Adaptive computing with the AMD Kria™ Robotics Starter Kit

ADIUVO
ENGINEERING AND TRAINING, LTD.

Technology. Passion. EBV.
ebv.com

EBVElektronik
| An Avnet Company |

Adaptive computing with the Kria Robotics Starter Kit

Robotics plays an increasing part in modern life across a broad spectrum of applications, from picking-and-packing robots in automated warehouses to surgical robots used in delicate operations.

Over the last 20 years, robots have moved from industrial and educational applications to the mainstream. However, that foundation of industrial robotics remains critical to modern manufacturing processes while the scope of industrial applications increases. Industrial robotics is no longer limited to production lines or areas where chemicals or other contaminants might be hazardous to health. Robots are also used in harvesting, pharmaceuticals, inspection and logistics on the production floor.

Robotics can be deployed for a range of use cases in configurations best suited for the application. However, this often leads to limited robotic solutions as they are designed for a specific use case. For example, a typical industrial robot that spray paints assembled components needs several axes of freedom to manipulate its end effector. As such, it often looks like a human arm. However, a logistics robot which moves heavy modules on a production floor moving in X, Y and Z directions would be implemented in a different manner. Because of this, industrial robots can be grouped into one of several categories:

- **Articulated robot**: This is the most common type of robot arm, providing several degrees of freedom and enabling a wide range of movement.
- **Cartesian robot**: These have three prismatic joints, each of which provides one degree of freedom and they are often provided by hydraulic or pneumatic cylinders. They also have three rotary joints to enable the orientation in space. These robots are often called X, Y, Z robots.
- **Cylindrical coordinate robot:** These robots contain one rotary joint at the base and one or more prismatic joints. This enables vertical and horizontal movement by sliding.
- **Sphere coordinate robot:** These robots have rotary joints and are commonly used in casting and plastic injection molding.
- **SCARA robot**: Selective Compliance Assembly Robot Arm robots have movement in the X and Y planes and are often used for assembly applications.
- **Delta robot:** This robot has multiple arms connected to the base and end effector. Delta robots are often used in pick-and-place applications.

As demonstrated by the classes of industrial robots, robotic solutions are a complex system of systems which combine several engineering disciplines, including electrical, electronic, control, mechanical and motors, and pneumatics / hydraulics depending upon the application. Developing such systems can be time consuming and complex, especially when it comes to the creation of the final application in addition to the control of the motors / drives and actuators. These applications need to be able to work out the kinematics of the solution to ensure the end effector is in the position required. This requires complex software solutions and simulation.

Robotic Operating System

To aid developers with the creation of the application and modelling of the robotic behavior, the Robot Operating System (ROS) was developed by research students at Stanford University. From its beginnings in 2007, ROS evolved to become one of the main operating systems used by roboticists. ROS enables not only control of robots and robotic systems, but also simulation and modelling of robotic systems, enabling the operation of the robot to be verified before it is deployed. Although named Robotic Operating System, ROS is a software development framework which typically runs on an embedded operating system such as embedded Linux. To enable accelerated development, the ROS framework provides developers with functionality such as hardware drivers, robot models, datatypes and support for perception and location mapping (SLAM). ROS also provides a series of tools that aid in the development or operation of the system such as Rviz, which provides a 3D visualisation, and Gazebo, a simulator.

ROS is architected around a graph architecture. Within this architecture, processing takes place in nodes that can receive and post data about the node such as sensor, control, planning, actuator positioning or current state. Nodes are connected on the ROS graph by topics. The topics are communication pipelines to which nodes can publish data and receive information.

Along with nodes and topics, a node may also advertise services. The services have a single result, such as capturing a frame of video, sampling a sensor or opening an actuator. The most used version of ROS is now ROS 2, which updated the ROS frameworks and tools to work with a wider variety of environments and provide support for real-time environments along with significant API updates. The first ROS 2 release, Ardent Apalone, was in December 2017. The latest release is Humble Hawksbill.

While frameworks such as ROS and ROS 2 offer considerable help to the robotic developer, there are still several challenges that must be addressed before higher level frameworks can be applied.

Challenges of robotics

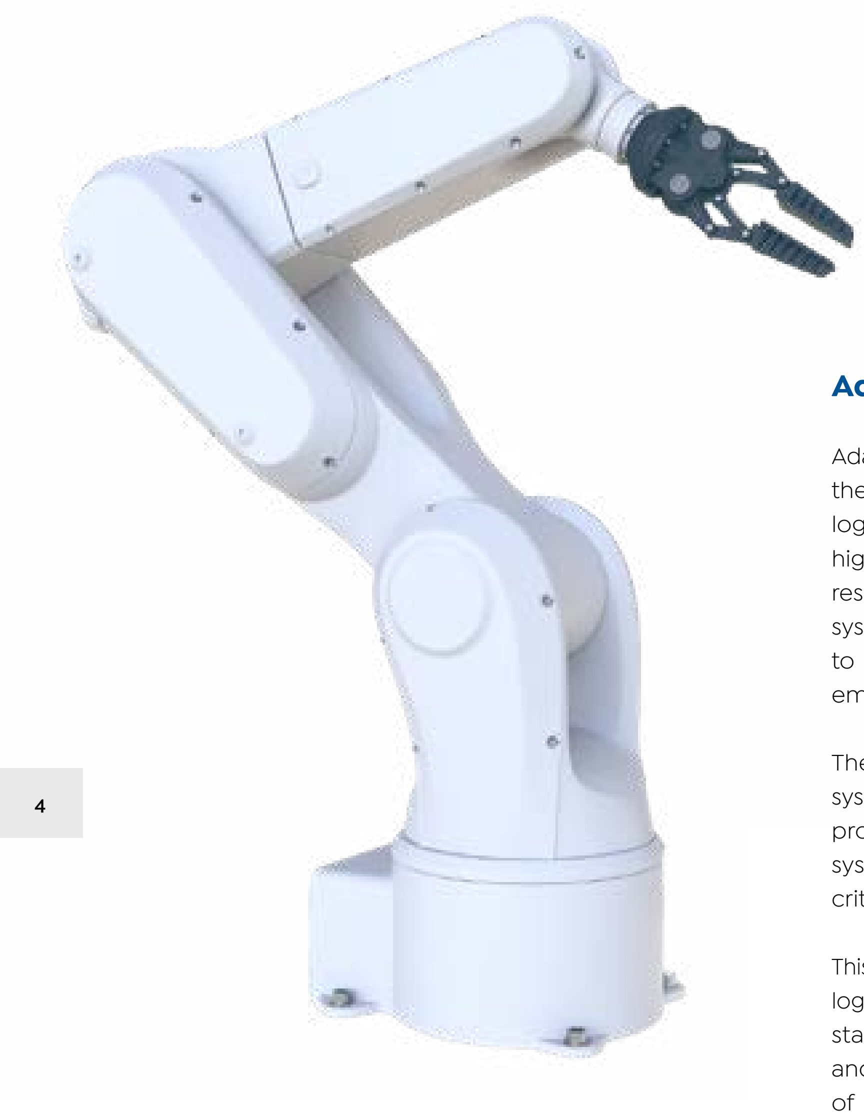
These challenges cover a wide range of topics, from complexity of application to safety / security, connectivity and power. Increasingly, robotic applications are required to work with a range of diverse sensors, such as image sensors and LIDAR / RADAR. Processing data from these sensors and running higher-level processing such as AI/ML inference and still achieving the desired response time to enable safe operation is a challenge. A common approach to address this issue is to deploy multiple high-performance processors. These high-performance processors may enable the processing needs to be achieved. However, they will also impact the overall power budget, reducing the battery life if the robot is mobile or increasing power costs if the robot is static.

Robots are also a system of systems, which means not only do they have to connect to the outside world via standardized interfaces such as Ethernet, EtherCAT, industrial networks (such as ModBus, RS485 and RS422), but they must also be able to interface with a wide range of sensors, actuators and drives to control the robot's position. This requirement brings an interfacing challenge for the system overall, as high-performance processors often do not offer the range of interfaces required. This often leads to the development of a co-processor, especially if bespoke or legacy interfaces are required.

On top of the performance and interfacing challenges, robots also face significant safety and security challenges. As the actuators and arms can cause serious damage, harm, injury or death, functional safety becomes critical to ensure that any failure is handled in a safe manner.

Similarly, the robotic solutions must also be secure from malicious actors who may attempt to impact the behaviour of the robot.

Addressing these obstacles can be a significant challenge. However, adaptive computing has you covered. With a combination of programmable logic and high-performance processors, systems can be designed to take advantage of the parallel nature of programmable logic as well as the sequential nature of high-performance processors.



Adaptive compute solutions for robotics

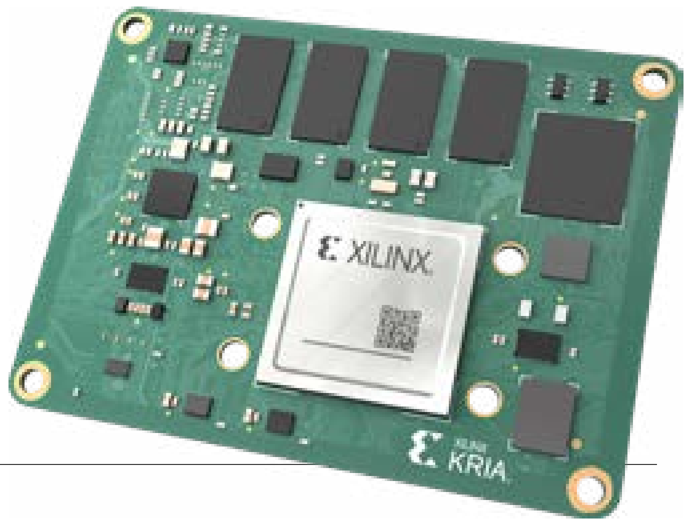
Adaptive computing solutions enable engineers to leverage the key capabilities of each element. The programmable logic provides the developer with the ability to implement high-performance parallel processing pipelines, which are responsive and deterministic, as there is no competition for system resources. The processing system has the capability to execute operating systems and frameworks such as embedded Linux and ROS 2.

The use of an adaptive compute solution also enables the system to be architected such that the programmable logic provides acceleration of functions from the processing system and provides deterministic processing for safety-critical functions.

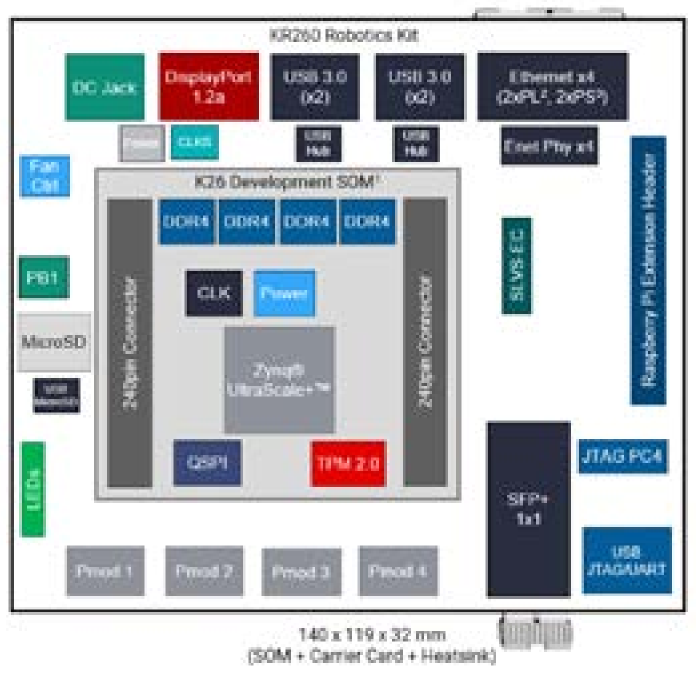
This combination of the processing system and programmable logic also provides a wide interfacing capability with industry-standard interfaces such as Gigabit Ethernet, UART, SPI and I2C. The programmable logic provides a wide range of interfacing solutions due to its flexible I/O structure, which enables any-to-any interfacing with the right external physical layer if voltage translation is required. This frees the solution from interfacing constraints and enables the support for bespoke and legacy interfaces.

When it comes to implementing an adaptive computing solution, developers have several choices. They can implement a chip-down solution or use a System on Module (SOM). A chip-down solution requires the developers to design the AMD Xilinx Adaptive compute device directly into their hardware design. This requires consideration for the power, clocking and memory architectures along with interfacing. A SOM approach uses a module, which includes the AMD Xilinx adaptive compute devices, but also the power, clocking and memory architecture. On a SOM, the adaptive compute devices I/O are broken out to a connector, which is mated with a carrier card to provide the interfaces to the robotic systems.

Developing using a SOM significantly reduces technical risk and reduces the overall development time of the application. The AMD Xilinx Kria SOM is an ideal adaptive compute platform, as it provides a high-performance heterogeneous SoC, volatile and non-volatile memories along with providing a range of I/O through dual connectors.



To aid development of robotic solutions, AMD Xilinx also offers the Kria™ KR260 Robotics Starter Kit, which provides not only the Kria K26 SOM, but also a carrier card that contains many standard interfaces used in the development of robotic systems. The Kria Robotics Starter Kit therefore enables development of robotic applications while custom carrier cards are being developed, reducing development timescales and technical risk.



The Kria Robotics Starter Kit offers the following interfaces across the 480 pins provided by the Kria SOM.

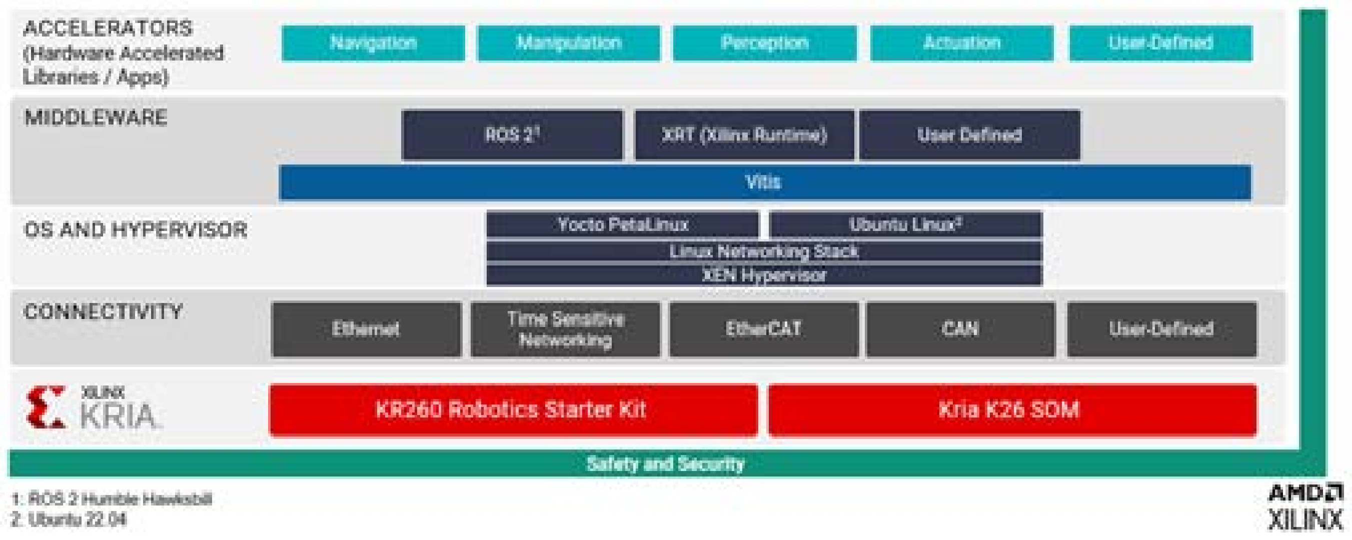
- DisplayPort 1.2
- Four USB 3.0
- Dual Ethernet connected to the processor system
- Dual Ethernet connected to the programmable logic
- Four Pmod interfaces
- 1 SFP+ cage
- RPI header
- SLVS-EX Rx
- USB UART / JTAG

This Quad Ethernet capability with connections to both the programmable logic and the processing system enables the developer to implement time-sensitive networking if deterministic communications are required.

The Kria Robotics Starter Kit runs the Kria Robotics Stack (KRS), which uses Ubuntu 2022.04 and ROS 2 Humble Hawksbill to create applications to be run on the processing system and accelerated or safety critical functions can be deployed on the programmable logic and integrated into the ROS 2 stack.

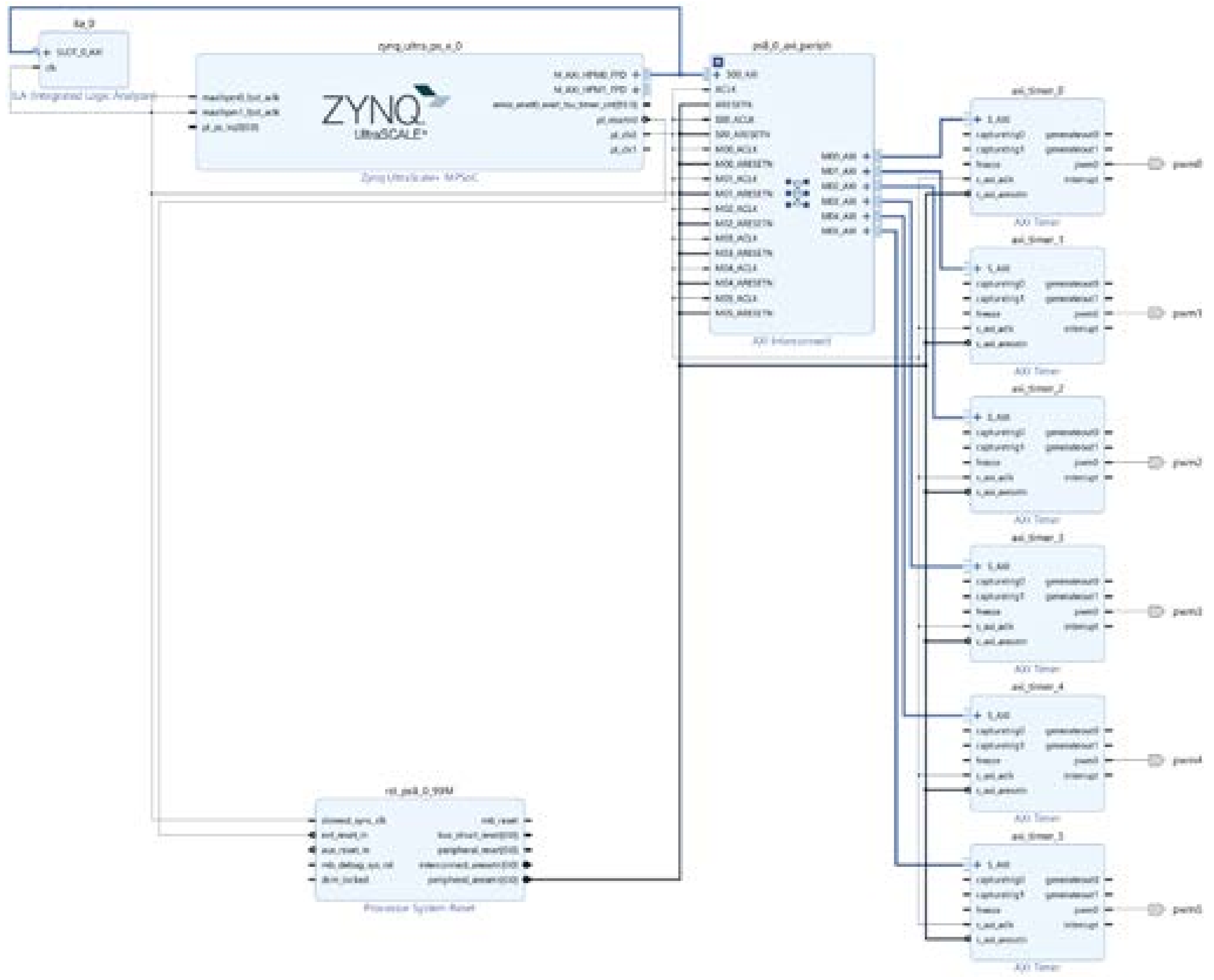
The Kria Robotics Starter Kit and KRS enable developers to create higher performance robotics solutions using familiar frameworks that address the challenges faced by modern robotic systems development.

One of the benefits of using an adaptive compute module is the programmable logic, which can be used to generate sensor drives and feedback to support a range of applications. The contents of the programmable logic can then be connected to the higher-level ROS 2 system to enable control.



Robotic arm example

Taking the example of a simple robotic arm which is servo controller or positioning, the Kria Robotics Starter Kit can be used to create the necessary pulse width modulation (PWM) drive signals. In the case of the robot arm, six joints are required for a full range of movement. The programmable logic of the Kria Robotic Starter Kit enables the implementation of six timers in the programmable logic, which can generate the PWM waveform to position the sensors.



When it comes to interfacing, the Pmod interfaces can be used in conjunction with a Pmod Con3, which provides a range of servo connections. This is important, as the servos typically run at a higher voltage and current than a Pmod port can provide. In this application, the 5V is provided by an external power source to the Pmod Con three and hence the servos.

The application software for the robotic arm can then be developed and executed within the processing system. In the initial version of software, it is a simple command line connected over a UART or Ethernet link enabling the verification of the robotic arm movement and positioning. Once the application is verified, the software control algorithms can be wrapped and created as ROS 2 nodes, which can be used with ROS 2 to create the overall robotic solution.