Application Note

AM263x Device Nomenclature and Subset Devices

Sri Vidya Gunturi, Sanjeev Karaiyan

ABSTRACT

The AM263x Sitara™ Arm® microcontrollers are built to meet the complex real-time processing needs of next-generation industrial and automotive embedded products. The AM263x MCU family consists of multiple pin-to-pin compatible devices with up to four 400-MHz Arm Cortex®-R5F cores. The family is designed for advanced motor control and digital power control applications with advanced analog modules. This document details the differences between these family of devices based on their part number and compared against the superset device AM2634.

Table of Contents

Trademarks

Sitara™ and LaunchPad™ are trademarks of Texas Instruments.
Arm® and Cortex® are registered trademarks of Arm Ltd.
All trademarks are the property of their respective owners.

1 Introduction

The current MCU_PLUS_SDK for AM263x is developed to support the entire series of AM263x devices. This document points out the examples of SDK designed for different subset devices of AM263x and explains the steps to modify existing examples to support the subset devices AM2632 and AM2631.

The AM263x LaunchPad™ Development Kit and ControlCARD are developed for the superset device AM2634. The users working on other AM263x SOCs can get started on the AM2634 EVMs as these are pin-to-pin compatible with AM2632 and AM2631. This document™ explains how to adapt these EVMs for the other family of devices. Users working on different AM263x family of devices can seamlessly use the same SDK libraries and EVMs for their different SOC projects.

2 Device Nomenclature

The features and characteristics of any AM263x IP are identified by the part number. Table 2-1 describes how to decrypt features. See the data sheet to get the details of speed, memory, number of Control IPs, networking settings, and so on. Table 2-2 is classified according to the part number as explained in Table 2-1.

Table 2-1. Device Nomenclature Based on Part Number

AM263	4	C	O	K	F	H	M	ZCZ	R	Q1
Common IP Name	Number of R5 cores	Device Revision	Device Operating Performance Points	Features	Functional Safety	Security	Junction Temperature	Package Designator	Carrier	Automotive Designator

Table 2-2 specifies differences between six part numbers. More information regarding these differences is available in the Device Comparison section of *AM263x Sitara™ Microcontrollers*, data sheet.

Table 2-2. Device Comparison of Six AM263x Part Numbers

Part Numbers	AM2634	AM2632	AM2631	AM2634-Q1	AM2632-Q1	AM2631-Q1
	AM2634COMFHA ZCZR	AM2632COLFHA ZCZR	AM2631CNDGHA ZCZR	AM2634COKFHM ZCZRQ1	AM2632COKFHM ZCZRQ1	AM2631CODGHM ZCZRQ1
Core Characteristics						
R5 Cores	4	2	1	4	2	1
Clock Speed	400 MHZ	400 MHZ	400 MHZ	400 MHZ	400 MHZ	400 MHZ
Memory	2 MB	2 MB	1 MB	2 MB	2 MB	2 MB
Safety and Security						
Functional Safety	Yes	Yes	No	Yes	Yes	No
Security	Yes	Yes	Yes	Yes	Yes	Yes
Auto Qual	No	No	No	Yes	Yes	Yes
Temperature	-40 to 105C	-40 to 105C	-40 to 105C	-40 to 150C	-40 to 150C	-40 to 150C
Control System Instances						
Analog	Enhanced Analog	Enhanced Analog	Standard Analog	Enhanced Analog	Enhanced Analog	Standard Analog
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
Networking Protocols						
Features	M	L	D	K	K	D
Bosch CAN-FD	Yes	Yes	Yes	Yes	Yes	Yes
EtherCAT	Yes	Yes	No	No	No	No
Kunbus Stacks (integrated Stacks)	Yes	No	No	No	No	No
ICSS-PRU	Yes	Yes	Yes	Yes	Yes	Yes
General Characteristics						
Package	15x15	15x15	15x15	15x15	15x15	15x15
Auto Qual	No	No	No	Yes	Yes	Yes

3 R5 Cores and TCM in AM263x

In AM263x multiple R5F cores are arranged in cluster with 256KB of shared tightly coupled memory (TCM) as shown in Figure 3-1.

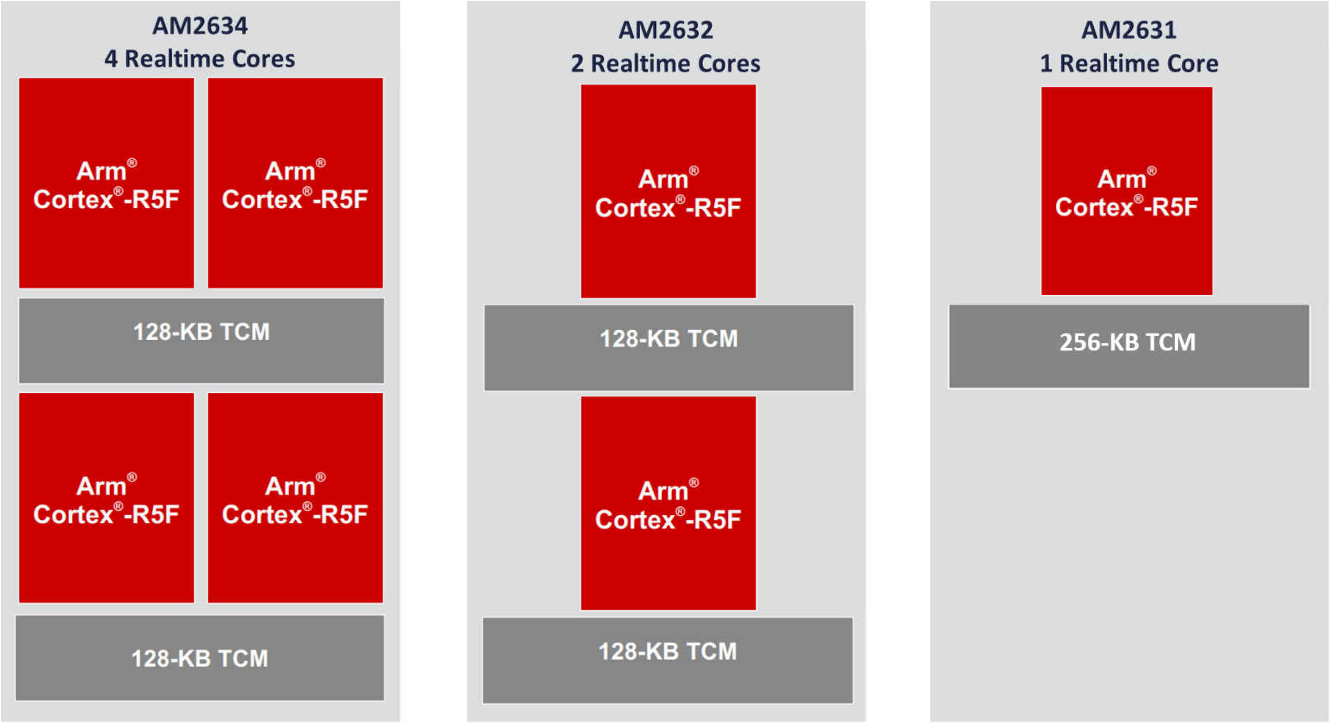


Figure 3-1. R5 cores and TCM in AM263x Family

3.1 R5 Core Nomenclature in Am263x

Here after in this document the R5 cores are defined as explained below based on the position of the core in the cluster:

AM2634 has two R5 clusters:

- The first cluster is called R50.
  - The first core in the first cluster is called core R50-0.
  - The second core in first cluster is called core R50-1.
- The second cluster is called R51.
  - The first core in the second cluster is called R51-0.
  - The second core in the second cluster is called R51-1.

AM2632 has two R5 clusters:

- The first cluster is called R50.
  - The first core in the first cluster is called core R50-0.
- The second cluster is called R51.
  - The first core in the second cluster is called R51-0.

Am2631 has one R5 cluster:

- The single cluster is called R50.
  - The first core in this cluster is called core R50-0

4 Example Support for AM263x Family

All the examples in the SDK are supported for AM2634 devices. Currently, IPC and Networking are not supported for some devices. The 8th character of the part numbers in Section 2 represent *Device Operating Performance Points* of the device. Similarly, the 6th character represents *number of R5 cores* in the device, see Section 2 for more information. For example in AM263**2**C**O**LFHAZCZR, *O* is the speed and memory setting and *2* represents two R5 cores. Section 4 lists supported examples in the MCU PLUS SDK for AM263x family of devices, based on the memory setting and number of cores.

Table 4-1. Examples in the MCU_PLUS_SDK for AM263x Family

Examples in the SDK	**Sub-modules**	**Memory Classification**		**Classification based on Number of Cores**		
		K,O,P	**N**	**4 cores**	**2 cores**	**1 core**
Example Projects		Supported	Supported	Supported	Supported	Supported
Software Diagnostics Library (SDL)		Supported	Supported	Supported	Supported	Supported
Empty Project (Multi-Core Example)		Supported	Supported	Supported	Supported, see also Section 6	Only r5ss0_0 example is supported. Multi-core example is not supported.
Hello World Project		Supported	Supported	Supported	Supported	Supported
Hello World C++ Project		Supported	Supported	Supported	Supported	Supported
OS Kernel and Driver Porting Layer (DPL)		Supported	Supported	Supported	Supported	Supported
SOC and Board Peripheral Drivers	Control IP Examples	Supported	Supported	Supported	Supported	Supported
	GPIO	Supported	Supported	Supported	Supported	Supported
	IPC (Multi-core example)	Supported	Supported	Supported	Supported, see also Section 5	Not Supported.
	HSM	Supported	Supported	Supported	Supported	Supported
	Serial Communication Examples	Supported	Supported	Supported	Supported	Supported
	EDMA	Supported	Supported	Supported	Supported	Supported
	Boot	Supported	Supported	Supported	Supported	Supported
	WatchDog	Supported	Supported	Supported	Supported	Supported
Secondary Bootloader (SBL)		Supported	Supported	Supported	Supported	Supported
Real Time Debug		Supported	Supported	Supported	Supported	Supported
Industrial Communications Toolkit		Supported	Supported	Supported	Supported	Supported

Table 4-1. Examples in the MCU_PLUS_SDK for AM263x Family (continued)

Examples in the SDK	Sub-modules	Memory Classification		Classification based on Number of Cores		
		K,O,P	N	4 cores	2 cores	1 core
Networking	Enet CPSW EST Example	Supported	Supported	Supported	Supported	Supported
	Enet Layer 2 CPSW Example	Supported	Supported	Supported	Supported	Supported
	Enet Layer 2 CPSW SWITCH Example	Supported	Supported	Supported	Supported	Supported
	Enet Layer 2 Multi-Channel Example	Supported	Supported	Supported	Supported	Supported
	Enet Loopback Example	Supported	Supported	Supported	Supported	Supported
	Enet Lwip CPSW RAW HTTP Server	Supported	Supported	Supported	Supported	Supported
	Enet Lwip Socket Example	Supported	Supported	Supported	Supported	Supported
	Enet Tx Scatter Gather Example	Supported	Not Supported	Supported	Supported, not supported if Memory setting is N	Supported, not supported if Memory setting is N
	Enet Lwip TCP Client Example	Supported	Not Supported	Supported	Supported, not supported if Memory setting is N	Supported, not supported if Memory setting is N
	Enet Lwip TCP Server Example	Supported	Not Supported	Supported	Supported, not supported if Memory setting is N	Supported, not supported if Memory setting is N
	Enet Lwip UDP IGMP Server Example	Supported	Not Supported	Supported	Supported, not supported if Memory setting is N	Supported, not supported if Memory setting is N
	Enet Lwip UDP Server Example	Supported	Not Supported	Supported	Supported, not supported if Memory setting is N	Supported, not supported if Memory setting is N
	Enet Lwip CPSW Example	Supported	Not Supported	Supported	Supported, not supported if Memory setting is N	Supported, not supported if Memory setting is N
	Enet CPSW TimeSync PTP Demo	Supported	Not Supported	Supported	Supported, not supported if Memory setting is N	Supported, not supported if Memory setting is N
	Enet CPSW Operation Modes Demo	Supported	Not Supported	Supported	Supported, not supported if Memory setting is N	Supported, not supported if Memory setting is N
MATHLIB Benchmark		Supported	Supported	Supported	Supported	Supported
SECURITY		Supported	Supported	Supported	Supported	Supported
Software Diagnostics Library (SDL)		Supported	Supported	Supported	Supported	Supported

5 IPC Example Support for Two-Core Devices (AM2632)

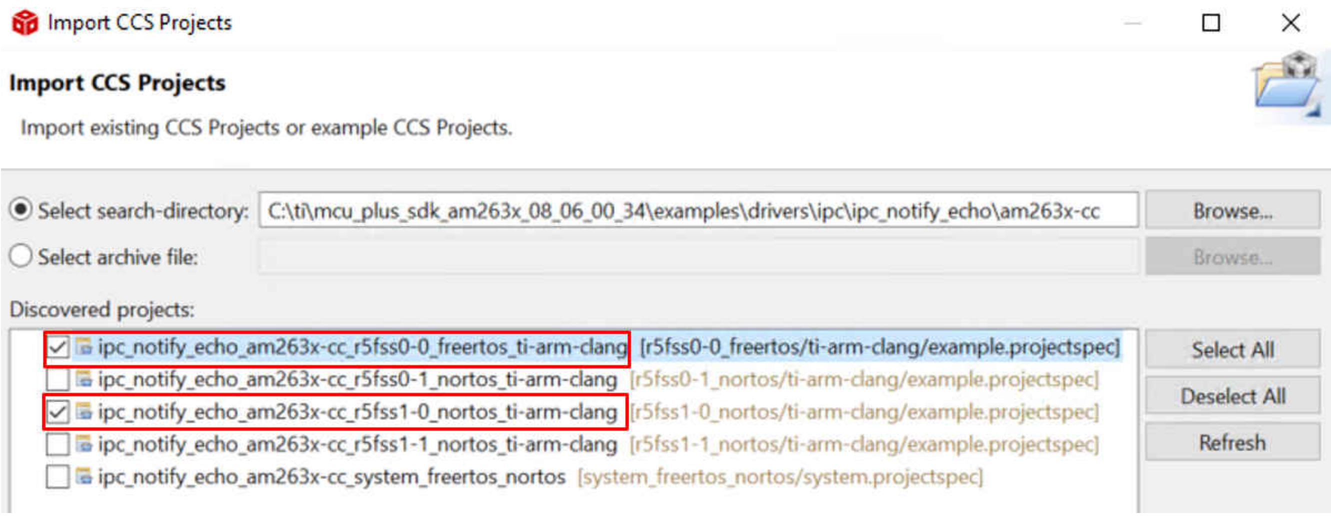
Changes in the IPC Notify Example to build through CCS to create a dual core .appimage for AM2632:

This section contains changes made in the IPC Notify example to create a Dual Lockstep Core (two cores) example from an existing four-core IPC Notify example.

5.1 Option 1 Using MulticoreImageGen.js

This section provides a simple method to create a Dual Lockstep Core (two cores) example from an existing four-core IPC Notify example. This example does not involve any system project; therefore, the user must use the command line to create a final combined app image. Steps for creating a Dual Core Appimage are described below.

- Import IPC Notify example for R50-0 and R51-0 (see also Section 3.1) from the SDK as shown below. Ignore the system project.



Import CCS Projects

Import CCS Projects

Import existing CCS Projects or example CCS Projects.

● Select search-directory: C:\ti\mcu_plus_sdk_am263x_08_06_00_34\examples\drivers\ipc\ipc_notify_echo\am263x-cc [Browse...]

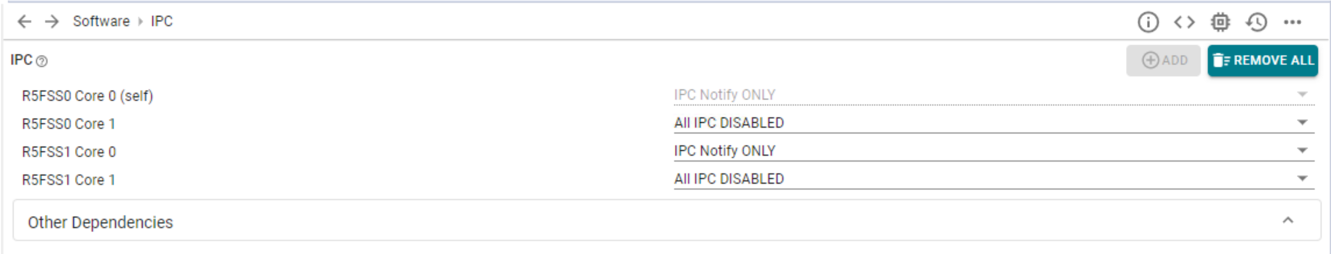
○ Select archive file: [Browse...]

Discovered projects:

☑ ipc_notify_echo_am263x-cc_r5fss0-0_freertos_ti-arm-clang [r5fss0-0_freertos/ti-arm-clang/example.projectspec]
□ ipc_notify_echo_am263x-cc_r5fss0-1_nortos_ti-arm-clang [r5fss0-1_nortos/ti-arm-clang/example.projectspec]
☑ ipc_notify_echo_am263x-cc_r5fss1-0_nortos_ti-arm-clang [r5fss1-0_nortos/ti-arm-clang/example.projectspec]
□ ipc_notify_echo_am263x-cc_r5fss1-1_nortos_ti-arm-clang [r5fss1-1_nortos/ti-arm-clang/example.projectspec]
□ ipc_notify_echo_am263x-cc_system_freertos_nortos [system_freertos_nortos/system.projectspec]

Figure 5-1. IPC Notify Example for R50-0 and R51-0

- Modify your example.syscfg to support the IPC notify feature to the cores 0-0 and 1-0. Subsequently, disable IPC Notify feature for cores 1-0 and 1-1.

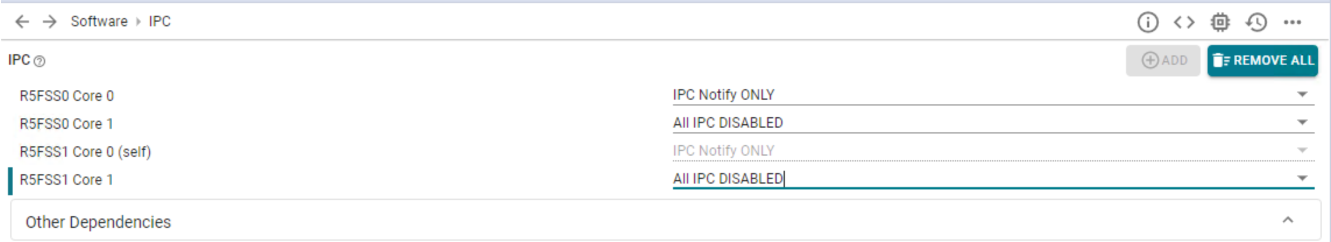


IPC

R5FSS0 Core 0 (self)	IPC Notify ONLY
R5FSS0 Core 1	All IPC DISABLED
R5FSS1 Core 0	IPC Notify ONLY
R5FSS1 Core 1	All IPC DISABLED

Other Dependencies

Figure 5-2. IPC Notify R50_0 Syscfg



IPC

R5FSS0 Core 0	IPC Notify ONLY
R5FSS0 Core 1	All IPC DISABLED
R5FSS1 Core 0 (self)	IPC Notify ONLY
R5FSS1 Core 1	All IPC DISABLED

Other Dependencies

Figure 5-3. IPC Notify R51_0 Syscfg

- Change the Remote Core ID list to support only core 1-0, and remove remaining cores as shown below. Make this change in ipc_notify_echo .c file of both ipc_notify_echo_am263x-cc_r5fss0-0_nortos_ti-arm-clang and ipc_notify_echo_am263x-cc_r5fss1-0_nortos_ti-arm-clang projects.

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

- Build these projects individually. After building, .rprc files are available for both the projects.
- Combine these .rprc to form a dual-core .appimage using the following command.

```
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
```

The command format for MulticoreImage Generation is given below.

```
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 Option Two (Migration Guide from a Four-Core System Project to a Two-Core System Project)

This process helps to import the existing IPC Notify example as a 2-core system project with R50_0 as the main core and R51_0 as the remote core. The following changes need to be done in the make files in the SDK for IPC Notify. Choose the folder depending on your device type CC or LP.

Open the `system_freertos_nortos` makefile at the locaion:`examples\drivers\ipc\ipc_notify_echo\am263x-cc\system_freertos_nortos\makefile`.

- Retain the CORE_0 definition. Remove CORE_1 and CORE_2 definitions from lines 18 and 19 as these are not required.
- Define CORE_1 as below:

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

- Remove $(CORE_3) and $(CORE_2) from CORES. And define CORES as shown.

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

- Remove r5fss0-1 and r5fss1-1 instances from all: syscfg section.

```
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)
```

- Remove r5fss0-1 and r5fss1-1 instances from clean section.

```
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)
```

- Remove r5fss0-1 and r5fss1-1 instances from scrub section.

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

- Remove r5fss0-1 and r5fss1-1 instances from MULTI_CORE_APP_PARAMS section.

```
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) \
```

- Remove r5fss0-1 and r5fss1-1 instances from MULTI_CORE_APP_PARAMS_XIP section.

```
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) \
```

- Remove r5fss0-1 and r5fss1-1 instances from MULTI_CORE_BOOTIMAGE_DEPENDENCY section.

```
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 \
```

Open the projectspec makefile at the location: `examples\drivers\ipc\ipc_notify_echo\am263x-cc\system_freertos_nortos\ makefile_projectspec`.

Make the following changes in this makefile:

- Remove r5fss0-1 and r5fss1-1 instances from clean section to support only r5fss0-0 and 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
```

Open the bootimage_gen makefile at the location: `examples\drivers\ipc\ipc_notify_echo\am263x-cc\system_freertos_nortos\ makefile_system_ccs_bootimage_gen`.

Make the following changes in this makefile:

- Remove r5fss0-1 and r5fss1-1 instances from MULTI_CORE_APP_PARAMS section.

```
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) \
```

- Remove r5fss0-1 and r5fss1-1 instances from MULTI_CORE_APP_PARAMS_XIP section.

```
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) \
```

Open the system projectspec file at the location: `examples\drivers\ipc\ipc_notify_echo\am263x-cc\system_freertos_nortos\ system.projectspec`.

Make the following changes in thisprojectspec file:

- Remove r5fss0-1 and r5fss1-1 projectspec files from getting imported with the system_freertos_nortos project (lines 4 and 6).

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

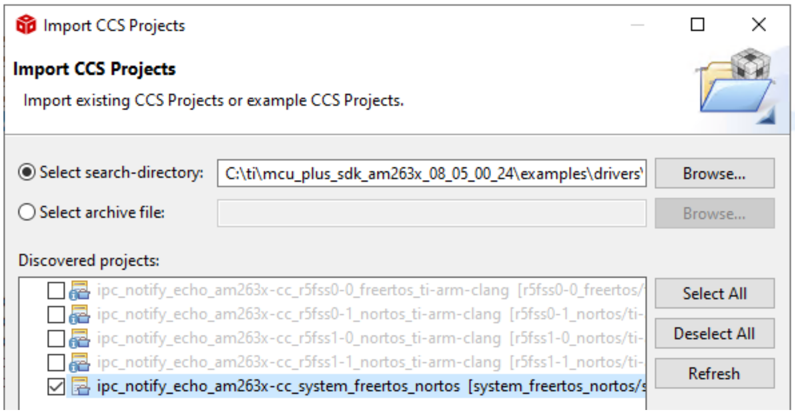
Open the system projectspec file at the location: `examples\drivers\ipc\ipc_notify_echo\am263x-cc\system_freertos_nortos\ system.xml`.

Make the following changes in this projectspec file:

- Remove r5fss0-1 and r5fss1-1 cores project configurations from the xml file and only keep 0-0 and 1-0 cores. Modify core 1 as r5fss1-0 core as shown below.

```
<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"/>
```

After modifying the system project make files, import this example into your CCS. This step is the same as the other system projects.



Import CCS Projects

Import existing CCS Projects or example CCS Projects.

● Select search-directory: C:\ti\mcu_plus_sdk_am263x_08_05_00_24\examples\drivers\

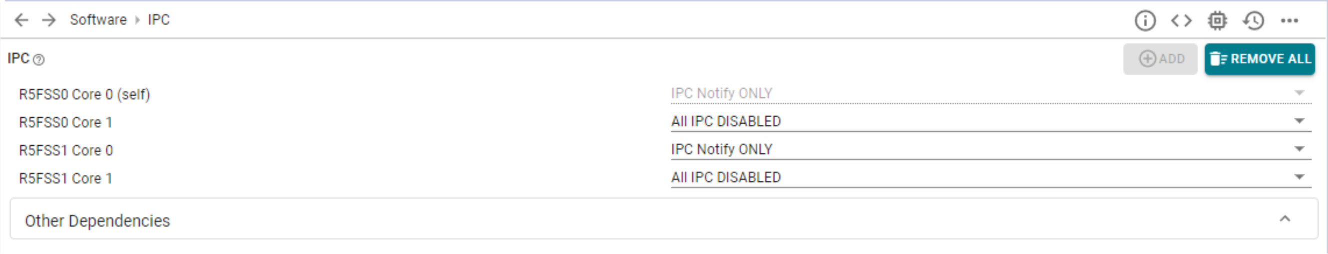
○ Select archive file:

Discovered projects:

- □ ipc_notify_echo_am263x-cc_r5fss0-0_freertos_ti-arm-clang [r5fss0-0_freertos/
- □ ipc_notify_echo_am263x-cc_r5fss0-1_nortos_ti-arm-clang [r5fss0-1_nortos/ti-
- □ ipc_notify_echo_am263x-cc_r5fss1-0_nortos_ti-arm-clang [r5fss1-0_nortos/ti-
- □ ipc_notify_echo_am263x-cc_r5fss1-1_nortos_ti-arm-clang [r5fss1-1_nortos/ti-
- ☑ ipc_notify_echo_am263x-cc_system_freertos_nortos [system_freertos_nortos/s

Figure 5-4. Importing Dual-Core System Project for IPC Notify

Modify your example.syscfg to support IPC notify feature to the cores 0-0 and 1-0. Subsequently, disable IPC Notify feature for cores 1-0 and 1-1.

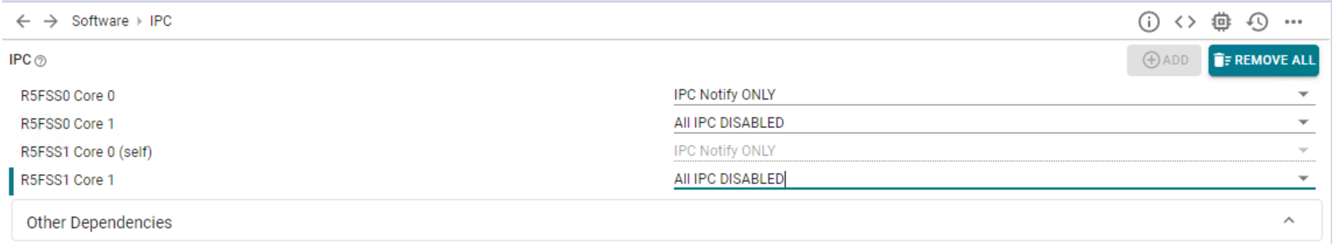


IPC

R5FSS0 Core 0 (self)	IPC Notify ONLY
R5FSS0 Core 1	All IPC DISABLED
R5FSS1 Core 0	IPC Notify ONLY
R5FSS1 Core 1	All IPC DISABLED

Other Dependencies

Figure 5-5. IPC Notify R50_0 Syscfg



R5FSS0 Core 0	IPC Notify ONLY
R5FSS0 Core 1	All IPC DISABLED
R5FSS1 Core 0 (self)	IPC Notify ONLY
R5FSS1 Core 1	All IPC DISABLED

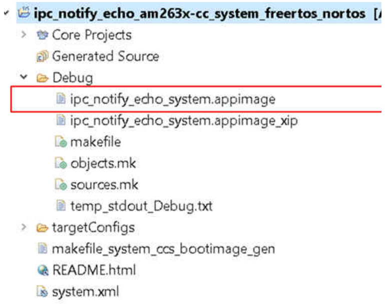
Other Dependencies

Figure 5-6. IPC Notify R50_0 Syscfg

Change the Remote Core ID List to support only core 1-0, and remove rest of the cores as shown below. Make this change in ipc_notify_echo .c file of both ipc_notify_echo_am263x-cc_r5fss0-0_nortos_ti-arm-clang and ipc_notify_echo_am263x-cc_r5fss1-0_nortos_ti-arm-clang projects.

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

Now build the system project to generate a combined app image for core 0-0 and 1-0. This app image can be flashed on to your AM2632 device.



```
ipc_notify_echo_am263x-cc_system_freertos_nortos [/
  Core Projects
  Generated Source
  Debug
    ipc_notify_echo_system.appimage
    ipc_notify_echo_system.appimage_xip
    makefile
    objects.mk
    sources.mk
    temp_stdout_Debug.txt
  targetConfigs
  makefile_system_ccs_bootimage_gen
  README.html
  system.xml
```

Figure 5-7. Appimage Generated for Dual-Core Project

Note

AM2631 is a single core device and IPC Examples are not supported for single-core devices.

6 System Project Example Support for Two-Core Devices (AM2632)

In MCU-PLUS-SDK_AM263x, system projects are configured for 2-core devices by default. But the examples are present for core - R50_0 and R50_1. These examples can be modified to enable the system project for the cores R50_0 and R51_0.

To find the system projects, go to → C:\ti\mcu_plus_sdk_am263x_08_05_00_24\examples\empty. Here, the SDK has 2 System Projects: freeRtos System Project and Nortos System Project. The engineer can choose a system project for the application.

The following files need to be modified:

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

Replace r5fss0-1 core with r5fss1-0 in these files. These changes are explained in the Section 5.

Note

AM2631 is a single core device; multi-core system projects are not supported for single-core devices.

7 Target Configuration in CCS

7.1 Prerequisites

- Make sure you have installed CCS as mentioned in *Download, Install and Setup CCS*.
- Make sure the UART port used for console is identified as mentioned in *Setup UART Terminal*.
- Make sure you have the EVM power cable, JTAG cable, UART cable connected as shown in *Cable Connections*.
- Make sure you have done the steps for a SOC initialization method.
  - Recommended method is *SOC Initialization using the Binary Flashed in QSPI memory*.
  - Other options, if recommended method cannot be used, are mentioned in *SOC Initialization*.
- Make certain EVM boot mode switch is setup correctly based on the SOC initialization method.
  - For the *recommended* method, *SOC Initialization using the Binary Flashed in QSPI memory,* the boot mode is *QSPI BOOT MODE*.
- Make certain the UART or CCS console logs on doing EVM Power-on indicate that SOC initialization is successful.
- Make certain that you have built the example of interest as mentioned in *Build a Hello World example*.

7.2 Creating a Target Configuration

- Go to *View* > *Target Configuration*.

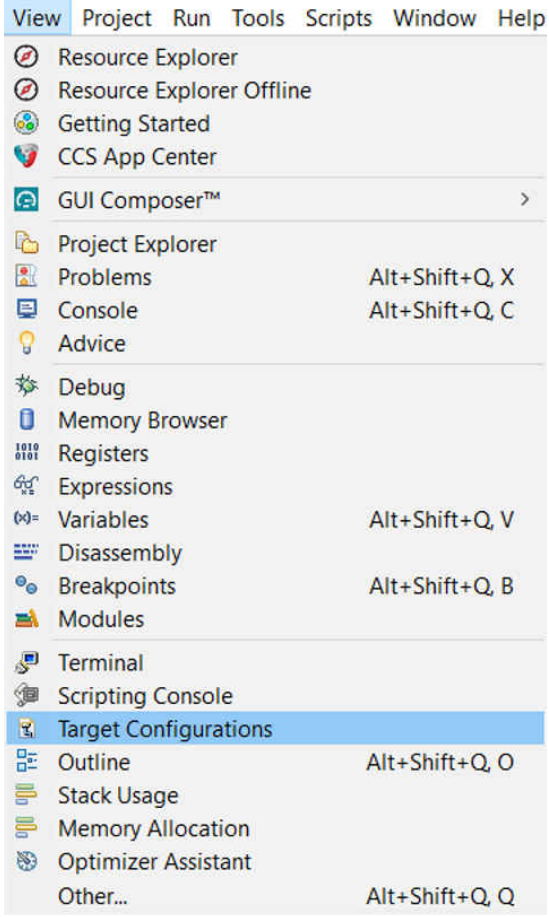


Figure 7-1. Target Configurations in View

- Create a new target configuration.

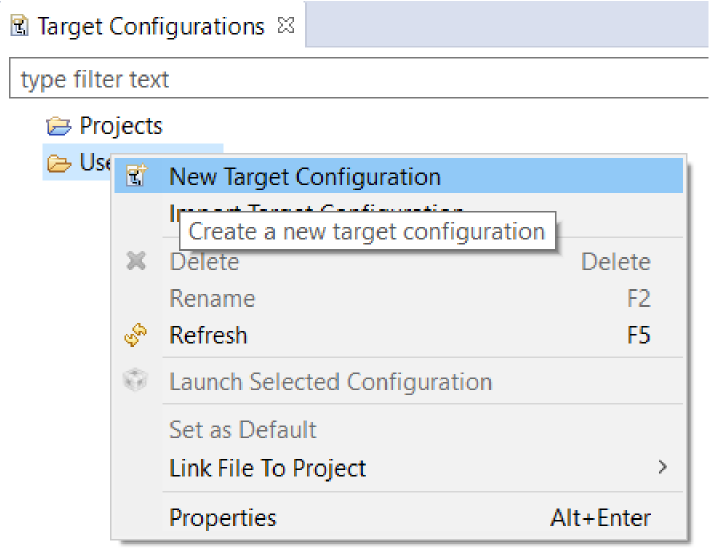
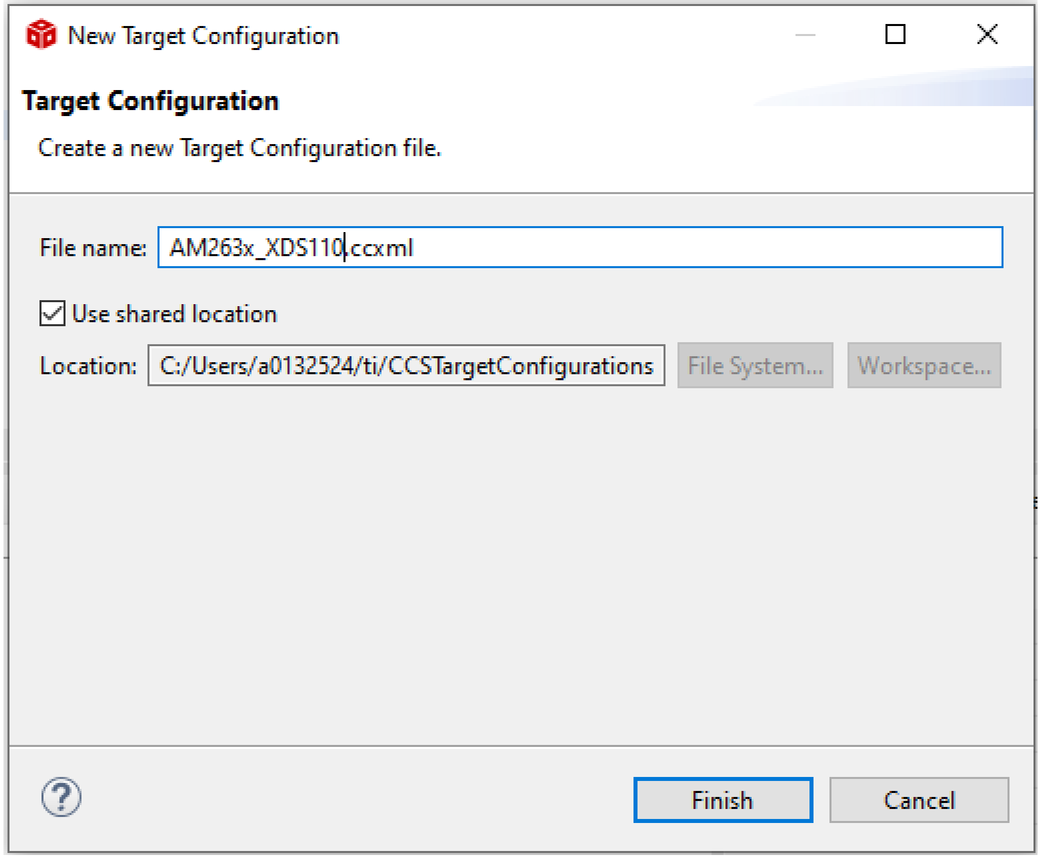


Figure 7-2. New Target Configuration

- Give a name to the new target configuration, typically {soc name}_{JTAG type}.



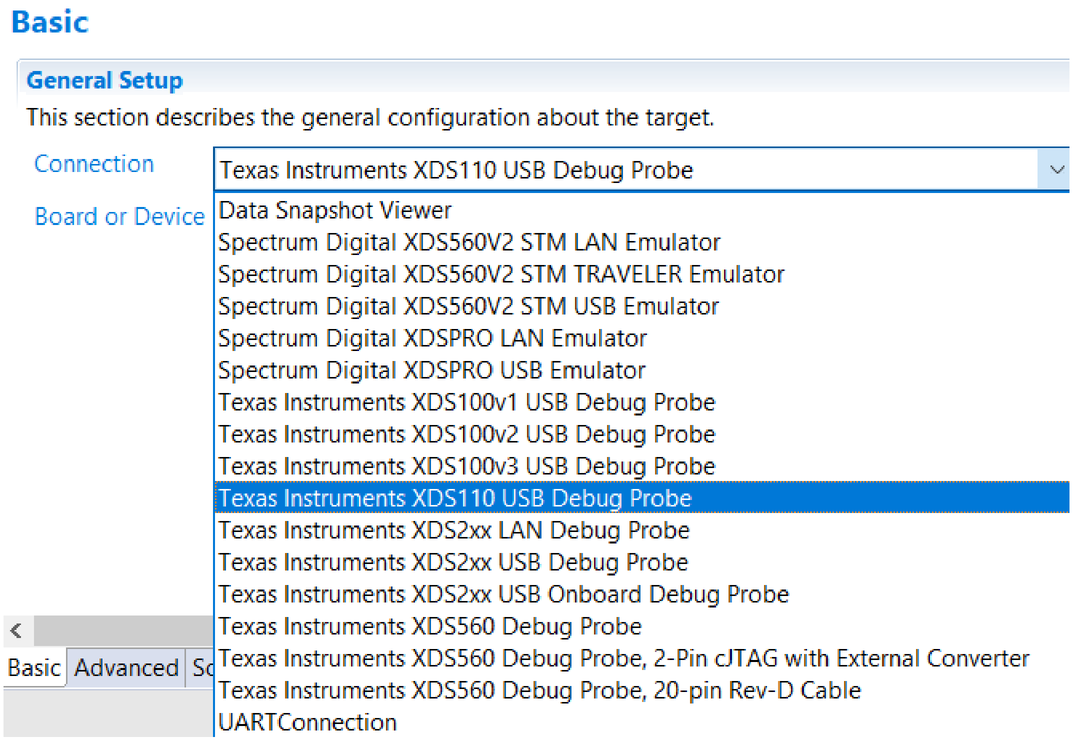
File name: AM263x_XDS110.ccxml

☑ Use shared location

Location: C:/Users/a0132524/ti/CCSTargetConfigurations

Figure 7-3. New Target Configuration Window

- Select connection as XDS110 USB Debug Probe.



Basic

General Setup

This section describes the general configuration about the target.

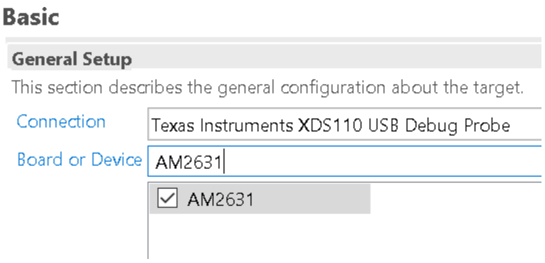
Connection: Texas Instruments XDS110 USB Debug Probe

Board or Device

Data Snapshot Viewer
Spectrum Digital XDS560V2 STM LAN Emulator
Spectrum Digital XDS560V2 STM TRAVELER Emulator
Spectrum Digital XDS560V2 STM USB Emulator
Spectrum Digital XDSPRO LAN Emulator
Spectrum Digital XDSPRO USB Emulator
Texas Instruments XDS100v1 USB Debug Probe
Texas Instruments XDS100v2 USB Debug Probe
Texas Instruments XDS100v3 USB Debug Probe
Texas Instruments XDS110 USB Debug Probe
Texas Instruments XDS2xx LAN Debug Probe
Texas Instruments XDS2xx USB Debug Probe
Texas Instruments XDS2xx USB Onboard Debug Probe
Texas Instruments XDS560 Debug Probe
Texas Instruments XDS560 Debug Probe, 2-Pin cJTAG with External Converter
Texas Instruments XDS560 Debug Probe, 20-pin Rev-D Cable
UARTConnection
<
Basic
Advanced
Sc

Figure 7-4. Debugger Configuration

- In *Board or Device Setting*, choose AM2632 or AM2631 depending on the device.



Basic

General Setup

This section describes the general configuration about the target.

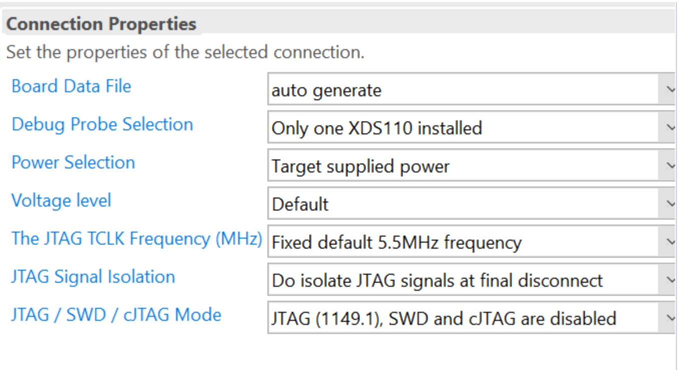
Connection: Texas Instruments XDS110 USB Debug Probe

Board or Device: AM2631

☑ AM2631

Figure 7-5. Choosing the Device or Part Number

- Under Advanced tab, make certain that all the settings are configured.



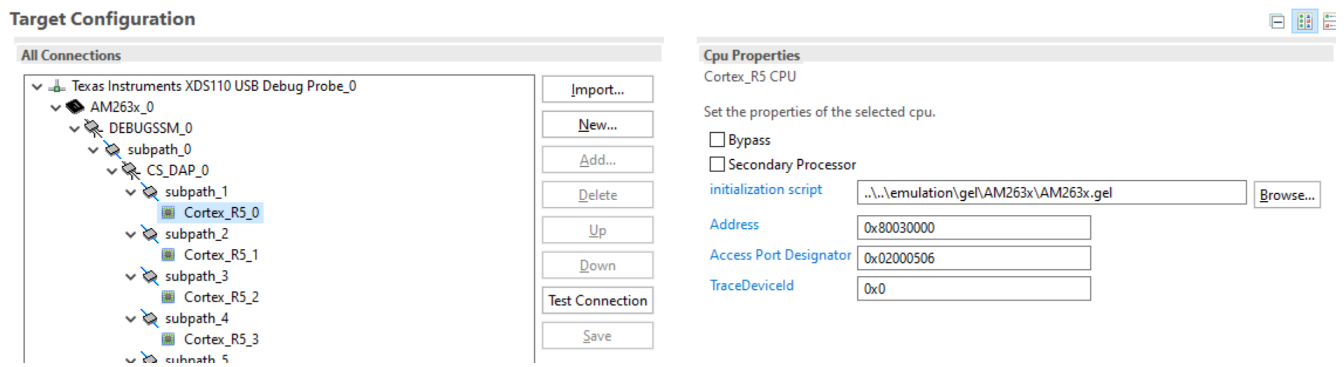
Connection Properties

Set the properties of the selected connection.

Board Data File	auto generate
Debug Probe Selection	Only one XDS110 installed
Power Selection	Target supplied power
Voltage level	Default
The JTAG TCLK Frequency (MHz)	Fixed default 5.5MHz frequency
JTAG Signal Isolation	Do isolate JTAG signals at final disconnect
JTAG / SWD / cJTAG Mode	JTAG (1149.1), SWD and cJTAG are disabled

Figure 7-6. JTAG Frequency Configuration

- Under Advanced tab, select Cortex_R5_0.
- Make certain that the device csp gel file path is in the Initialization script field.



Target Configuration

All Connections

Texas Instruments XDS110 USB Debug Probe_0
AM263x_0
DEBUGSSM_0
subpath_0
CS_DAP_0
subpath_1
Cortex_R5_0
subpath_2
Cortex_R5_1
subpath_3
Cortex_R5_2
subpath_4
Cortex_R5_3
subpath_5

Cpu Properties

Cortex_R5 CPU

Set the properties of the selected cpu.

□ Bypass
□ Secondary Processor

initialization script: ..\..\emulation\gel\AM263x\AM263x.gel
Address: 0x80030000
Access Port Designator: 0x02000506
TraceDeviceId: 0x0

Figure 7-7. GEL Scripts

- Click *Save* to save the newly created target configuration.
- Now complete the *EVM Setup* to prepare the EVM for running programs.

8 Connecting to the Target Core

While connecting to the cores to flash your image, you can come across multiple cores supported by all AM263x devices. Please select the following cores depending on your device only.

Table 8-1. Available Target Cores

Device	Project	Cores that can be connected to
AM2631	Single Core Project	Cortex_R5_0_0
AM2632	Single Core Project	Cortex_R5_0_0 or Cortex_R5_1_0
	Dual Lockstep Core Project	Cortex_R5_0_0 Cortex_R5_1_0
AM2634	Any combination project is possible	Cortex_R5_0_0 Cortex_R5_0_1 Cortex_R5_1_0 Cortex_R5_1_1

Note

Lockstep and Dual Core Configuration Handling in subset devices: The individual clusters of AM2632 and AM2631 are configured in lockstep mode. Dual Core mode of a cluster does not exist in these devices.

9 Hardware Description for Launch Pad and Control Card

LaunchPads and ControlCARDs are only built with the superset device (AM2634). Then customers can scale down by purchasing select OPN devices (AM2642 or AM2631) that meet their system requirements.

Launch Pad and ControlCARD for AM2634, AM2632, and AM2631 for Enhanced Analog devices have similar hardware descriptions and pin mux. These hardware details and pin mapping are listed in the *AM263x LaunchPad User Guide* and *Control Card User Guide*. See the *Features Character* (9th character) in the part number in Section 2. Table 9-1 has differences between Standard Analog and Enhanced Analog devices. The only difference is the number of control peripherals integrated.

Table 9-1. Enhanced Versus Standard Analog Devices

Analog	Enhanced Analog		Standard Analog	
Feature Description	**J, K, L, M**		**C, D, E, F**	
	Number	**Instances**	**Number**	**Instances**
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

Table 9-1. Enhanced Versus Standard Analog Devices (continued)

Analog	Enhanced Analog		Standard Analog	
Feature Description	**J, K, L, M**		**C, D, E, F**	
	Number	**Instances**	**Number**	**Instances**
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 Launchpad Pinout for Standard Analog Devices

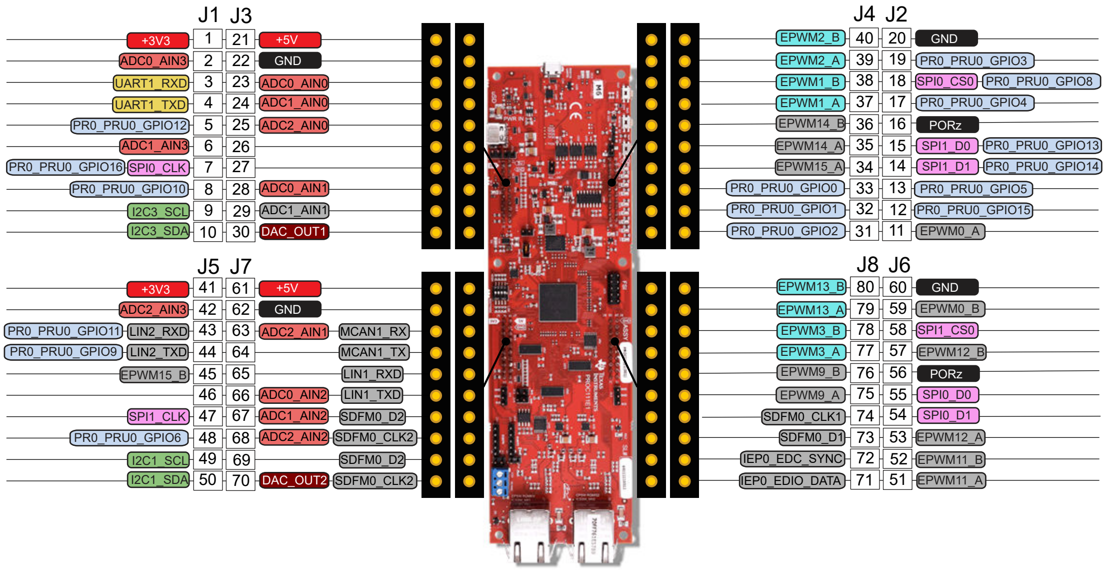


Figure 9-1. AM263x Standard Analog Launchpad PinOut

The above picture shows LaunchPad pinout for standard analog devices. In this image, the signals that are shown in color are signals that comply with the LaunchPad BoosterPack Pinout Standard. The signals shown in gray are signals that the AM263x supports, but these signals do not comply with the LP/BP pinout standard. The AM263x LaunchPad supports two fully independent BoosterPack XL connectors. BoosterPack site #1 (J1/J3, J2/J4) is located in between the SoC and the micro-B USB Connector. BoosterPack site #2 (J5/J7, J6/J8) is located in between the SoC and the RJ45 connectors. Each GPIO has multiple functions available through the GPIO mux. The signals connected from the SoC to the BoosterPack headers include:

- Various ADC inputs
- DAC Out
- UART1
- Various GPIO signals
- SPI0 and SPI1
- I2C1 and I2C3
- Various EPWM Channels
- LIN1 and LIN2
- MCAN1
- SDFM0

9.2 ADC and DAC Mapping in Launchpad for Standard Analog

The AM263x LaunchPad maps 16 ADC inputs to the BoosterPack header in standard analog devices. All of the ADC inputs that are used in the Launchpad are ESD protected. The ADCs which are present in enhanced analog but not present in the standard analog devices are marked with an *X* as shown in Figure 9-2.

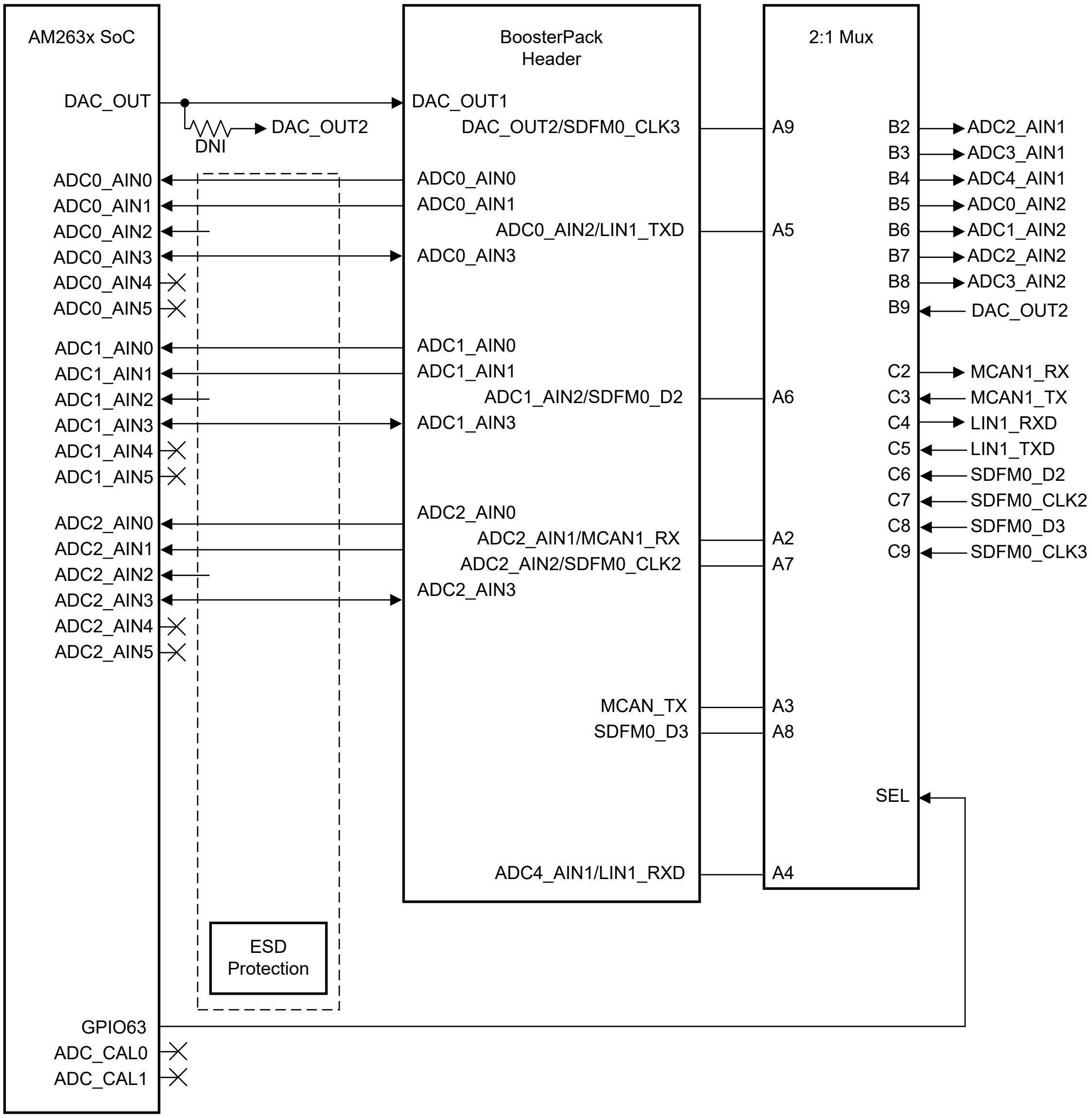


Figure 9-2. ADC and DAC Mapping for Standard Analog Devices

9.3 Pinmux Mapping - Standard Analog - Launch Pad

The various pinmux options for the BoosterPack connector pins are given in Table 9-2.

Table 9-2. Pinmux Options for J1

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
J1.1	3V3									
J1.2	ADC0_AIN3									
J1.3	UART1_RXD	LIN1_RXD				EPWM16_A	GPMC0_AD6	GPIO75		
J1.4	UART1_TXD	LIN1_TXD				EPWM16_B	GPMC0_AD7	GPIO76		
J1.5	PR0_PRU0_GPIO12		RMII2_TXD1	RGMII2_TD1	MII2_TXD1	EPWM28_B	GPMC0_A8	GPIO100		
J1.6	ADC1_AIN3									

Table 9-2. Pinmux Options for J1 (continued)

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
J1.7	SPI0_CLK	UART3_TXD	LIN3_TXD				FSITX0_CLK	GPIO12		
	PR0_PRU0_GPIO16			RGMII2_TXC	MII2_TXCLK	EPWM27_A	GPMC0_A5	GPIO97		
J1.8	PR0_PRU0_GPIO10		RMII2_CRS_DV	PR0_UART0_RTSn	MII2_CRS	EPWM23_A	GPMC0_WAIT0	GPIO89		
J1.9	EPWM8_B	UART4_RXD	I2C3_SCL				FSITX2_DATA0	GPIO60		
J1.10	EPWM8_A	UART4_TXD	I2C3_SDA				FSITX2_CLK	GPIO59		

Table 9-3. Pinmux Options for J2

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
J2.11	EPWM0_A							GPIO43		
J2.12	PR0_PRU0_GPIO15		RMII2_TX_EN	RGMII2_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			RGMII2_TD3	MII2_TXD3	EPWM29_B	GPMC0_A10	GPIO102		
J2.15	SPI0_D0						FSITX0_DATA0	GPIO13		
	PR0_PRU0_GPIO13			RGMII2_TD2	MII2_TXD2	EPWM29_A	GPMC0_A9	GPIO101		
J2.16	PORz									
J2.17	PR0_PRU0_GPIO4			RGMII2_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			RGMII2_RD3	MII2_RXD3	EPWM26_B	GPMC0_A4	GPIO96		
J2.20	GND									

Table 9-4. Pinmux Options for J3

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
J3.21	5 V									
J3.22	GND									
J3.23	ADC0_AIN0									
J3.24	ADC1_AIN0									

Table 9-4. Pinmux Options for J3 (continued)

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
J3.25	ADC2_AIN0									
J3.26	ADC3_AIN0									
J3.27	ADC4_AIN0									
J3.28	ADC0_AIN1									
J3.29	ADC1_AIN1									
J3.30	DAC_OUT									

Table 9-5. Pinmux Options for J4

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
J4.31	PR0_PRU0_GPIO2			RGMII2_RD2	MII2_RXD2	EPWM26_A	GPMC0_A3	GPIO95		
J4.32	PR0_PRU0_GPIO1		RMII2_RXD1	RGMII2_RD1	MII2_RXD1	EPWM25_B	GPMC0_A2	GPIO94		
J4.33	PR0_PRU0_GPIO0		RMII2_RXD0	RGMII2_RD0	MII2_RXD0	EPWM25_A	GPMC0_A1	GPIO93		
J4.34	EPWM15_A	UART5_TXD	MII1_COL				GPMC0_AD4	GPIO73		
J4.35	EPWM14_A	UART1_DSRn					GPMC0_AD2	GPIO71		
J4.36	EPWM14_B		MII1_RX_ER				GPMC0_AD3	GPIO72		
J4.37	EPWM1_A							GPIO45		
J4.38	EPWM1_B							GPIO46		
J4.39	EPWM2_A							GPIO47		
J4.40	EPWM2_B							GPIO48		

Table 9-6. Pinmux Options for J5

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
J5.41	3 V									
J5.42	ADC2_AIN3									
J5.43	LIN2_RXD	UART2_RXD	SPI2_D0					GPIO21		
	PR0_PRU0_GPIO11		RMII2_TXD0	RGMII2_TD0	MII2_TXD0	EPWM28_A	GPMC0_A7	GPIO99		
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_AD5	GPIO74		

Table 9-6. Pinmux Options for J5 (continued)

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
J5.46	ADC3_AIN3									
J5.47	SPI1_CLK	UART4_RXD	LIN4_RXD			XBAROUT2	FSIRX0_CLK	GPIO16		
J5.48	PR0_PRU0_GPIO6		RMII2_REF_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		

Table 9-7. Pinmux Options for J6

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
J6.51	EPWM11_A	UART2_CTSn					GPMC0_CLKLB	GPIO65		
J6.52	EPWM11_B	UART3_RTSn					GPMC0_OEn_REn	GPIO66		
J6.53	EPWM12_A	UART3_CTSn	SPI4_CS1				GPMC0_WEn	GPIO67		
J6.54	SPI1_D1	UART5_RXD				XBAROUT4	FSIRX0_DATA1	GPIO18		
J6.55	SPI1_D0	UART5_TXD				XBAROUT3	FSIRX0_DATA0	GPIO17		
J6.56	PORz									
J6.57	EPWM12_B	UART1_DCDn					GPMC0_CSn0	GPIO68		
J6.58	SPI1_CS0	UART4_TXD	LIN4_TXD			XBAROUT1		GPIO15		
J6.59	EPWM0_B							GPIO44		
J6.60	GND									

Table 9-8. Pinmux Options for J7

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
J7.61	5 V									
J7.62	GND									
J7.63	ADC2_AIN1									
	MCAN1_RX	SPI4_D0						GPIO9		
J7.64	ADC3_AIN1									
	MCAN1_TX	SPI4_D1						GPIO10		

Table 9-8. Pinmux Options for J7 (continued)

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
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_ADVn_ALE	GPIO126	SDFM0_CLK2	
J7.69	ADC3_AIN2									
	MCAN3_RX							GPIO129	SDFM0_D3	
J7.70	DAC_OUT									
	MCAN3_TX	UART5_RXD						GPIO128	SDFM0_CLK3	

Table 9-9. Pinmux Options for J8

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
J8.71	PR0_PRU1_GPIO18		UART3_TXD	PR0_IEP0_EDIO_DATA_IN_OUT31	TRC_CTL	XBAROUT14	GPMC0_WAIT1	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_D1	
J8.74	PR0_PRU1_GPIO7	CPTS0_TS_SYNC	UART5_RTSn	PR0_IEP0_EDC_SYNC_OUT1		I2C3_SDA		GPIO124	SDFM0_CLK1	
J8.75	EPWM9_A						FSITX2_DATA1	GPIO61		
J8.76	EPWM9_B	UART1_RTSn					FSIRX2_CLK	GPIO62		
J8.77	EPWM3_A							GPIO49		
J8.78	EPWM3_B							GPIO50		

Table 9-9. Pinmux Options for J8 (continued)

Pin#	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9
J8.79	EPWM13_A	UART1_RIn					GPMC0_AD0	GPIO69		
J8.80	EPWM13_B	UART1_DTRn					GPMC0_AD1	GPIO70		

9.4 ADC and DAC Mapping in Control Card for Standard Analog

The AM263x Control Card has two revisions called E1 and E2. Control Cards are only built with the superset device (AM2634). Customers can scale down by purchasing select OPN devices (AM2642 or AM2631) that meet their system requirements.

The Control Card had various design changes for the E2 revision of the board. The changes are mentioned in *AM263x Sitara Control Card Hardware User's Guide*. The differences in ADC and DAC Mapping in Control Card for Standard Analog Devices is mentioned in Table 9-10. For the Enhanced Analog Devices, see also *AM263x Sitara Control Card Hardware User's Guide.*

Table 9-10. E1 Versus E2 Pin Mapping for Standard Analog Devices

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
25	ADC1_AIN2_p	NC/ADC_CAL0
26	ADC1_AIN2_n	ADC1_AIN5
27	ADC2_AIN0_p	NC/ADC_CAL1

Table 9-10. E1 Versus E2 Pin Mapping for Standard Analog Devices (continued)

HSEC	E1	E2
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 Summary

All the AM263x family of devices are pin-to-pin compatible to each other. The differences between these devices can be identified based on their part numbers as explained in this document. The MCU_PLUS_SDK and EVMs for AM2634 are developed to be integrated to any of the subset devices. Engineers can use MCU_PLUS_SDK_AM263x and EVMs to develop and launch their projects on any of the AM263x devices.

11 References

1. Texas Instruments, *AM263x Sitara™ Microcontrollers* data sheet.
2. Texas Instruments, *AM263x Control Card Hardware* user's guide.
3. Texas Instruments, *AM263x LaunchPad* user's guide.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265