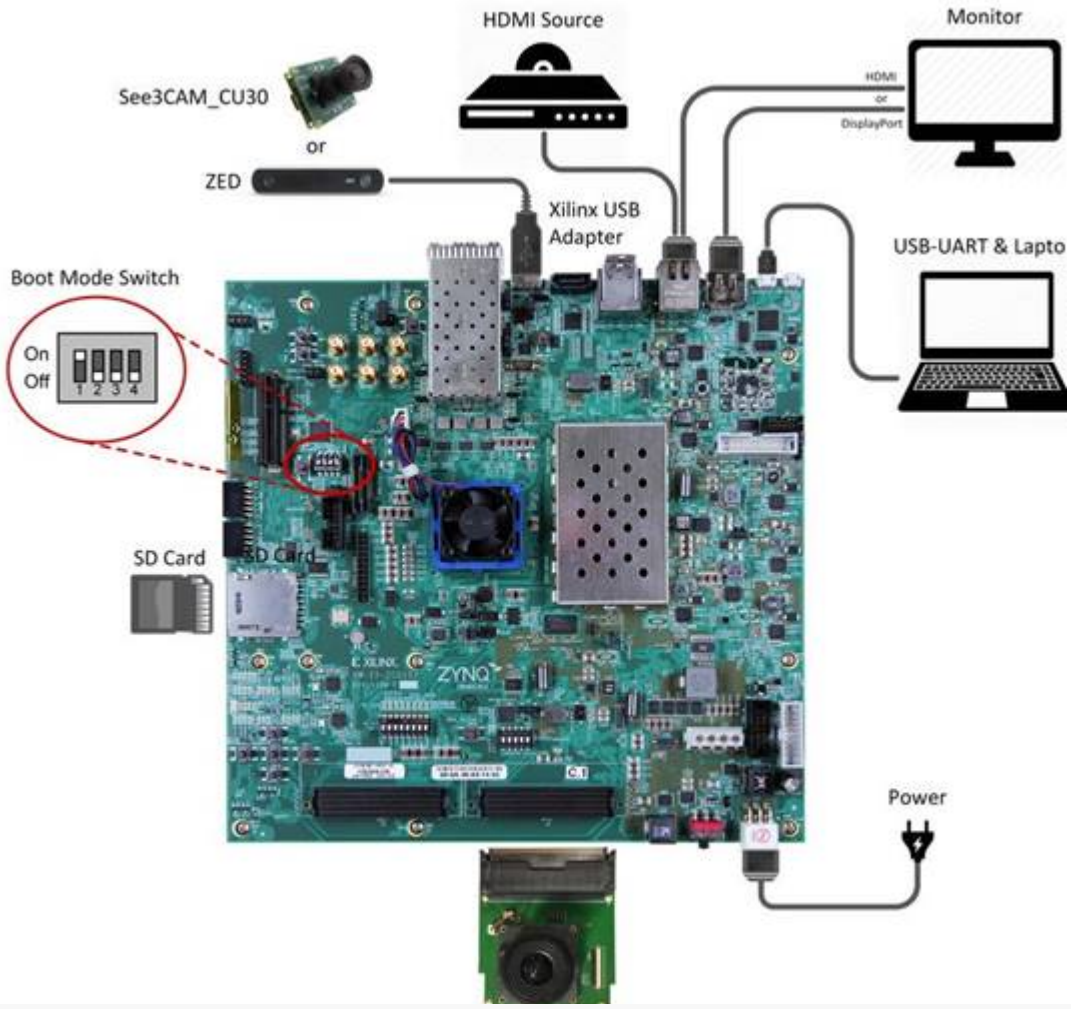
Setting Up the ZCU102/ZU4 Evaluation Board

The Xilinx ZCU102 evaluation board uses the mid-range ZU9 Zynq® UltraScale+™ MPSoC to enable you to jumpstart your machine learning applications. For more information on the ZCU102 board, see the ZCU102 product page on the Xilinx website: https://www.xilinx.com/ products/boards-and-kits/ek-u1-zcu102-g.html.

The main connectivity interfaces for ZCU102 are shown in the following figure.

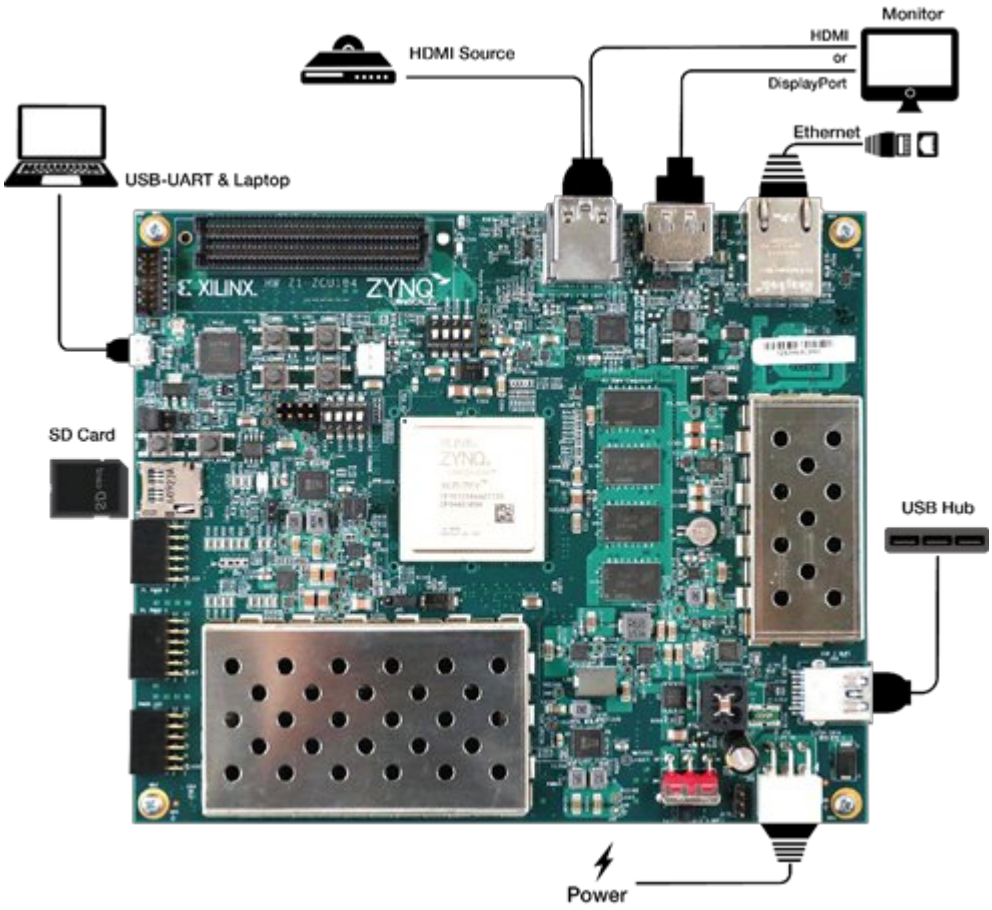
Figure: Xilinx ZCU102 Evaluation Board and Peripheral Connections



your machine learning applications. For more information on the ZCU104 board, see the Xilinx website: https://www.xilinx.com/products/boards-and-kits/zcu104.html.

The main connectivity interfaces for ZCU104 are shown in the following figure.

Figure: Xilinx ZCU104 Evaluation Board and Peripheral Connections

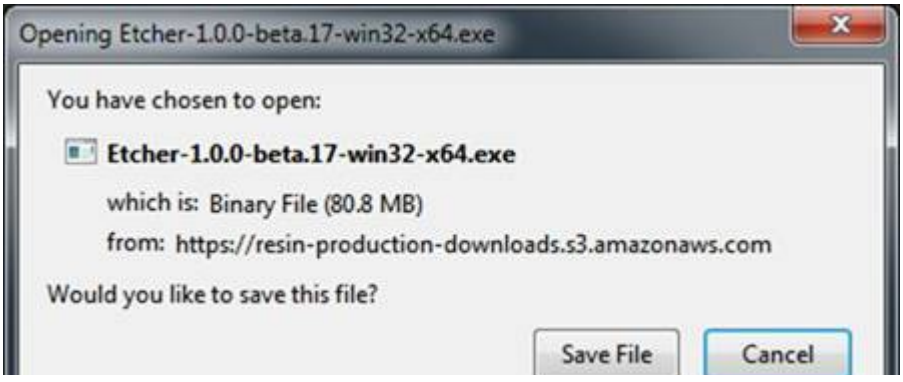


In the following sections, ZCU102 is used as an example to show the steps to setup the Vitis AI running environment on the evaluation boards.

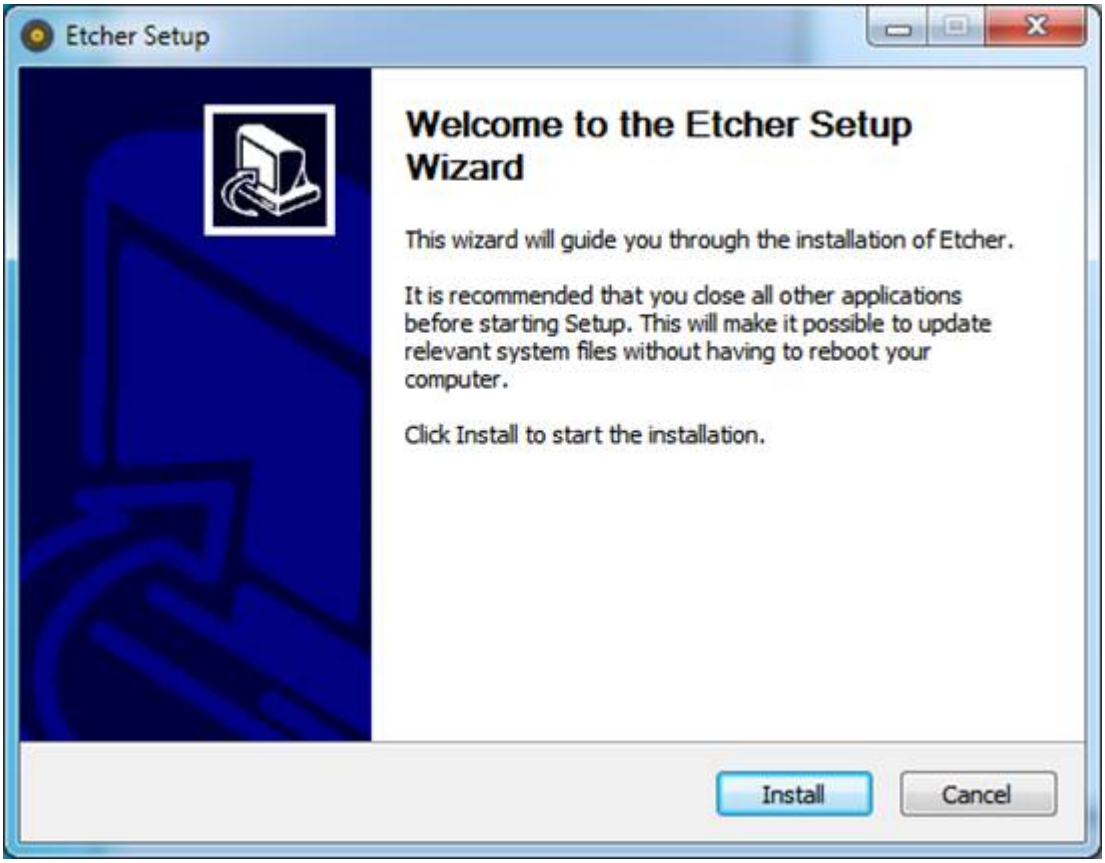
Flashing the OS Image to the SD Card

For ZCU102, the system images can be downloaded from here; for ZCU104, it can be downloaded from here. One suggested software application for flashing the SD card is Etcher. It is a cross-platform tool for flashing OS images to SD cards, available for Windows, Linux, and Mac systems. The following example uses Windows.

1. Download Etcher from: https://etcher.io/ and save the file as shown in the following figure.

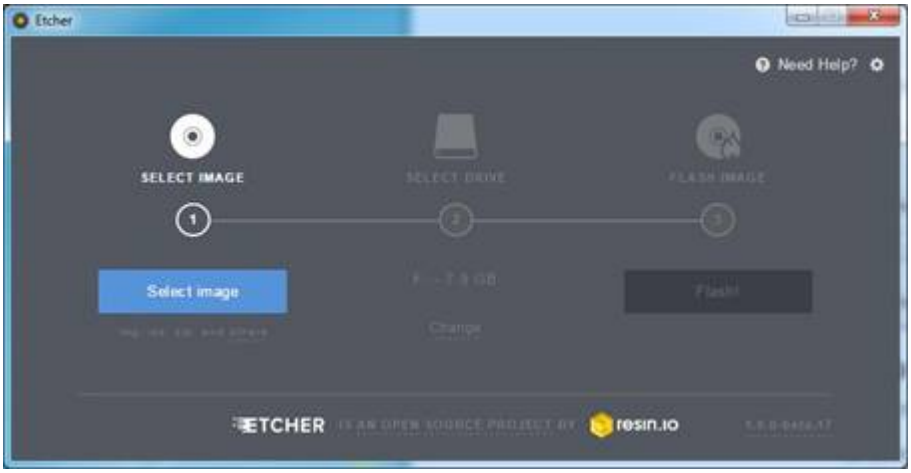


2. Install Etcher, as shown in the following figure.

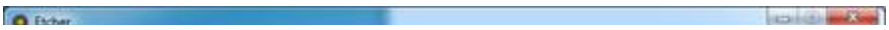


3. Eject any external storage devices such as USB flash drives and backup hard disks. This makes it easier to identify the SD card. Then, insert the SD card into the slot on your computer, or into the reader.
4. Run the Etcher program by double clicking on the Etcher icon shown in the following figure, or select it from the Start menu.

Etcher launches, as shown in the following figure.



5. Select the image file by clicking **Select Image**. You can select a **.zip** or **.gz** compressed file.
6. Etcher tries to detect the SD drive. Verify the drive designation and the image size.
7. Click **Flash!**.



Booting the Evaluation Board

This example uses a ZCU102 board to illustrate how to boot a Vitis AI evaluation board. Follow the steps below to boot the evaluation board.

1. Connect the power supply (12V ~ 5A).
2. Connect the UART debug interface to the host and other peripherals as required.
3. Turn on the power and wait for the system to boot.
4. Login to the system.
5. The system needs to perform some configurations for its first boot. Then reboot the board for these configurations to take effect.

Accessing the Evaluation Board

There are three ways to access the ZCU102 board:

- UART port
- Ethernet connection
- Standalone

UART Port

Apart from monitoring the boot process and checking Linux kernel messages for debugging, you can login to the board through the UART. The configuration parameters of the UART are shown in the following example. A screenshot of a sample boot is shown in the following figure. Login into the system with username “root” and password “root”.

- baud rate: 115200 bps
- data bit: 8
- stop bit: 1
- no parity

Note: On a Linux system, you can use Minicom to connect to the target board directly; for a Windows system, a USB to UART driver is needed before connecting to the board through a serial port.

Using the Ethernet Interface

The ZCU102 board has an Ethernet interface, and SSH service is enabled by default. You can log into the

system using an SSH client after the board has booted.

Use the `ifconfig` command via the UART interface to set the IP address of the board, then use the SSH client to log into the system.

Using the Board as a Standalone Embedded System

The ZCU102 board allows a keyboard, mouse, and monitor to be connected. After a successful boot, a Linux GUI desktop is displayed. You can then access the board as a standalone embedded system.

Installing Vitis AI Runtime on the Evaluation Board

To improve user experience, the Vitis AI Runtime packages, VART samples, Vitis-AI-Library samples and models have been built into the board image. And the examples are precompiled. Therefore, user does not need to install Vitis AI Runtime packages and model package on the board separately. However, users can still install the model or Vitis AI Runtime on their own image or on the official image by following these steps.

With an Ethernet connection established, you can copy the Vitis™ AI runtime (VART) package from github to the evaluation board and set up Vitis AI running environment for the ZCU102 board.

1. Download the `vitis-ai-runtime-1.2.x.tar.gz` from here. Untar it and copy the following files to the board using scp.

   ```
   $tar -xzvf vitis-ai-runtime-1.2.x.tar.gz
   $scp -r vitis-ai-runtime-1.2.x/aarch64/centos root@IP_OF_BOARD:~/
   ```

   Note: You can take the rpm package as a normal archive, and extract the contents on the host side, if you only need some of the libraries. Only model libraries can be separated dependently, while the others are common libraries. The operation command is as follows.

   ```
   $rpm2cpio libvart-1.2.0-r<x>.aarch64.rpm | cpio -idmv
   ```

2. Log in to the board using ssh. You can also use the serial port to login.
3. Install the Vitis AI runtime. Execute the following commands in order.

   ```
   #cd ~/centos
   #rpm -ivh --force libunilog-1.2.0-r<x>.aarch64.rpm
   #rpm -ivh --force libxir-1.2.0-r<x>.aarch64.rpm
   #rpm -ivh --force libtarget-factory-1.2.0-r<x>.aarch64.rpm
   #rpm -ivh --force libvart-1.2.0-r<x>.aarch64.rpm
   ```

   If you want to run the example based on Vitis-AI-Library, execute the following command to install the Vitis-AI-Library runtime package.

   ```
   #rpm -ivh --force libvitis_ai_library-1.2.0-r<x>.aarch64.rpm
   ```

   After the installation is complete, the Vitis AI Runtime library will be installed under `/usr/lib`.