PR0_PRU0_GPIO9			PR0_UART0_CTSn	MII2_COL	EPWM22_B	GPMC0_CLK	GPIO88		
J5.45	EPWM15_B	UART5_RXD	MII1_CRS				GPMC0_A D5	GPIO74		
J5.46	ADC3_AIN3									
J5.47	SPI1_CLK	UART4_RXD	LIN4_RXD			XBAROUT2	FSIRX0_CLK	GPIO16		
J5.48	PR0_PRU0_GPIO6		RMII2_RE F_CLK	RGMII2_RXC	MII2_RXCLK	EPWM24_A	GPMC0_CSn1	GPIO91		
J5.49	I2C1_SCL		SPI3_CS0			XBAROUT7		GPIO23		
J5.50	I2C1_SDA		SPI3_CLK			XBAROUT8		GPIO24		

**表 9-7. J6 的引脚多路复用选项**

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J6.51	EPWM11_A	UART2_CTSn					GPMC0_C LKLB	GPIO65		
J6.52	EPWM11_B	UART3_RTSn					GPMC0_0En_REn	GPIO66		
J6.53	EPWM12_A	UART3_CTSn	SPI4_CS1				GPMC0_WEn	GPIO67		
J6.54	SPI1_D1	UART5_RXD				XBAROUT4	FSIRX0_D ATA1	GPIO18		
J6.55	SPI1_D0	UART5_TXD				XBAROUT3	FSIRX0_D ATA0	GPIO17		
J6.56	PORz									
J6.57	EPWM12_B	UART1_CDn					GPMC0_CSn0	GPIO68		
J6.58	SPI1_CS0	UART4_TXD	LIN4_TXD			XBAROUT1		GPIO15		
J6.59	EPWM0_B							GPIO44		
J6.60	GND									

**表 9-8. J7 的引脚多路复用选项**

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J7.61	5V									
J7.62	GND									
J7.63	ADC2_AIN1									
	MCAN1_RX	SPI4_D0						GPIO9		

**表 9-8. J7 的引脚多路复用选项 (continued)**

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J7.64	ADC3_AIN1									
	MCAN1_TX	SPI4_D1						GPIO10		
J7.65	ADC4_AIN1									
	LIN1_RXD	UART1_RXD	SPI2_CS0			XBAROUT5		GPIO19		
J7.66	ADC0_AIN2									
	LIN1_TXD	UART1_TXD	SPI2_CLK			XBAROUT6		GPIO20		
J7.67	ADC1_AIN2									
	UART5_RXD							GPIO127	SDFM0_D2	
J7.68	ADC2_AIN2									
	UART5_TXD					I2C3_SCL	GPMC0_A DVn_ALE	GPIO126	SDFM0_C LK2	
J7.69	ADC3_AIN2									
	MCAN3_RX							GPIO129	SDFM0_D 3	
J7.70	DAC_OUT									
	MCAN3_TX	UART5_RXD						GPIO128	SDFM0_C LK3	

**表 9-9. J8 的引脚多路复用选项**

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J8.71	PR0_PRU1_GPIO18		UART3_TXD	PR0_IEP0 _EDIO_DATA_IN_OUT31	TRC_CTL	XBAROUT14	GPMC0_W AIT1	GPIO120		EQEP1_B
J8.72	PR0_PRU1_GPIO19		UART3_RXD	PR0_IEP0 _EDC_SYNC_OUT0	TRC_CLK	XBAROUT13		GPIO119		EQEP1_A
J8.73	PR0_PRU1_GPIO17		UART5_CTSn	PR0_IEP0 _EDIO_DATA_IN_OUT30				GPIO125	SDFM0_D 1	
J8.74	PR0_PRU1_GPIO7	CPTS0_TSn_SYNC	UART5_RTSn	PR0_IEP0 _EDC_SYNC_OUT1		I2C3_SDA		GPIO124	SDFM0_C LK1	
J8.75	EPWM9_A						FSITX2_D ATA1	GPIO61		

**表 9-9. J8 的引脚多路复用选项 (continued)**

引脚编号	Mode0	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7	模式 8	Mode9
J8.76	EPWM9_B	UART1_RTsn					FSIRX2_C LK	GPIO62		
J8.77	EPWM3_A							GPIO49		
J8.78	EPWM3_B							GPIO50		
J8.79	EPWM13_A	UART1_RIn					GPMC0_AD0	GPIO69		
J8.80	EPWM13_B	UART1_DTRn					GPMC0_AD1	GPIO70		

## 9.4 标准模拟器件 ControlCARD 中的 ADC 和 DAC 映射

AM263x ControlCARD 有两个版本，称为 E1 和 E2。ControlCARD 仅使用超集器件 (AM2634) 构建。客户可以通过购买满足其系统要求的精选 OPN 器件 (AM2642 或 AM2631) 来缩减规模。

E2 版 ControlCARD 的电路板具有多处设计变更。[AM263x Sitara ControlCARD 硬件用户指南](#) 中介绍了这些变更。有关标准模拟器件 ControlCARD 中 ADC 和 DAC 映射的差异，请参阅表 9-10。有关增强模拟器件，也请参阅 [AM263x Sitara ControlCARD 硬件用户指南](#)。

**表 9-10. 标准模拟器件的 E1 与 E2 引脚映射**

HSEC	E1	E2
1	NC	NC
2	NC	NC
3	TMS	TMS
4	NC	NC
5	TCK	TCK
6	TDO	TDO
7	GND	GND
8	TDI	TDI
9	DAC_OUT	ADC0_AIN0/DAC_OUT
10	GND	GND
11	ADC0_AIN0_P	ADC0_AIN1/DAC_OUT
12	ADC0_AIN0_n	ADC1_AIN0
13	GND	GND
14	ADC0_AIN1_p	ADC1_AIN1
15	ADC0_AIN1_n	ADC0_AIN2
16	GND	GND
17	ADC0_AIN2_p	ADC0_AIN3
18	ADC0_AIN2_n	ADC1_AIN2
19	GND	GND
20	ADC1_AIN0_p	ADC1_AIN3
21	ADC1_AIN0_n	ADC0_AIN4
22	GND	GND
23	ADC1_AIN1_p	ADC0_AIN5
24	ADC1_AIN1_n	ADC1_AIN4

表 9-10. 标准模拟器件的 E1 与 E2 引脚映射 (continued)

HSEC	E1	E2
25	ADC1_AIN2_p	NC/ADC_CAL0
26	ADC1_AIN2_n	ADC1_AIN5
27	ADC2_AIN0_p	NC/ADC_CAL1
28	ADC2_AIN0_n	NC
29	GND	GND
30	ADC2_AIN1_p	NC
31	ADC2_AIN1_n	ADC2_AIN0
32	NC	GND
33	ADC2_AIN2_p	ADC2_AIN1
34	ADC2_AIN2_n	NC
35	GND	GND
36	NC	NC
37	NC	ADC2_AIN2
38	GND	GND
39	NC	ADC2_AIN3
40	NC	NC
41	NC	NC
42	NC	NC
43	ADC_VREFLO	GND
44	NC	NC
45	ADC_VREFhi	ADC_VREFH

## 10 总结

所有 AM263x 系列器件都彼此引脚对引脚兼容。可以根据器件型号来确定这些器件之间的差异，如本文档中所述。AM2634 的 MCU\_PLUS\_SDK 和 EVM 可集成到任何子集器件中。工程师可以使用 MCU\_PLUS\_SDK\_AM263x 和 EVM 在任何 AM263x 器件上开发和启动工程。

## 11 参考文献

1. 德州仪器 (TI) , [AM263x Sitara™ 微控制器 数据表](#)。
2. 德州仪器 (TI) , [AM263x ControlCARD 硬件用户指南](#)。
3. 德州仪器 (TI) , [AM263x LaunchPad 用户指南](#)。

## 重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2023，德州仪器 (TI) 公司