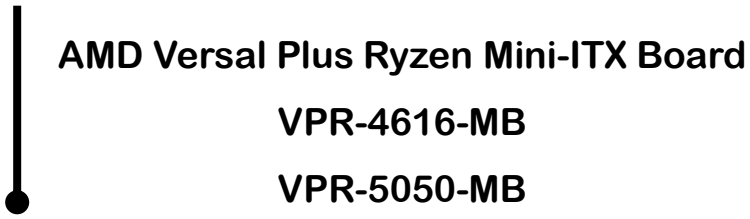
FPGA Software Installation and Firmware Update Instructions – RAVE 1.3, XRT



TRADEMARK

All products and company names are trademarks or registered trademarks of their respective holders.

These specifications are subject to change without notice.

Manual Revision 1.3

November 25, 2026

SAPPHIRE - Embedded+ Initial Platform Install & Config

This page captures the steps to bring-up the Embedded+ platform configuration for the x86 Host and Versal SoC device enablement.

Hardware Setup:

Embedded+ system is comprised of a Versal SoC and a Ryzen SoC. The primary interfaces between these SoCs are PCIe and JTAG. Users interact with the Ryzen SoC through conventional means such as a keyboard and monitor, or via SSH once Linux is installed. Two UART ports are connected to the Versal SoC for debugging purposes only; they are not intended as the primary interface for Embedded+.

This documentation assumes the user possesses an Embedded+ system with OSPI pre-programmed at the factory, as this is the standard shipping configuration. The user will install Linux on the Ryzen X86 first, enabling subsequent interfacing with the Versal SoC.

ⓘ The OSPI binary prebuilts in this release targets VPR-4616, VPR-4616-sys, and VPR-5050. They do not target VPR-5050A as VPR-5050A has a different Versal HW configuration (DDR & MIO configs).

x86 Host OS

This section guide user to install Linux on X86. The Embedded+ platform's X86 host XRT driver for Embedded+ has been validated with Ubuntu 22.04 OS and the XRT Ubuntu library is aligned with the GA 5.15 Linux kernel. Thus, if the default installed Ubuntu image is using a later kernel, the following steps are required to update the x86 Ubuntu OS to be aligned:

ⓘ These steps are necessary for VPR-4616/5050-MB only. VPR-4616/5050-SYS has preloaded software and this section is not necessary.

If you have already done Ubuntu installation for AMR flow, skip to step 2, OSPI flashing.

1. Install the x86 host OS. Instructions and the image are available directly from Canonical:
    a Requires a USB stick, keyboard, monitor, mouse, and ethernet connection
    b Install instructions: https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview
    c OS image download: https://releases.ubuntu.com/jammy/

    Important:
    On systems with an existing Ubuntu installation that previously used an XRT, please uninstall it first

    ```
    sudo rmmod xdma
    ```

2. Update OSPI with XRT OSPI.

   a OSPI binary at <https://account.amd.com/en/forms/downloads/xef.html?filename=emb-plus-apu-image-emb-plus-ve2302-xrt.rootfs-20260609231841.bin > and https://github.com/Xilinx/embpf-bootfw-update-tool follow the instruction of the link to update: GitHub – Xilinx/embpf-bootfw-update-tool. You will need to install git and hw_server.

   b After installation of the utility and its dependencies, the command to update OSPI is:
   sudo ./prog_spi.sh -d embplus -Vpv -i <path to ../emb-plus-ospi-vmr-emb-plus-ve2302-xrt-<version>.bin>

3. Install the XRT drivers on the x86 host.In this release, the Embedded+ platform support is built on 2026.1 and XRT dependencies are built on 2026.1.

   a Get the latest XRT, xrt_202610.2.23.243_22.04-amd64-xrt.deb from the automated builds at: https://account.amd.com/en/forms/downloads/xef.html?filename=xrt_202610.2.23.243_22.04-amd64-xrt.deb

   > ⓘ xrt version
   > Ensure that the XRT version is 2.23.243.

   b Move xrt.deb package to the Embedded+ platform running Ubuntu 22.04

   c Install the xrt package with

   Install XRT driver

   ```
   sudo dpkg -i xrt_202610.2.23.243_22.04-amd64-xrt.deb
   ```

   d The previous step may take some time as it will build the driver locally on target. It may also require resolving dependencies that is not installed. After it completes verify that the drivers are installed correctly using: *lsmod*

   lsmod

   ```
   d773@d773-desktop:~/Desktop/7D773 Linux test tools$ lsmod | grep xocl
   xocl                 2220032  0
   libcrc32c              12288  2 xclmgmt,xocl
   drm                   765952  15 gpu_sched,drm_kms_helper,drm_suballoc_helper,xocl,drm_display_helper,drm_buddy,amdgpu,drm_ttm_helper,ttm,amdxc
   d773@d773-desktop:~/Desktop/7D773 Linux test tools$ lsmod | grep xclmgmt
   xclmgmt              1294336  0
   libcrc32c              12288  2 xclmgmt,xocl
   ```

4. Install the Embedded+ VE2302 "base" device package

   a Get the latest base package from
   https://account.amd.com/en/forms/downloads/xef.html?filename=xrt-emb-plus-ve2302-base_1.3.deb

   b Move package to the Embedded+ platform.

   c Install with:

   Install VE2302 base design files

   ```
   sudo dpkg -i xrt-emb-plus-ve2302-base_1.3.deb
   ```

5. Install the Embedded+ VE2302 XRT platform test bitstream packages
   a Get the latest test bitstream packages from:
      xrt-verify-test-ve2302_1.3.deb:
      https://account.amd.com/en/forms/downloads/xef.html?filename=xrt-verify-test-ve2302_1.3.deb
      xrt-bandwidth-dma-test-ve2302_1.3.deb
      https://account.amd.com/en/forms/downloads/xef.html?filename=xrt-bandwidth-dma-test-ve2302_1.3.deb

      xrt-aie-test-ve2302_1.3.deb
      https://account.amd.com/en/forms/downloads/xef.html?filename=xrt-aie-test-ve2302_1.3.deb
   b Move the packages to the Embedded+ platform.
   c Install with:

Install XRT test bitstreams
sudo dpkg -i xrt-verify-test-ve2302_1.3.deb sudo dpkg -i xrt-bandwidth-dma-test-ve2302_1.3.deb sudo dpkg -i xrt-aie-test-ve2302_1.3.deb

6. Install the Versal APU SW package
   a Get the latest APU SW package from:
      https://account.amd.com/en/forms/downloads/xef.html?filename=xrt-apu-linux-ve2302_1.3.deb
   b Move the package to the Embedded+ platform.
   c Install with:

Install Versal APU SW
sudo dpkg -i xrt-apu-linux-ve2302_1.3.deb

7. Reboot the system by doing a shut down (sudo poweroff) make sure that bootmode J5 is set to the default (not populated, QSPI boot) before powering back on again.

On-target Self Test

The following are self-test that the user can run to test that the Versal and x86 host are set-up and configured correctly.

PCIe Connectivity

Use *lspci* to inspect if Versal device is present on PCIe bus.

lspci -vd 10ee:

```
rrrrr@rrrrr-desktop:~/Desktop$ sudo lspci -vd 10ee:
[sudo] password for rrrrr:
01:00.0 Processing accelerators: Xilinx Corporation Device 5710
        Subsystem: Xilinx Corporation Device 000e
        Flags: bus master, fast devsel, latency 0, IOMMU group 8
        Memory at ffd0000000 (64-bit, prefetchable) [size=256M]
        Memory at fff0010000 (64-bit, prefetchable) [size=64K]
        Memory at ffe8000000 (64-bit, prefetchable) [size=128M]
        Capabilities: [40] Power Management version 3
        Capabilities: [60] MSI-X: Enable- Count=32 Masked-
        Capabilities: [70] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [188] Alternative Routing-ID Interpretation (ARI)
        Capabilities: [1c0] Secondary PCI Express
        Capabilities: [1f0] Virtual Channel
        Capabilities: [450] Access Control Services
        Capabilities: [600] Vendor Specific Information: ID=0020 Rev=0 Len=010 <?>
        Kernel driver in use: xclmgmt
        Kernel modules: xclmgmt

01:00.1 Processing accelerators: Xilinx Corporation Device 5711
        Subsystem: Xilinx Corporation Device 000e
        Flags: fast devsel, IOMMU group 9
        Memory at ffc0000000 (64-bit, prefetchable) [size=256M]
        Memory at fff0000000 (64-bit, prefetchable) [size=64K]
        Memory at ffe0000000 (64-bit, prefetchable) [size=128M]
        Capabilities: [40] Power Management version 3
        Capabilities: [60] MSI-X: Enable- Count=32 Masked-
        Capabilities: [70] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [188] Alternative Routing-ID Interpretation (ARI)
        Capabilities: [450] Access Control Services
        Capabilities: [600] Vendor Specific Information: ID=0020 Rev=0 Len=010 <?>
        Kernel driver in use: xocl
        Kernel modules: xocl
```

XRT Tests

The XRT "validate" tests are a set of PL/AIE design used to exercise basic functionality of the system. They have been installed with xrt*test*.deb packages.
Source the XRT tools:

Source XRT

```
source /opt/xilinx/xrt/setup.sh
```

Now the system is set up to run tests via xbutil command: verify, dma, mem-bw and aie. Instructions in following sections:

XRT Platform Inspection

Use XRT xbmgmt to see platform information.

XRT Platform Inspection
xbmgmt examine
xbmgmt capture

```
rrrrr@rrrrr-desktop:~/Desktop$ xbmgmt examine
System Configuration
  OS Name              : Linux
  Release              : 6.8.0-124-generic
  Machine              : x86_64
  CPU Cores            : 16
  Memory               : 15394 MB
  Distribution         : Ubuntu 22.04.4 LTS
  GLIBC                : 2.35
  Model                :
  BIOS Vendor          : American Megatrends Inc.
  BIOS Version         : 5.16
  Processor            : AMD Ryzen Embedded V2748 with Radeon Graphics

XRT
  Version              : 2.23.243
  Branch               : 2026.1
  Hash                 : 7562163b4cf813e3ee22ac44e640013d1f8657df
  Hash Date            : Wed, 20 May 2026 21:44:20 +0530
  xocl Version         : 2.23.243, 7562163b4cf813e3ee22ac44e640013d1f8657df
  xclmgmt Version      : 2.23.243, 7562163b4cf813e3ee22ac44e640013d1f8657df
  Firmware Version     : N/A

Device(s) Present
|BDF             |Shell     |Logic UUID                            |Device ID       |Device Ready*  |
|----------------|----------|--------------------------------------|----------------|---------------|
|[0000:01:00.0]  |emb-plus  |00000000-0000-0000-0000-0000274E6586  |mgmt(inst=256)  |Yes            |


* Devices that are not ready will have reduced functionality when using XRT tools
```

Verify Test

The "Verify" test is a simple "hello world" application for testing core ability to download a user kernel captured as an xclbin and have an expected data transfer read back from that kernel "Hello World".

Run the test:

Verify Test
xrt-smi validate -r verify -d 0000:01:00.1 --verbose

Expected output:

Verify test results

```
rrrrr@rrrrr-desktop:~/Desktop$ xrt-smi validate -r verify -d 0000:01:00.1 --verbose
Verbose: Enabling Verbosity
Validate Device           : [0000:01:00.1]
    Platform              : emb-plus
    SC Version            : 0.0.0
    Platform ID           : 00000000-0000-0000-0000-0000274E6586
-------------------------------------------------------------------------------
Test 1 [0000:01:00.1]     : verify
    Description           : Run 'Hello World' kernel test
    Test Status           : [PASSED]
-------------------------------------------------------------------------------
Validation completed
```

DMA Test

The "DMA" test is a simple DMA test that transfer data between Versal and Ryzen using DMA on Versal.

Run the test:

DMA test
xrt-smi validate -r dma -d 0000:01:00.1 --verbose

Expected output:

DMA test results

```
rrrrr@rrrrr-desktop:~/Desktop$ xrt-smi validate -r dma -d 0000:01:00.1 --verbose
Verbose: Enabling Verbosity
Validate Device           : [0000:01:00.1]
    Platform              : emb-plus
    SC Version            : 0.0.0
    Platform ID           : 00000000-0000-0000-0000-0000274E6586
-------------------------------------------------------------------------------
Test 1 [0000:01:00.1]     : dma
    Description           : Run dma test
    Details               : Buffer size - '16 MB' Memory Tag - 'MC_NOC'
                            Host -> PCIe -> FPGA write bandwidth = 3198.5 MB/s
                            Host <- PCIe <- FPGA read bandwidth = 3430.0 MB/s
    Test Status           : [PASSED]
-------------------------------------------------------------------------------
Validation completed
```

Bandwidth Test

The "bandwidth" test runs a limited bandwidth test on DDR memory and PCIe data transfers.

Run the test:

Bandwidth test
xrt-smi validate -r mem-bw -d 0000:01:00.1 --verbose

Expected output:

Bandwidth test results

```
rrrrr@rrrrr-desktop:~/Desktop$ xrt-smi validate -r mem-bw -d 0000:01:00.1 --verbose
Verbose: Enabling Verbosity
Validate Device           : [0000:01:00.1]
    Platform              : emb-plus
    SC Version            : 0.0.0
    Platform ID           : 00000000-0000-0000-0000-0000274E6586
-------------------------------------------------------------------------------
Test 1 [0000:01:00.1]     : mem-bw
    Description           : Run 'bandwidth kernel' and check the throughput
    Details               : Throughput (Type: DDR) (Bank count: 1) : 18590.1 MB/s
                            Throughput of Memory Tag: MC_NOC : 18590.1 MB/s
    Test Status           : [PASSED]
-------------------------------------------------------------------------------
Validation completed
```

AIE Test

The "aie" test runs a AIE tile functionality test.

Run the test:

AIE test
xrt-smi validate -r aie -d 0000:01:00.1 --verbose

Expected output:

AIE test results

```
rrrrr@rrrrr-desktop:~/Desktop$ xrt-smi validate -r aie -d 0000:01:00.1 --verbose
Verbose: Enabling Verbosity
Validate Device           : [0000:01:00.1]
    Platform              : emb-plus
    SC Version            : 0.0.0
    Platform ID           : 00000000-0000-0000-0000-0000274E6586
-------------------------------------------------------------------------------
Test 1 [0000:01:00.1]     : aie
    Description           : Run AIE PL test
    Test Status           : [PASSED]
-------------------------------------------------------------------------------
Validation completed
```

Filter 2D Examples

There are two more example applications that can be found in https://github.com/Xilinx/emb-plus-examples.

Follow the instructions in filter2d_aie and filter2d_pl to run the example applications.

Debug Tools – Versal Serial Console

The Versal serial console is connected to the Ryzen device on the motherboard. Therefore user can access the uart outputs from Ryzen. In Ubuntu, first download picocom:

Install picocom
sudo apt-get install picocom

Then user can access the com ports on commandline from Ubuntu:

Connect to APU serial output:
sudo picocom -b 115200 /dev/ttyUSB1

Connect to PLM/RPU serial output:
sudo picocom -b 115200 /dev/ttyUSB2

Repositories

The following links are the sources of this example artifacts
https://github.com/Xilinx/emb_plus_vitis_platforms/releases
https://github.com/Xilinx/emb-plus-examples/releases

The following repository contains the Yocto meta layer for the Embedded+ platforms:
https://github.com/ilinx/meta-embedded-plus

Released Artifacts, Source Code, and License Information:
https://edf.docs.amd.com/en/latest/downloads-and-release-notes.html#embedded-images

FPGA (Xilinx) F/W Update

The VPR-4616/5050-MB and VPR-4616-SYS are both shipped with OSPI image. You can check their version with “xbmgmt examine” command. Each release is based on a particular AMD tool version, and the released OSPI bin file contains a specific UUID the rest of the applications look for.

Release	AMD tool version	Bin file name	UUID
0.5	2023.2	emb-plus-ospi-emb-plus-ve2302-20240620051522.bin	00000000-0000-0000-0000-000079DB078F
1.0	2024.2	emb-plus-ospi-emb-plus-ve2302-sdt-20250227030657.bin	00000000-0000-0000-00000000E97E0A06
1.2	2025.1.1	emb-plus-ospi-vmr-emb-plus-ve2302-xrt-20250912165216.bin	00000000-0000-0000-0000-00003A8A4EB8
1.3	2026.1	emb-plus-ospi-vmr-emb-plus-ve2302-xrt-20260609231841.bin	00000000-0000-0000-0000-0000274E6586

Upgrading OSPI is necessary if the UUID does not match the release.

Please follow the instruction of the link for update
GitHub – Xilinx/embpf-bootfw-update-tool