Kria KR260 Robotics Starter Kit

User Guide

UG1092 (v1.0) May 17, 2022

Xilinx is creating an environment where employees, customers, and partners feel welcome and included. To that end, we're removing non-inclusive language from our products and related collateral. We've launched an internal initiative to remove language that could exclude people or reinforce historical biases, including terms embedded in our software and IPs. You may still find examples of non-inclusive language in our older products as we work to make these changes and align with evolving industry standards. Follow this link for more information.

AMD
XILINX

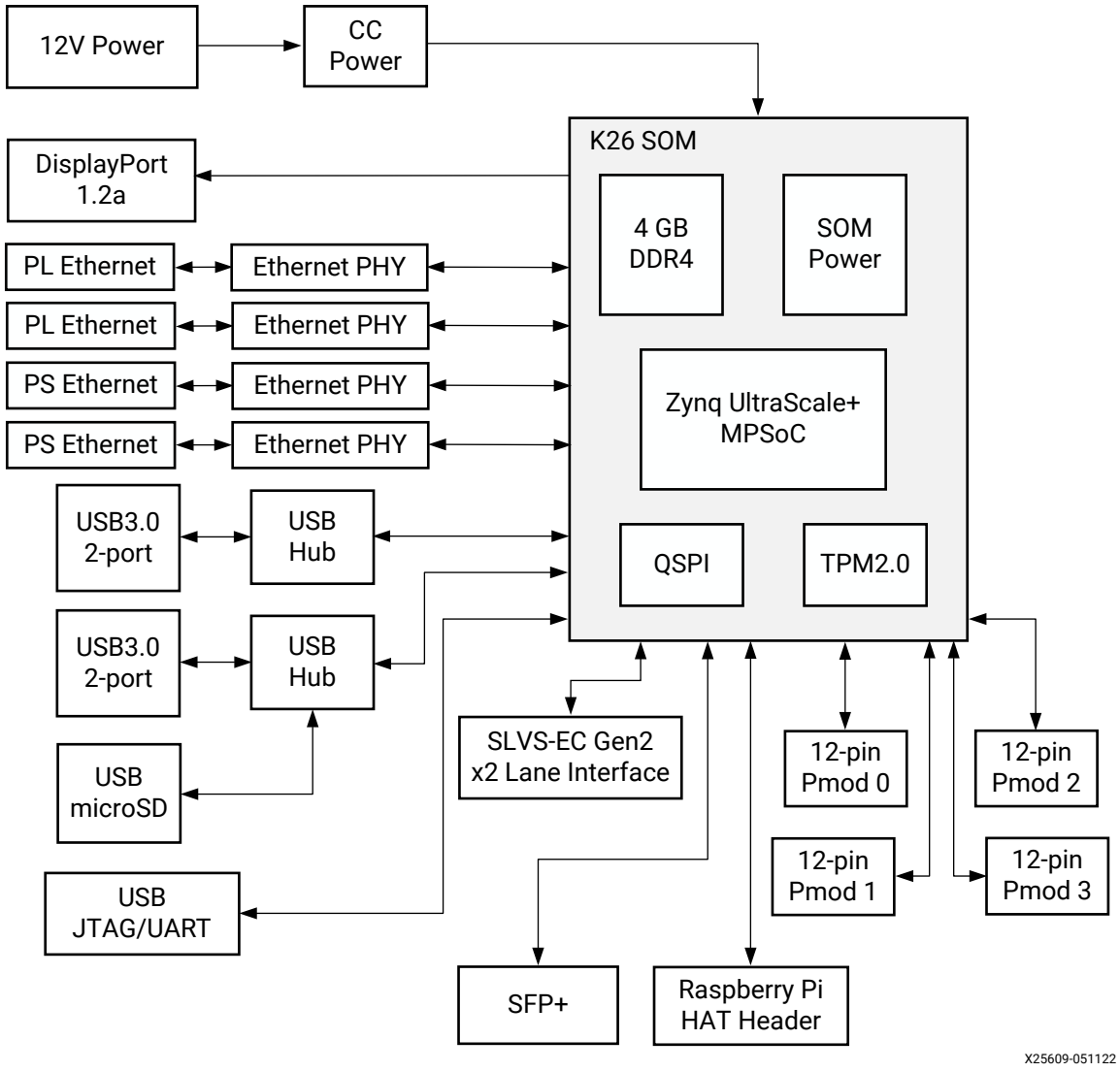
AMD
XILINX

Table of Contents

Summary

The Xilinx® Kria™ KR260 Robotics Starter Kit is comprised of a non-production version of the K26 system-on-module (SOM), carrier card, and thermal solution. The SOM integrates core digital hardware components including a Zynq® UltraScale+™ MPSoC, run-time memory, non-volatile boot devices, an integrated power solution, and a security module. The robotic-focused carrier card provides various application peripheral options including a sensor input, video display outputs, USB, SD card, Raspberry Pi HAT interface, Pmod headers, SFP+ connector, and Ethernet physical interfaces. The thermal solution includes a heat sink, heat sink cover, and fan. The Kria KR260 Robotics Starter Kit is designed to provide customers a platform to evaluate their target applications and ultimately design their own carrier card with K26 SOMs. Key target robotics applications are supported with an emphasis on industrial and automation markets.

Figure 1: **KR260 Starter Kit Block Diagram**



What's in the Box?

The Kria KR260 Robotics Starter Kit includes a K26 SOM, integrated thermal solution, and carrier card. The kit is only meant for SOM platform evaluation with the carrier card providing a variety of interfaces for integrating different peripherals. The Kria KR260 Robotics Starter Kit also includes the following accessories inside the box: power supply an its adapters, Ethernet cable, USB A-male to micro B cable, a microSD with adapter, and developer stickers. The box also includes a *Getting Started* card that directs you to the getting started web page and product page. This guide lists the Supported Peripherals that can be purchased separately.

Table 1: **Summary of Box Contents**

Line Item	Items	Quantity
1	Starter Kit (SOM with fansink thermal solution on top of robotic carrier card)	1
2	*Getting Started* card	1
3	Developer stickers	1
4	Power supply and its adapters	1
5	Ethernet cable	1
6	USB A-male to micro B cable	1
7	microSD with adapter	1

Interfaces

The following figure and table provide an overview of the physical connections, their designators, and relative position on the starter kit. The table uses the following abbreviations to indicate if a specific designator is located on the carrier card or on the SOM.

- CC = Device or interface is located on the carrier card
- SOM = Device or interface is located on the SOM

On the carrier card, there are four USB ports. USB0 and USB1 are each connected to a pair or USB physical port interfaces. There are four Ethernet interfaces with one pair connected to PS GEMs and one pair to PL-based GEMs. As shown in the following figure, GEM1 on J10C is the default firmware and software Ethernet interface used for the image recovery application and the primary Ethernet interface in Linux.

Figure 2: **Interfaces and Connectors—Top of Card**

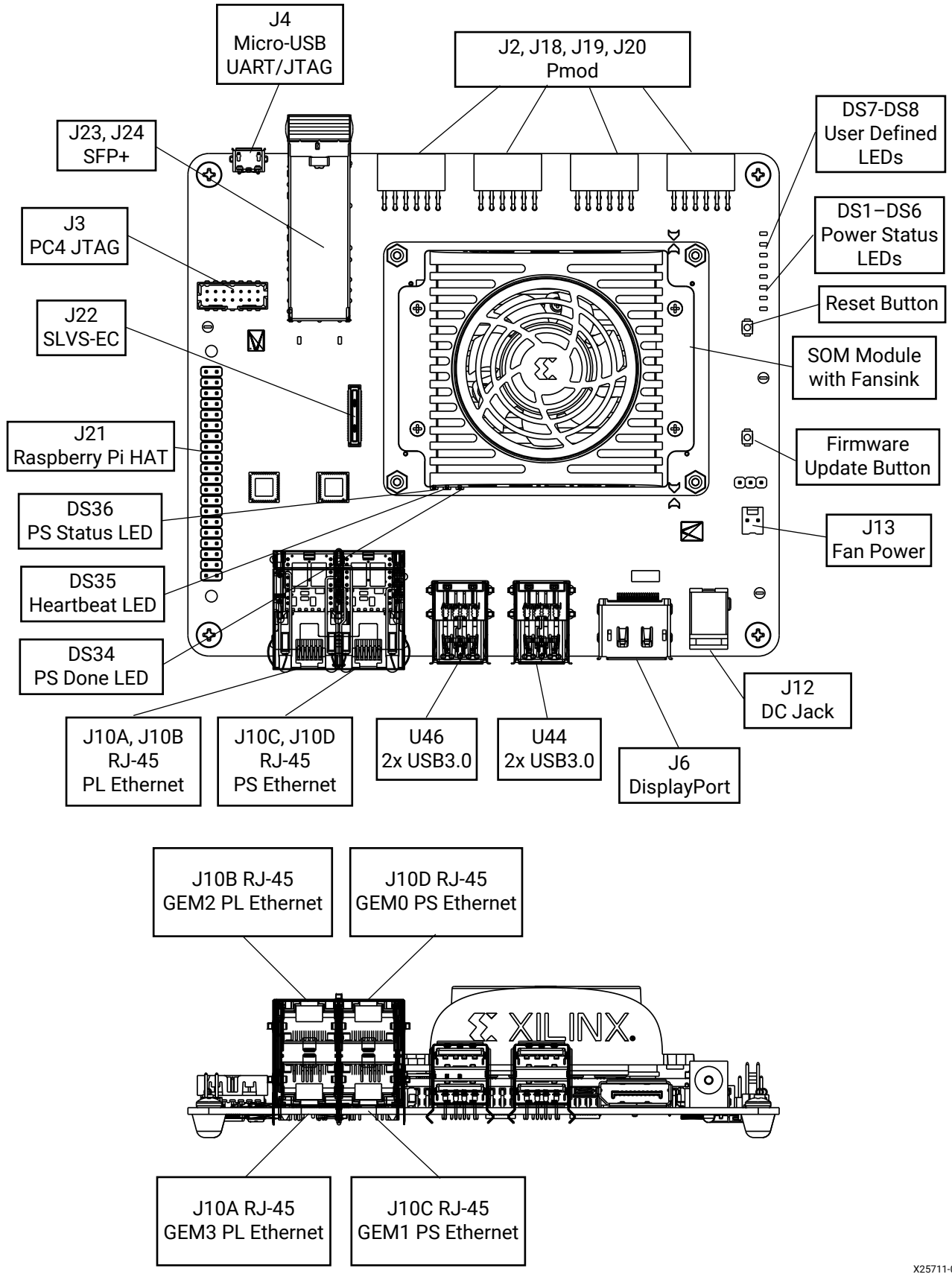


Figure 3: **Interfaces and Connectors—Bottom of Card**

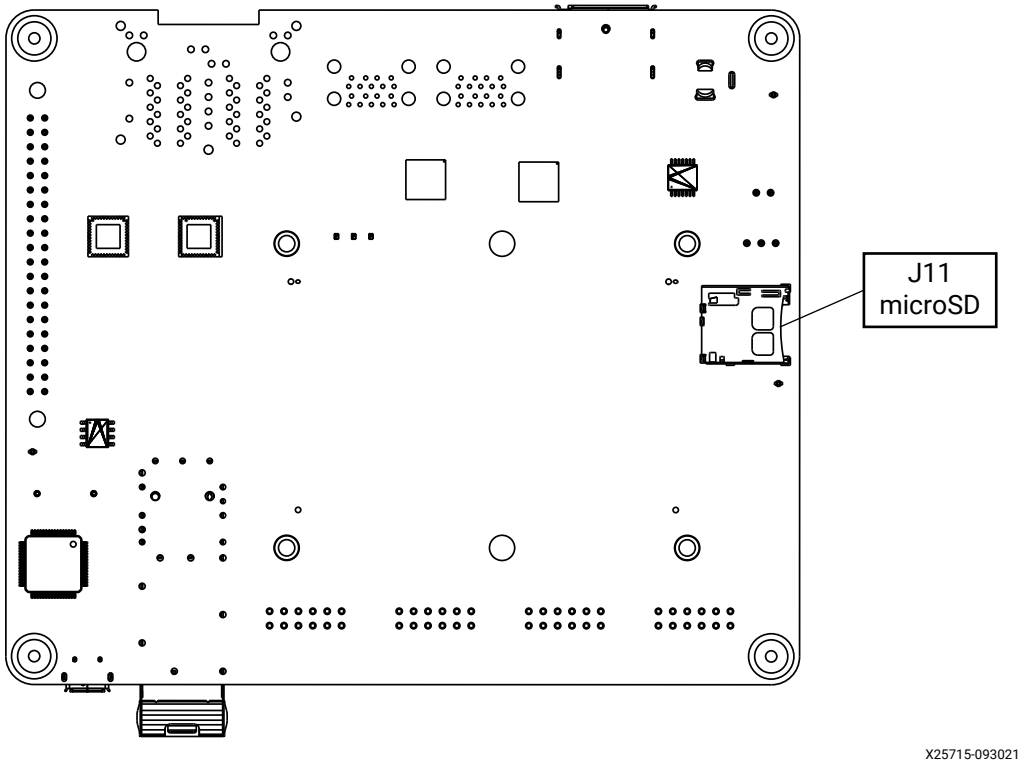


Table 2: **Descriptions and Locations**

Location	Name	Description
SOM DS34	PS done LED	Lit indicates that the PS has successfully loaded a PL design.
SOM DS35	Heartbeat LED	Periodic flashing green LED driven by the Zynq UltraScale+ MPSoC APU processor.
SOM DS36	PS status LED	Status LED, when lit indicates a successful application load.
CC DS1-DS6	Power status LEDs	Indicates various power supply and power domain status. Green LED indicates *good* status.
CC J2, J18, J19, J20	Pmod	Digilent Pmod 2x6 expansion header
CC J3	PC4 JTAG	Direct JTAG interface, bypasses the FTDI device.
CC J4	FTDI USB2.0 UART and JTAG	Integrated JTAG and device UART interface via USB2.0
CC J6	DisplayPort	DisplayPort video output
CC J10A	Ethernet RJ45 jack	1 Gb/s PL GEM3 RGMII Ethernet interface on HPB
CC J10B	Ethernet RJ45 jack	1 Gb/s PL GEM2 RGMII Ethernet interface on HPA
CC J10C	Ethernet RJ45 jack	1 Gb/s PS GEM1 RGMII Ethernet interface
CC J10D	Ethernet RJ45 jack	1 Gb/s PS GEM0 SGMII Ethernet interface
CC J11	microSD card	microSD card boot device
CC J12	12V power input	12V power input jack
CC J13	Fan power	12V SOM fan power interface
CC J21	Raspberry Pi HAT	Raspberry Pi expansion header for HAT interface
CC J22	SLVS-EC	Framos FPA SLVS-EC interface
CC J23	SFP+	SFP+ connector
CC J24	SFP+ cage	SFP+ cage
CC SW1	Firmware update button	Push button used during the boot firmware update process
CC SW2	Reset button	Push button that resets the SOM via the device POR_B signal
CC U44	USB0	Two USB3.0 or USB2.0 compatible connectors
CC U46	USB1	Two USB3.0 or USB2.0 compatible connectors

Initial Setup

Powering the Starter Kit and Power Budgets

The Kria™ KR260 Robotics Starter Kit requires a 12V, 3A power supply adapter to power the kit. The adapter is included in the kit. The power supply adapter barrel connector plugs into the DC jack (J12) to supply the +12V power source to KR260 Starter Kit.

Powering the K26 SOM

1. The KR260 Starter Kit carrier card on-board regulator generates a 5V supply and provides power to other voltage regulators.
2. The SOM power rail (V_{CC_SOM}) is powered by the 5V supply.
3. Next, the SOM on-board power-on sequencing starts.
4. The carrier card provides the programmable logic (PL) the V_{CCO} voltage rails after the SOM asserts the $V_{CCOEN\_PS\_M2C}$ and $V_{CCOEN\_PL\_M2C}$ signals.

Power Telemetry

A power monitor device is available on the SOM power rail (SOM_5V0). You can access the total power consumed by the SOM module through the I2C bus and Xilinx provided utilities.

Powering Peripherals

The KR260 Starter Kit carrier card supplies power to the I/O peripherals as specified by the following interface specifications.

USB3.0

There are four USB3.0 interface ports available on the KR260 Starter Kit carrier card. There are two independent USB controllers, but they share a power source. Each port can deliver a 5V supply to the attached I/O peripherals with up to 900 mA per port. All ports are protected against an over-current event through 1.0A power switches per pair.

Note: The total current (across all four ports) is allocated at 2.0A.

Pmod Connector

The 12-pin Pmod interface (from Digilent Inc.) is specified to be 3.3V, 100 mA. The four ports are supported by a 3.3V, 1.0A shared capacity across all connectors.

Raspberry Pi Expansion Header

The Raspberry Pi expansion header is for use with Raspberry Pi HATs. This 40-pin interface connector is supported by 3.3V and 5.0V supply voltages. There is a 1.0A limit per voltage rail.

SFP+

Pluggable SPF+ transceiver modules are supported by the SFP+ cage that provides a 3.3V, 600 mA supply budget. The total power consumed must fall within the power budget for the SPF+ module. The SFP+ power is not explicitly limited, thus care must be taken when attaching optional accessories to your carrier card.

Framos FPA SLVS-EC

The Framos FPA SLVS-EC connector is supported with two voltages. The carrier provides 1.8V at 800 mA and 3.8V with a limit of 1.0A. Because the 1.8V is not explicitly limited, care must be taken when attaching accessories to ensure a proper power budget.

microSD Card

The microSD card is supported by the 3.3V supply voltage with a power budget of 200 mA. This should be more than adequate for standard cards. It is also not explicitly limited. The starter kit supports up to 64 GB microSD cards.

Fan and Heat Sink

The KR260 Starter Kit is built with an integrated active cooling solution (see Figure 2). The integrated fansink allows you to exercise the full 10W Zynq® UltraScale+™ MPSoC application power budget without any additional accessories.

Out of the box, the 12V fan should already be plugged into the starter kit. If it is not, be sure that the fan is plugged into the connector designated in Table 2. The fan connector is keyed to ensure proper orientation.

By default, the fan runs at a constant speed. Variable fan speed control can be implemented through a FPGA based PWM fan controller. The fan gating signal is connected to a FPGA HD I/O bank pin for control. Consult the corresponding KR260 Starter Kit carrier card schematic for specific pin assignments.

AMD XILINX

Chapter 3

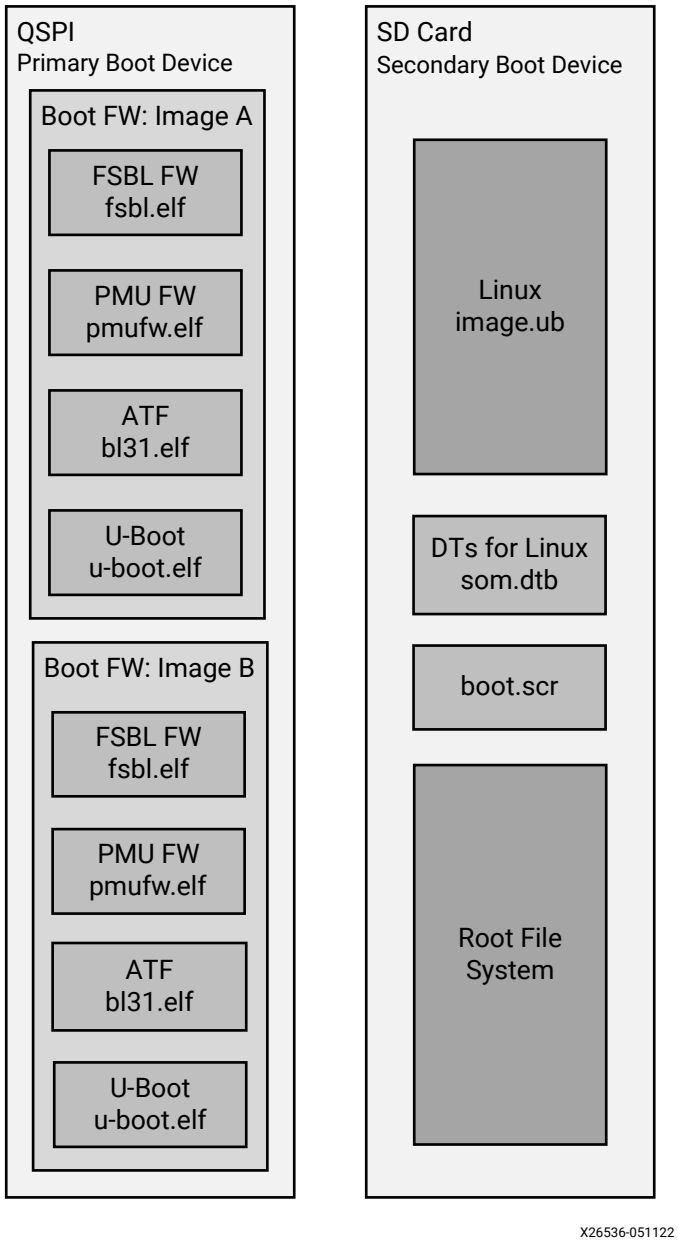
Boot Devices and Firmware Overview

The Kria™ KR260 Robotics Starter Kit has a primary and secondary boot device that provides isolation of platform-specific boot firmware from the runtime operating system and application. This allows you to focus on developing and updating your application code within the application image without having to rebuild and flash boot firmware. The primary boot device is a QSPI memory located on the SOM and the secondary boot device is an SD card interface on the carrier card. By default, the KR260 Starter Kit carrier card sets the XCK26 boot mode to QSPI32. The SOM boots up to U-Boot using the QSPI contents and then U-Boot does a hand-off to the secondary boot device.

***Note*:** You must flash the SD card image and populate the microSD card in the carrier card for the kit to successfully boot to Linux.

The overall boot device definition and firmware contents are outlined in the following figure.

Figure 4: **Boot Devices**



IMPORTANT! *Production SOMs provide both QSPI and eMMC devices on the SOM PCB to support integrated primary and secondary boot configurations.*

Primary Boot Device

The primary boot device is a QSPI device located on the SOM. The necessary elements are packaged in a Zynq® UltraScale+™ MPSoC specific format and file captured as `BOOT.BIN`. The `BOOT.BIN` file contains the board-specific boot firmware that consists of the following elements:

- **FSBL:** First-stage boot-loader firmware
- **PMU:** Platform management unit firmware
- **ATF:** Arm® trusted firmware
- **U-Boot:** Second-stage boot loader

U-Boot provides the functionality for the hand-off between the primary boot device and the secondary boot device. It will search for both the SD card and eMMC secondary boot devices; if both are detected it will provide a menu interface to you to select the desired Linux boot target.

The primary boot device provides a redundant copy of boot firmware arranged in an A/B configuration. The A/B configuration provides a dynamic primary and secondary image operation with corresponding update mechanisms. On boot, the system automatically boots from the defined primary image, and, if boot fails, it falls back to the previously known good boot image.

Secondary Boot Device

The secondary boot device on the KR260 Starter Kit is the SD card. It contains the operating system image and associated application files. The KR260 Starter Kit accelerated application references are built on the Linux operating system. The *Getting Started* webpage provides a pre-built reference image that can be written to a microSD card for out-of-the-box functionality. SOM board support packages (BSPs) are also available if you want to customize your OS.

RECOMMENDED: *The SOM is designed to use SDHC standard microSD cards. See AR66779 for a list of tested microSD cards.*

Software Getting Started

To get started with the KR260 Starter Kit, prior to powering, booting the board, and running your first application, you need to follow the instructions on the *Getting Started with Kria KR260 Robotic Starter Kit* pages to download and write the `Xilinx SOM Starter Linux` image to a microSD card. The webpage guides you to power on the KR260, boot Linux, and run a number of pre-built accelerated applications to start evaluation of the capabilities on the platform.

Platform Management Utility

The following section outlines the platform management utility called `xmutil` that is included in the SOM Linux image to help you configure and work with the SOM. The table provides a list and description of the functions available from Xilinx. You should use the `-h` or help functions with each utility to get detailed use instructions. Using *sudo* is required with many of the `xmutil` functions.

Table 3: **SOM Utility Functions**

Utility Function	Description
xmutil boardid	Reads all board EEPROM contents. Prints information summary to command line interface.
xmutil bootfw_status	Reads primary boot device information. Prints A/B status information, image IDs, and checksums to command line interface.
xmutil bootfw_update	Tool for updating the primary boot device with a new boot image in the inactive partition.
xmutil getpkgs	Queries Xilinx package feeds and provides a summary to the debug interface of relevant packages for the active platform based on board ID information.
xmutil listapps	Queries on the target hardware resource manager daemon of pre-built applications that are available on the platform and provides a summary to the debug interface.
xmutil loadapp	Loads the integrated HW+SW application inclusive of the bitstream, and starts the corresponding pre-built application software executable.
xmutil unloadapp	Removes accelerated application inclusive of unloading its bitstream.
xmutil platformstats	Reads and prints a summary of the following performance related information: CPU frequency, RAM usage, temperature, and power information.
xmutil ddrqos	Utility for changing configuration of PS DDR quality of service (QoS) settings. Initial implementation focuses on PS DDR memory controller *traffic class* configuration.
xmutil axiqos	Utility for changing configuration of PS/PL AXI interface quality of service (QoS) settings. Initial implementation focuses on AXI port read/write priority configurations.

Accelerated Applications

The Xilinx SOM platforms are enabled with a number of accelerated applications (AA) that can be dynamically installed on the SOM platform. The SOM starter Linux image is application agnostic and provides a set of utilities for pulling the hardware accelerated application examples from the SOM Linux package feeds.

Accelerated applications are software controllable, application-specific reference designs for roboticist, AI developers, embedded developers, and system architects to customize and enhance the functionality through software control or updating the AI models.

The Kria™ robotics stack (KRS) is an integrated set of robot libraries and utilities that use hardware to accelerate the development, maintenance, and commercialization of industrial-grade robotic solutions. It adopts the robot operating system (ROS) Software Development Kit (SDK) and enables a ROS 2-centric development approach that spans from the creation of computational graphs to the commercialization of ROS 2 overlay workspaces found in the Kria App Store. Some robotic accelerated applications are developed using the KRS.

The following table outlines some featured accelerated applications available for the KR260 Starter Kit. Consider this list as a starting point. Visit the SOM *Getting Started* webpage for the most up-to-date accelerated applications availability.

Table 4: **KR260 Accelerated Applications Overview**

Name	Description
ROS 2 Perception Node	Using the hardware accelerated perception for ROS 2 developers makes it easier to build high-performance solutions on Kria platforms. The focus of this accelerated application is to improve the throughput of image processing, an important facet to roboticists leveraging the Kria robotic stack (KRS) and Vitis Vision Library. The accelerated application reduces the load on the host CPU while providing significant performance gain. The data source and virtual environment are provided by the Gazebo simulator with Ubuntu Linux 22.04.
ROS 2 Multi-Node Communications via TSN	The ROS 2 multi-node communication via TSN is an accelerated application that focuses on applying ROS 2 within the context of an time-sensitive networking (TSN) based communications infrastructure that is developed using the Kria robotics stack.
10GigE Vision Camera	The 10GigE vision camera is a hardware accelerated machine vision application that is used in defect detection on the KR260 platform. It uses the Framos SLVS-EC sensor interface, the Euresys 10GigE vision interface, and the defect detection algorithm using the Vitis Vision Library.

Accelerated Application Package Selection

RECOMMENDED: *Public Ethernet connectivity is necessary to dynamically pull down the latest accelerated application designs.*

1. If you have not already verified Internet connectivity do so before proceeding via ping test or DNS lookup (e.g., nslookup).
2. Example applications are deployed using the Linux package management framework for over-the-air deployment. The `sudo xmutil getpkgs` lists a series of package groups that can be installed on your platform. The package group naming convention is: `packagegroup-kit_name-application_name`. For example, the machine vision application for the KR260 platform has the following package group name `packagegroup-kr260-machine-vision`. You can install any number of matching accelerated applications to your platform. For exact commands to install, refer to the Kria SOM Wiki for further details.

   ***Note*:** You should only install package-groups that are compatible with your particular starter kit configuration.
3. For any applications installed on the local file system via the package feed, the platform can now dynamically load and swap those applications. To see a list of the applications local to the system, execute `sudo xmutil listapps`. You can also see what applications are local by manually exploring the `/opt/xilinx` directory.
4. By default, kr260-dp is loaded on boot. From the applications list, check for an active application loaded (active = 1 in the `xmutil listapps` output). If there is a loaded application, unload it by running the `sudo xmutil unloadapp` command to unload the current application before proceeding to the next step.
5. From the application list, start the new application by running `sudo xmutil loadapp application_name`. The platform configuration is automatically handled and starts the application.
6. Applications with a Jupyter-based cockpit will start-up automatically. You need to point your web-browser to the associated IP address and port. The associated IP address and port information is printed to the UART at boot. To query your Jupyter lab server URL after the initial boot, run: `sudo jupyter notebook list`.

Supported Peripherals

The following table outlines external peripherals that are tested with the corresponding accelerated applications. It is recommended that you use a peripheral from the list to ensure that you realize maximum platform performance.

Table 5: **Accelerated Application Peripherals**

Accelerated Application	Peripheral	Part Number
10GigE Vision Camera	Sony IMX547 camera kit color for SLVS_EC (J22)	Sony HW-IMX547C-SK-G
	Sony IMX547 camera kit monochrome for SLVS_EC (J22)	Sony HW-IMX547M-SK-G
	10GBase-SR SFP+ transceiver (J23)	10Gtek AXS85-192-M3

The following table outlines external peripherals that are functionally verified with the KR260 Starter Kit.

Table 6: **KR260 Starter Kit Functionally Tested Peripherals**

Peripheral	Part Number
Pmod RS485 isolated communications	Digilent 410-310
USB 4K camera	Logitech BRIO

For a complete list of KR260 compatible accessories, refer to the Kria SOM Forum.

Xilinx Tools Integration

The K26 SOM and KR260 Starter Kit are integrated with the Vitis™ software development platform and Vivado® Design Suite for rapid development of your unique applications on the platform.

Vivado Board Flow

The K26 SOM is enabled in the Vivado Design Suite through the Vivado Board Flow functionality. Vivado Board Flow enables a level of hardware abstraction that automatically configures peripherals fixed on the SOM card (e.g., DDR4), defines associated timing constraints, and presents the customizable physical I/O available on the SOM connector(s).

The Vivado SOM board model is available through the Vivado installation process as well as on the Vivado board file GitHub repository. The following KR260 related Vivado board files are available.

- **KR260 Starter Kit:** Configured K26 SOM with Robotics Starter Kit companion card
- **SM-K26-XCL2GC:** K26 commercial grade production SOM
- **SM-K26-XCL2GI:** K26 industrial grade production SOM

The Xilinx SOM board flow infrastructure provides starter kit carrier card awareness through the Vivado tools *companion card* mechanism. Automation for I/O connection and peripheral IP configuration when selecting a SOM and an associated carrier card, such as the KR260 Starter Kit, is used to create a hardware configuration.

For additional information on using the Vivado tools and the Vivado board flow, refer to the *Vitis Model Composer User Guide* (UG1483).

AMD XILINX

Chapter 6

Board Reset, Firmware Update, and Recovery

This section outlines the update and recovery mechanisms built into the KR260 Starter Kit. Two tools are provided for firmware updates. The first is a Linux based A/B update tool that supports remote and redundant firmware updates to the A/B firmware partitions of the QSPI device with custom or Xilinx provided updates. The second tool is the Ethernet recovery tool that is intended to be used only when recovering a full platform to the original factory firmware.

Firmware Update

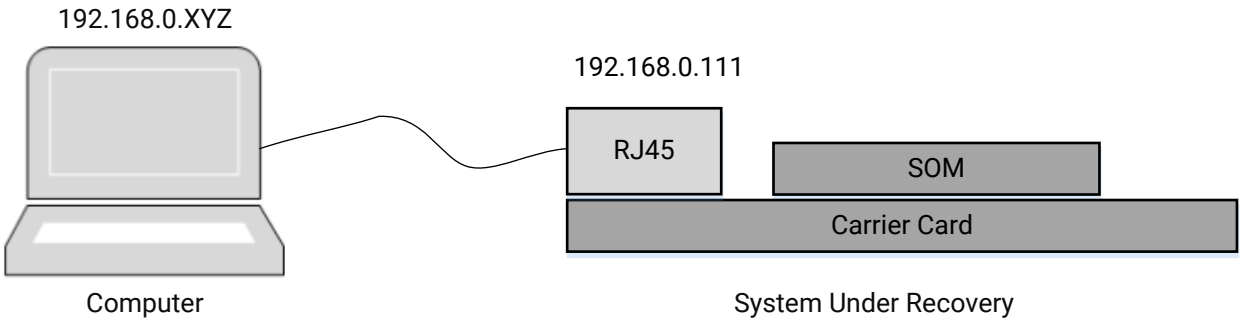
The firmware update button is the physical SW1/FWUEN push button located on the KR260 Starter Kit carrier card. The button is used to support two features:

1. To force the platform into a recovery mode application during power-on. The recovery application is described in the Ethernet Recovery Tool section of this document.
2. Security mechanism to prevent remote update of the boot firmware, without the user being physically present.

Ethernet Recovery Tool

The Ethernet recovery tool is a small application included in the Xilinx provided KR260 Starter Kit QSPI image. It provides a simple Ethernet-based interface and application for updating the boot firmware. This application and interface is initiated by holding the firmware update button during the power-on sequence. The application uses a fixed IP address of 192.168.0.111. The following figure shows an overview of the set-up.

Figure 5: **Ethernet Recovery Tool Setup**

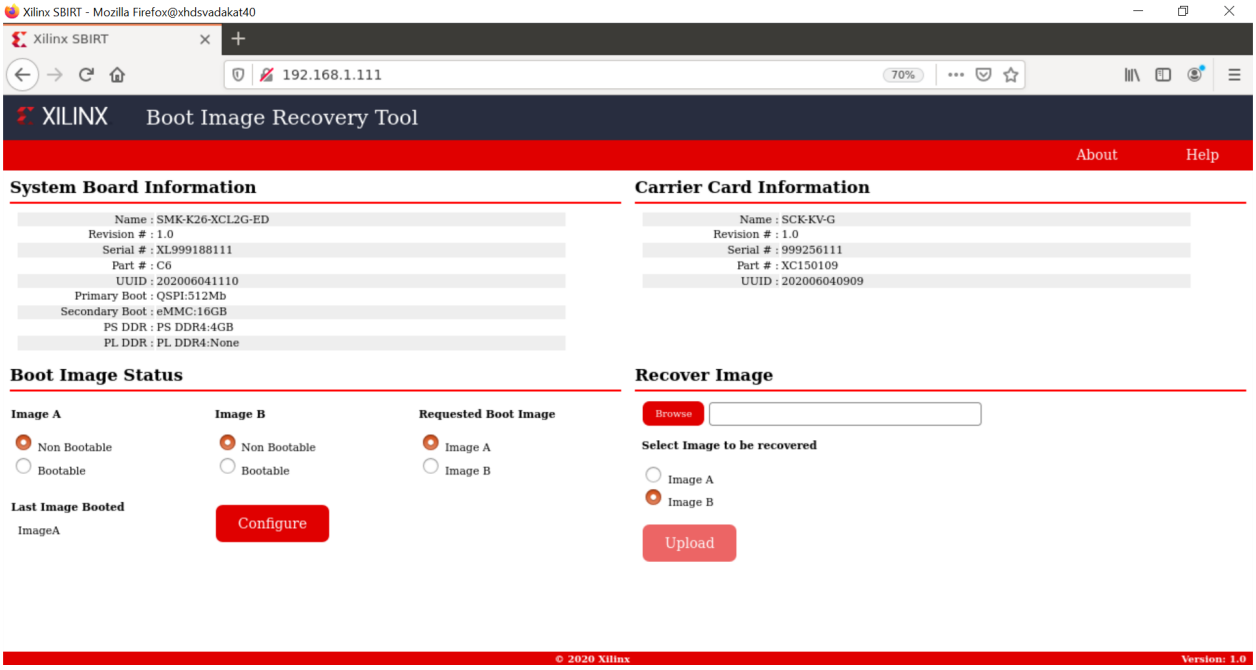


The Ethernet recovery tool provides a mechanism for updating either of the dynamic boot partitions within the primary boot device if Linux is not functional. If Linux is functional, the recommendation is to update the boot firmware using the `xmutil` boot firmware update utilities. The associated update content is a Xilinx XCK26 binary boot image captured as `BOOT.BIN`. To support platform recovery, the KR260 Starter Kit factory `BOOT.BIN` image is made available on the Xilinx SOM *Getting Started* web page. You can also use this tool when customizing the platforms boot firmware with your own `BOOT.BIN` generated through the Xilinx Vitis and PetaLinux tools.

To use the Ethernet recovery tool, follow these steps:

1. Connect the PC to the KR260 Starter Kit via Ethernet as shown in Figure 5. Ensure that the system is connected to the KR260 J10C Ethernet port. The other ports will not work for platform recovery.
2. Set the PC to a static IP address that is on the same subnet as the recovery tool (192.168.0.XYZ), but not 192.168.0.111.
3. Hold the firmware update button when powering on the device. You should also see the UART print outs from the recovery application.
4. Use a web-browser (e.g., Chrome or Firefox) on the PC to navigate to the URL `http://192.168.0.111` for access to the Ethernet recovery tool.
5. Use the Ethernet recovery tool GUI in the web-browser to update either the A or B boot firmware partitions with a `BOOT.BIN` file from the file system on the PC. The Ethernet recovery tool interface is shown in the following figure.

Figure 6: **Ethernet Recovery Tool Interface**



Boot Firmware A/B Update

As outlined in the Chapter 3: Boot Devices and Firmware Overview section, the Xilinx starter kit and SOM provide two copies of the boot firmware in the QSPI device. This mechanism has a robust update infrastructure through a ping-pong methodology, where the last known good boot image is always kept available in the platform. The Xilinx starter kit Linux provides a utility for doing these updates entirely on-target. The steps for using the Xilinx SOM A/B update tool are:

1. Place the new `BOOT.BIN` in the Linux file system.
2. Execute the A/B update process through these steps:
   a. Go to the directory where you copied the `BOOT.BIN` in step 1.
   b. Type `sudo xmutil bootfw_update <location of new BOOT.BIN>`.
   c. The tool returns the image (A or B) that is updated, and is marked for boot on the next boot.
   d. You can verify the updated status of the boot firmware using the `sudo xmutil bootfw_status` utility.
3. Power cycle the device or press the board reset button.

   To access the latest boot firmware functionality for starter kits, find the latest BOOT.BIN on the Kria SOM wiki page and do a firmware update to latest boot firmware.

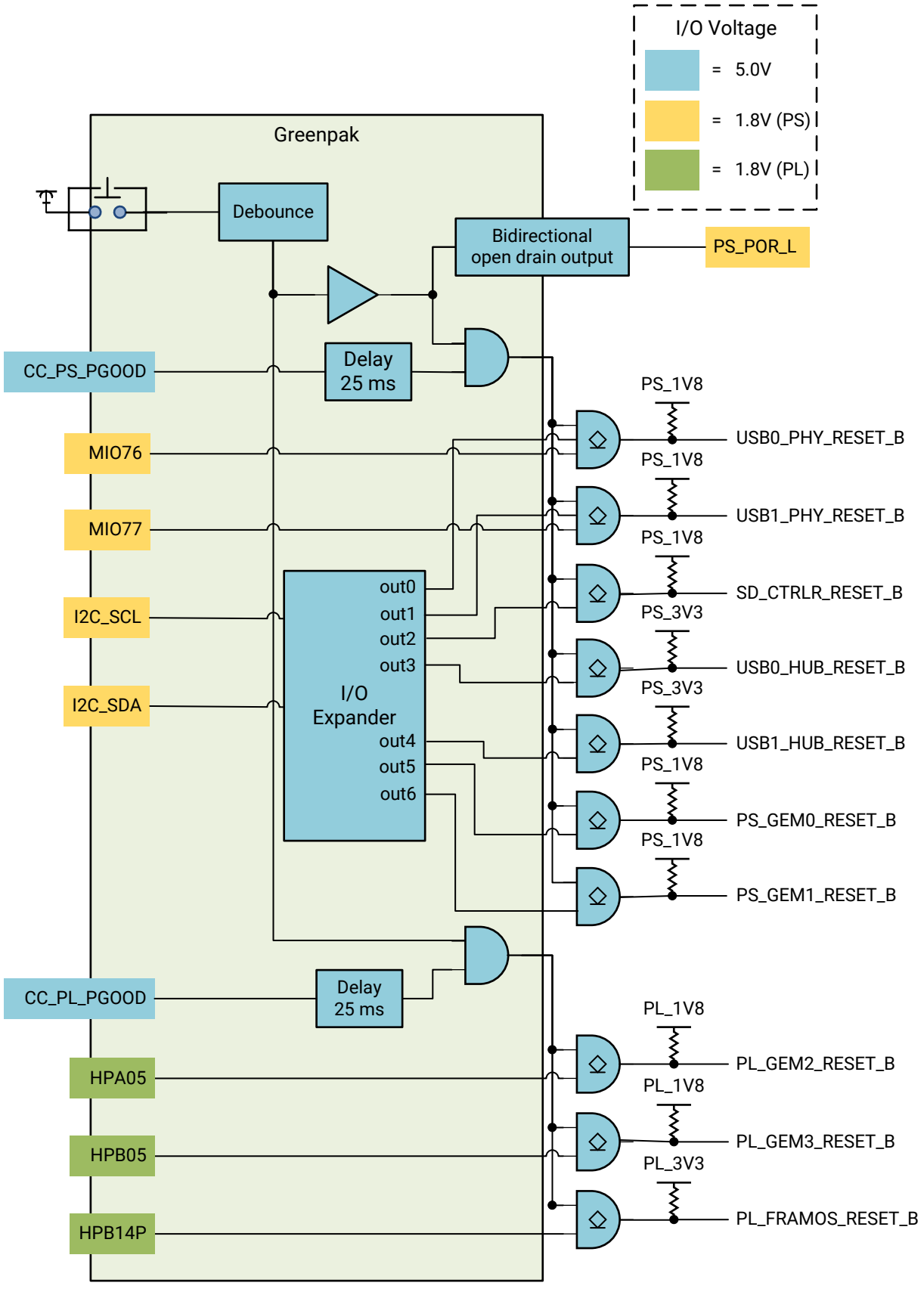
Board Reset

Power-on Reset

1. The SOM reset signal (PS_POR_L) is held in reset until the CC_PS_PGOOD signal is asserted on the carrier card.
2. All the PS and PL I/O device reset signals on the carrier card are held in reset until 25 ms after the PS and PL power domain are powered up and stable.
3. You can perform a hard reset on the KR260 Starter Kit by pressing the reset button (CC SW2) or by commanding a reset through software.

The following figure shows the reset functions supported on the KR260 Starter Kit.

Figure 7: **Device Reset**



X25670-082421

AMD XILINX

Appendix A

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator (DocNav) provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open DocNav:

- From the Vivado® IDE, select **Help → Documentation and Tutorials**.
- On Windows, select **Start → All Programs → Xilinx Design Tools → DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In DocNav, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

***Note*:** For more information on DocNav, see the Documentation Navigator page on the Xilinx website.

References

These documents provide supplemental material useful with this guide:

1. Kria SOM GitHub.io documentation
2. *Kria KR260 Robotics Starter Kit Data Sheet* (DS988)
3. *Kria K26 SOM Data Sheet* (DS987)
4. *Kria SOM Carrier Card Design Guide* (UG1091)
5. *Vitis Model Composer User Guide* (UG1483)
6. *Vitis Unified Software Platform Documentation: Application Acceleration Development* (UG1393)

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
5/17/2022 Version 1.0	
Initial release.	N/A

Please Read: Important Legal Notices

www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

Copyright