FPGA Software Installation and Firmware Update Instructions – RAVE 2.0, AMR

AMD Versal Plus Ryzen Mini-ITX Board

VPR-4616-MB

VPR-5050-MB

These specifications are subject to change without notice.

Manual Revision 2.0
September 19, 2025

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.

x86 Host OS

This section guide user to install Linux on X86, update OSPI and install AMI driver, tool and library

> ⓘ Steps 1-3 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 XRT flow, skip to step 5, 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/

2. Once the x86 host OS is installed and booted from its SSD. Update the kernel to the 5.15 generic kernel with these steps

    a. Install the generic kernel

    Generic kernel install

    ```
    sudo apt install linux-image-generic
    sudo apt install -f
    ```

b. Replace the "GRUB_DEFAULT" string in: */etc/default/grub* with *GRUB_DEFAULT="Advanced options for Ubuntu>Ubuntu, with Linux 5.15.0-###-generic".*

Note that the ### in the string above needs to be aligned with the generic kernel number used in the install of the previous step

c. Update grub config & reboot:

Kernel update

```
sudo update-grub
sudo reboot now
```

3. Validate the kernel update using:

Validate kernel

```
uname -r
```

4. Install linux-header:

Install Linux header

```
sudo apt-get install linux-headers-$(uname -r)
```

5. Update OSPI with AMR OSPI.

a OSPI binary at https://account.amd.com/en/forms/downloads/member-xef.html?filename=emb-plus-ospi-amr-emb-plus-ve2302-amr-20250912165411.bin and 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:

1. sudo ./prog_spi.sh -d embplus -Vpv   -i ../ <path to ______insert right name emb-plus-ospi-amr-emb-plus-ve2302-amr.bin>

6. Install the AMI drivers on the x86 host.

a Get the latest AMI *.deb files* from:

https://account.amd.com/en/forms/downloads/member-xef.html?filename=amitool_1.0.0-0.60d33f87.20250912_amd64_22.04.deb

https://account.amd.com/en/forms/downloads/member-xef.html?filename=ami_1.0.0-0.60d33f87.20250912_amd64_22.04.deb

https://account.amd.com/en/forms/downloads/member-xef.html?filename=libami_1.0.0-0.60d33f87.20250912_amd64_22.04.deb b Move *.deb packages to the Embedded+ platform running Ubuntu 22.04

c Install the AMI driver, tool and library package with the following, and install any dependencies that’s required:

Install AMI driver

```
sudo dpkg -i ami_x.x.x.xxx.xxx_amd64_22.04.deb

sudo dpkg -i amitool_x.x.x.xxx.xxx_amd64_22.04.deb

sudo dpkg -i libami.x.x.xxx.xxx_amd64_22.04.deb
```

d The previous step may take some time as it will build the driver locally on target. After it completes verify that the drivers are installed correctly using: *lsmod*

lsmod

```
amd@amd-desktop:~/rave_release$ sudo lsmod |grep -i ami

ami                          253952   0
```

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.

Refer to https://github.com/Xilinx/amr for more information on how to build AMR firmware, software and how to use AMI tools.

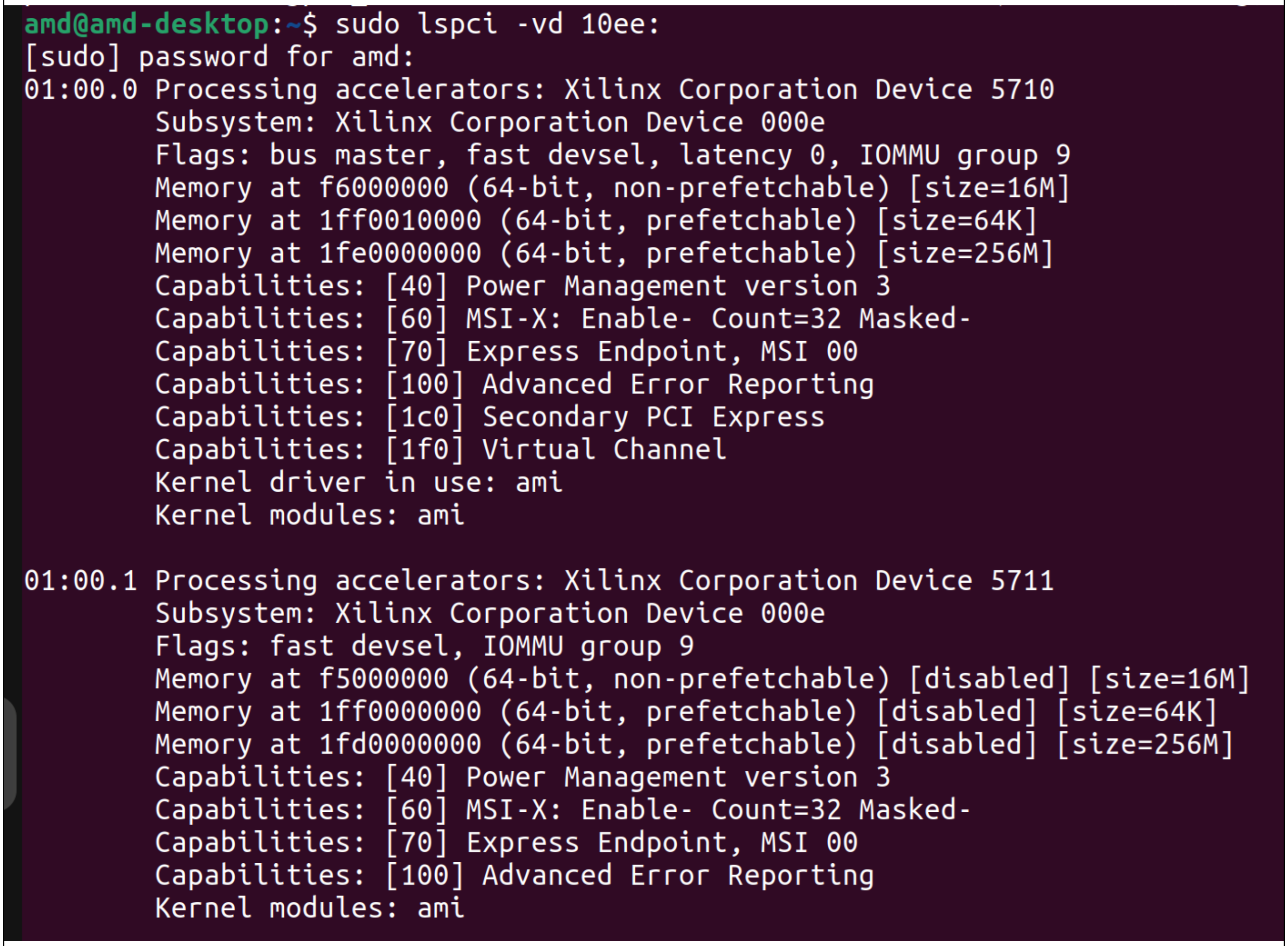
On-target Test

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



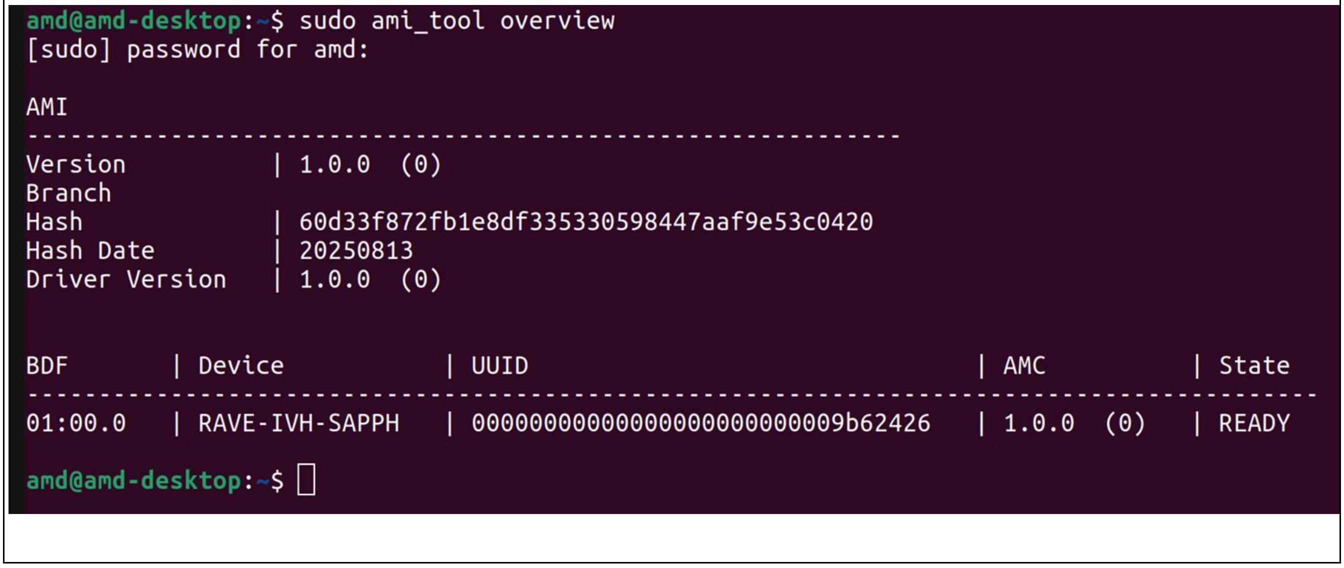
```
amd@amd-desktop:~$ sudo lspci -vd 10ee:
[sudo] password for amd:
01:00.0 Processing accelerators: Xilinx Corporation Device 5710
        Subsystem: Xilinx Corporation Device 000e
        Flags: bus master, fast devsel, latency 0, IOMMU group 9
        Memory at f6000000 (64-bit, non-prefetchable) [size=16M]
        Memory at 1ff0010000 (64-bit, prefetchable) [size=64K]
        Memory at 1fe0000000 (64-bit, prefetchable) [size=256M]
        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: [1c0] Secondary PCI Express
        Capabilities: [1f0] Virtual Channel
        Kernel driver in use: ami
        Kernel modules: ami

01:00.1 Processing accelerators: Xilinx Corporation Device 5711
        Subsystem: Xilinx Corporation Device 000e
        Flags: fast devsel, IOMMU group 9
        Memory at f5000000 (64-bit, non-prefetchable) [disabled] [size=16M]
        Memory at 1ff0000000 (64-bit, prefetchable) [disabled] [size=64K]
        Memory at 1fd0000000 (64-bit, prefetchable) [disabled] [size=256M]
        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
        Kernel modules: ami
```

AMI Platform Inspection

Use ami overview to see platform information.

ami_tool



```
amd@amd-desktop:~$ sudo ami_tool overview
[sudo] password for amd:

AMI
-------------------------------------------------------------
Version          | 1.0.0  (0)
Branch
Hash             | 60d33f872fb1e8df335330598447aaf9e53c0420
Hash Date        | 20250813
Driver Version   | 1.0.0  (0)


BDF       | Device            | UUID                                | AMC            | State
------------------------------------------------------------------------------------------
01:00.0   | RAVE-IVH-SAPPH    | 00000000000000000000000009b62426    | 1.0.0  (0)     | READY

amd@amd-desktop:~$ 
```

Note that the UUID compatible with AMR flow release 2.0 is 00000000000000000000000009b62426

AXI PCIE GPIO Test

The GPIO Driver for PCIe Interface test can be found at https://github.com/Xilinx/axi-gpio-pcie-module

Refer to https://github.com/Xilinx/axi-gpio-pcie-module for instructions to run GPIO tests.

BRAM, GPIO and DMA Test

Get the latest test bitstream packages from: https://account.amd.com/en/forms/downloads/member-xef.html?filename=amr-bram_gpio-dma-test-ve2302_2.0.deb

Refer to https://github.com/Xilinx/axi-gpio-pcie-module for instructions to run GPIO tests.

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/amr

https://github.com/Xilinx/axi-gpio-pcie-module