



Clara Holoscan Developer Kit User Guide

Version 1.4

Document History

Document Version 1.4

Version	Date	Authors	Description of Change
1.1	July 7, 2021	Peter G.	Initial release
1.2	September 23, 2021	Peter G.	Add further Recovery mode instructions.
1.3	October 19, 2021	Peter G.	Add checklist and additional instructions.
1.4	November 8, 2021	Peter G.	Update to Holoscan, additional instructions.

Table of Contents

- Introduction..... 1**
- Hardware Setup 2**
 - Checklist for Setting up the Developer Kit..... 2
 - Requirements..... 2
 - Precautions 3
- System Overview 4**
 - Main Components 4
 - I/O and external interfaces 4
- Getting Started 7**
 - Powering up the System 7
 - Flashing and Updating Clara Holoscan Developer Kit using SDK Manager 7
 - Switching between iGPU and dGPU 8
 - Reinstalling dGPU Mode Clara Holoscan SDK Packages 9
 - Setting up SSD Storage..... 10
 - Completing Docker Setup 10
 - Additional Resources 10

Introduction

This user guide provides instructions to flash, setup, and start using a Clara Holoscan Developer Kit.

Disclaimer: The Clara Holoscan Developer Kit is not an approved medical device and is not intended for clinical use.

Hardware Setup

Checklist for Setting up the Developer Kit

After receiving the Clara Holoscan Developer Kit, ensure the following actions are taken before developing on the kit. Each action is described in its corresponding section of this user guide.

- Read through the **Hardware Setup** requirements and precautions.
- Familiarize yourself with the **System Overview**: the main components and system I/O.
- Power up the system.
- Flash and update the Clara Holoscan Developer Kit using SDK Manager.
- Switch from iGPU to dGPU mode and reinstall the dGPU mode packages.
- Set up the 250GB SSD storage and Docker.

Requirements

- A Clara Holoscan Developer Kit
- A compatible power cable
 - Your NVIDIA Clara Holoscan Developer Kit may not include a power cable compatible with your local electrical requirements.
 - A compatible cable should meet the following requirements:
 - Provides a certified local 3-prong AC power plug.
 - Provides a C13 connector.
 - Supports ratings of 100-120VAC/6A, 200-240VAC/3A, or higher with minimum wire thickness of 18AWG and insulation rating of 300V or higher.
- An Ubuntu 18.04 host system (for use during flashing)
- A standard USB-A to USB-C or USB-C to USB-C cable with data enabled (for use during flashing)
- Connection to the Internet for the host system before and during flashing, and for the Clara Holoscan Developer Kit after flashing
- A keyboard, mouse, and monitor with HDMI for the Clara Holoscan Developer Kit

Precautions

- Only connect and disconnect a PCIe card (e.g. miniSAS or dGPU) when the system is powered down.
- Apply extra care when plugging and removing PCIe cards to avoid stress on the PCIe connectors (wearing, bending, breaking).
- The rightmost USB connector is USB 2.0 (even if the color is blue). The other two USB connectors are USB 3.0.



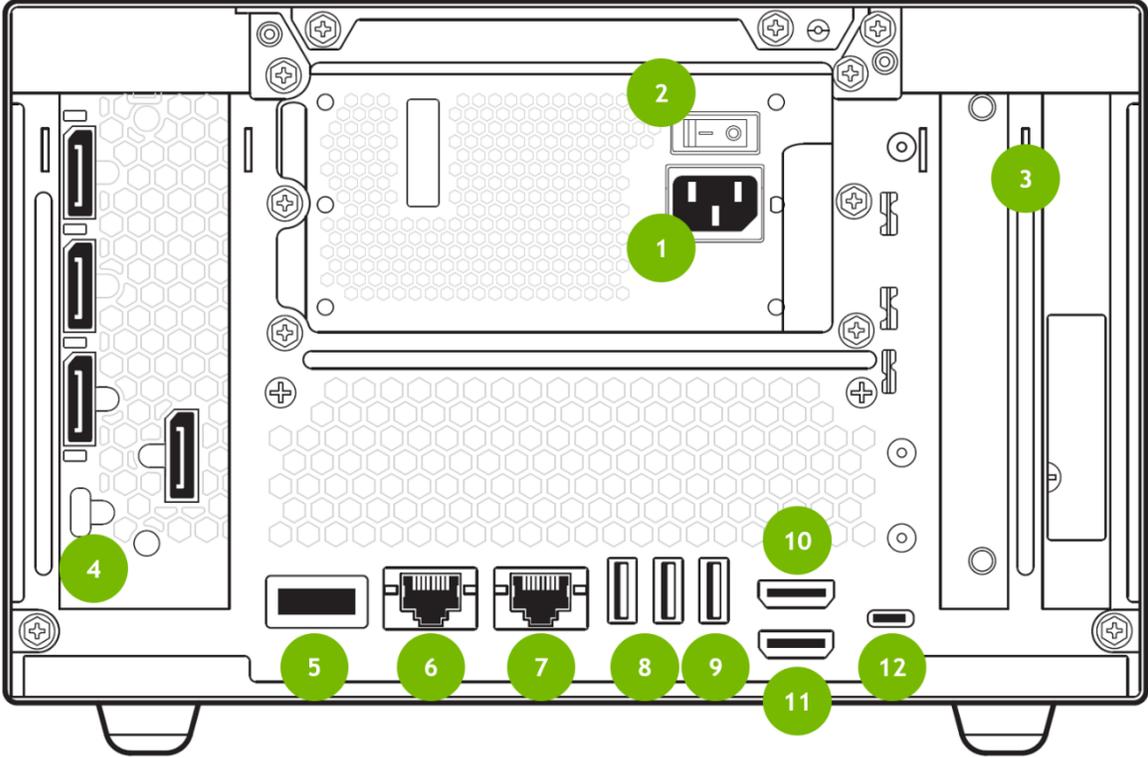
System Overview

Main Components

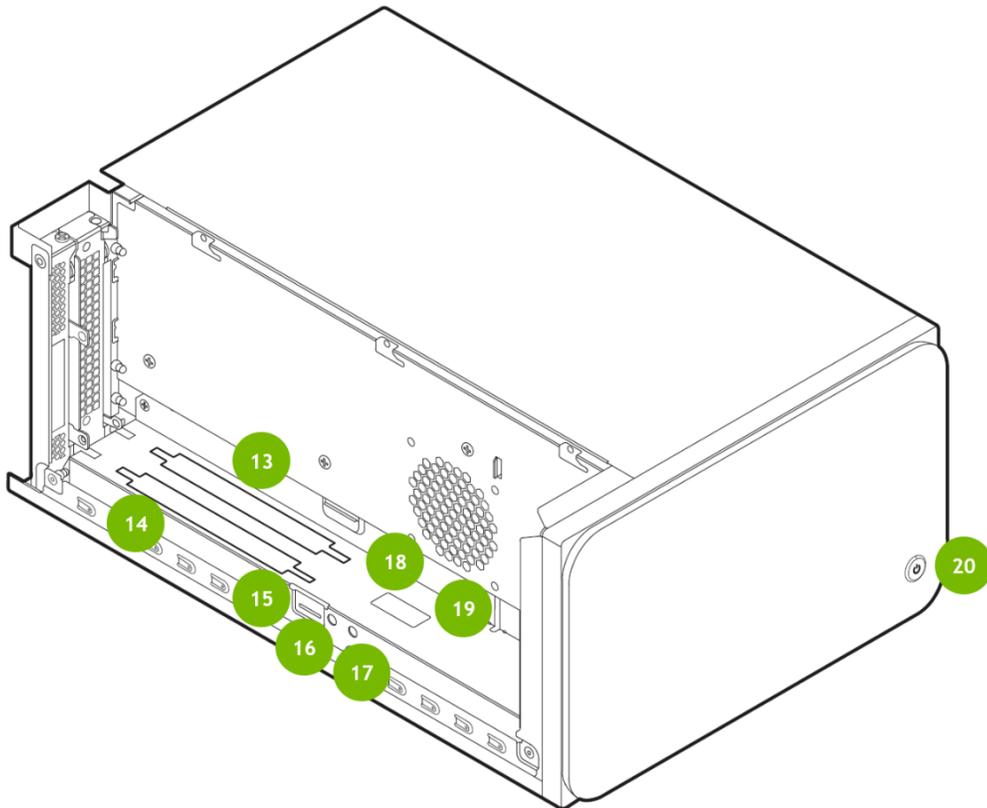
The Clara Holoscan Developer Kit contains the following major components:

- [AGX Xavier 32 GB Module](#)
- [Quadro RTX 6000 discrete GPU](#)
- [ConnectX-6 IC](#)
- 250GB Removeable SSD

I/O and external interfaces



- 1) Power cable connection
- 2) Power switch
- 3) PCIe slots for customer cards
- 4) dGPU
 - a) Reference the [GPU section](#) below to determine which display output to use.
- 5) 100 GbE QSFP28 Ethernet connector to NVIDIA ConnectX-6
- 6) 10 GbE RJ45 Ethernet connector to NVIDIA ConnectX-6
 - a) The 10GbE connector only supports 10 GbE speeds.
- 7) 1 GbE RJ45 Ethernet connector to Xavier module
- 8) USB 3.0 ports (2x)
- 9) USB 2.0 port (1x)
- 10) HDMI out
 - a) Reference the [GPU section](#) below to determine which display output to use.
- 11) HDMI in: Connect to instruments that output HDMI to the platform
- 12) Debug USB-C port: Connect to the Linux host system for flashing and serial port connections.



- 13) x8 PCIe slot (x16 physical) for customer card, to Xavier Module

- 14) x8 PCIe slot (x16 physical) for customer card, to RTX6000
 - a) Used for cards supporting GPU Direct RDMA data transfer to RTX6000 GPU
- 15) SD card slot
- 16) Recovery button
- 17) Reset button
- 18) Main secondary compartment fan connector (there is an additional system fan inside the chassis)
- 19) Auxiliary fan connector for card fans



20) Power button

Getting Started

Powering up the System

1. Connect all peripherals to the system before powering up the system.
2. Connect the power cable to the system in the slot labeled (1) in the picture above.
3. Once the power is connected, press the power button (20) about 1 second. It should light up. The machine can be powered off by depressing the power button for approximately 10 seconds.
4. If you have a display connected, you might already see the system booting on it. During flashing or re-flashing, use the HDMI out port (10) to connect to the display. Reference the [GPU section](#) below to determine how to choose between display outputs.

Flashing and Updating Clara Holoscan Developer Kit using SDK Manager

1. Register and activate an NVIDIA Developer Account [here](#).
2. If you are running a VPN on your host system, log off before flashing the Clara Holoscan Developer Kit.
3. From the host system, download and install the latest version of [NVIDIA SDK Manager](#). Instructions for downloading and setting up NVIDIA SDK Manager can be found [here](#).
4. Connect the Clara Holoscan Developer Kit to the host system via USB-C (12).
5. From the NVIDIA SDK Manager, download and flash the Clara Holoscan Developer Kit. See the [step-by-step instructions](#) for more details.

Note: The Clara Holoscan SDK includes DeepStream 6.0 EA, but DeepStream 5.1 may also appear as a stand-alone **Additional SDK** in SDK Manager. Only one version of DeepStream may be installed at the same time, so be sure to unselect DeepStream 5.1 during the Clara Holoscan SDK installation.

- a) If you joined the Clara Holoscan SDK program after your initial SDK Manager login, you will need to log out and log in again on SDK Manager for the permissions to take effect.

- b) If the SDK Manager does not see the Clara Holoscan Developer Kit, it may need to be placed in Recovery mode. Follow these steps to place the Clara Holoscan Developer Kit in recovery mode:
 - i) Power off the unit.
 - ii) Remove the left-hand side cover to expose ports 13-19 by unscrewing the two Phillips screws that secure it at the back of the machine.
 - iii) Hold down the Recovery button (16) while pressing the Power button (20) to turn on the unit.
 - iv) Release the Power Button, then the Recovery button.
 - c) If Recovery mode doesn't help, try resetting the unit:
 - i) While the unit is powered on, press the Recovery button (16) + Reset button (17), then release both buttons.
6. Use the default credentials (shown below) when SDK Manager is preparing to flash the Clara Holoscan Developer Kit. If Automatic Setup fails, try switching to Manual Setup.
- a) Username: ubuntu
 - b) Password: ubuntu

Switching between iGPU and dGPU

The Clara Holoscan Developer Kit can use either the Xavier AGX module GPU (iGPU, – integrated GPU) or the RTX6000 add-in card GPU (dGPU, – discrete GPU). You can only use one type of GPU at a time.

By default, the Clara Holoscan Developer Kit uses the iGPU. Switching between the iGPU and dGPU is performed using the `nvgpuswitch.py` script contained in the `/usr/local/bin` directory.

To switch from the iGPU to the dGPU, follow these steps:

1. Connect the Clara Holoscan Developer Kit to the Internet using one of the following methods:
 - a) An Ethernet cable connected to a router or Wi-Fi extender
 - i) The 10 GbE RJ45 Ethernet connector only supports 10 GbE speeds.
 - b) A USB Wi-Fi receiver
 - i) Not all USB Wi-Fi receivers will work out of the box on the Clara Holoscan Developer Kit.
 - ii) The TP-Link Archer T2U Nano USB Wi-Fi Adapter has been tested with the Clara Holoscan Developer Kit.
2. To view the currently installed drivers and their version, use the query command:

```
$ nvgpuswitch.py query
iGPU (nvidia-l4t-cuda, 32.5.0-20201012161040)
```

3. To install the dGPU drivers, use the install command with the dGPU parameter (note that sudo must be used to install drivers):

```
$ sudo nvgpuswitch.py install dGPU
```

The install command will print out the list of commands that will be executed as part of the driver install, then continue to execute those commands. This aids with debugging if any of the commands fail to execute for any reason. The following arguments may also be provided with the install command:

```
-d Does a dry run, showing the commands that would be
    executed by the install but does not execute them.

-v Enable verbose output (used with '-d' to describe
    each of the commands that would be run).

-i Run commands interactively (asks before running
    each command).

-l [LOG]
    Writes a log of the install to the file 'LOG'.
```

4. The dGPU driver install may be verified using the query command:

```
$ nvgpuswitch.py query
dGPU (cuda-drivers, 455.32.00-1)
```

Note: CUDA installs its runtime binaries such as nvcc into its own versioned path that is not included by the default `$PATH` environment variable. Because of this, attempts to run commands like `nvcc` will fail on dGPU unless the CUDA 11.1 path is added to the `$PATH` variable. To add the CUDA 11.1 path for the current user, add the following line to `$HOME/.bashrc` after the switch to dGPU:

```
$ export PATH=/usr/local/cuda-11.1/bin:$PATH
```

If at any time you want to switch back to iGPU, use the install command with the iGPU parameter:

```
$ sudo nvgpuswitch.py install iGPU
```

After the iGPU drivers have been installed, rebooting the system will complete the switch back to iGPU. At this point the Ubuntu desktop will be output via the onboard HDMI, so the display cable must be switched from the DisplayPort on the dGPU (4) to the onboard HDMI (10). If the output connection isn't switched, the terminal screen will hang during booting.

Note: The GPU settings will persist through reboots until it is changed again with `nvgpuswitch.py`.

Reinstalling dGPU Mode Clara Holoscan SDK Packages

After rebooting the system, refer to “Switching Between iGPU and dGPU; Reinstalling Clara Holoscan SDK Packages” in the Clara Holoscan SDK [Documentation](#) for instructions on how to reinstall the libraries and samples needed to run sample applications.

Setting up SSD Storage

Make sure you have followed the **Storage Setup (m2 SSD)** instructions in the Clara Holoscan SDK [Documentation](#) to mount the 250 GB SSD onto your system and move the docker storage to the SSD. Without this step, you will likely quickly fill up the root directory with Docker image pull operations, since a complete installation of the Clara Holoscan SDK leaves only about 10GB of storage remaining in the root 32GB.

Completing Docker Setup

Complete the Docker [Post-installation steps for Linux](#) before starting to develop your applications using Docker.

Additional Resources

For other documentation, see release notes from [Clara Holoscan SDK page](#).

For further Jetson documentation, see the [L4T documentation](#).

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

Arm

Arm, AMBA and Arm Powered are registered trademarks of Arm Limited. Cortex, MPCore and Mali are trademarks of Arm Limited. All other brands or product names are the property of their respective holders. "Arm" is used to represent Arm Holdings plc; its operating company Arm Limited; and the regional subsidiaries Arm Inc.; Arm KK; Arm Korea Limited.; Arm Taiwan Limited; Arm France SAS; Arm Consulting (Shanghai) Co. Ltd.; Arm Germany GmbH; Arm Embedded Technologies Pvt. Ltd.; Arm Norway, AS and Arm Sweden AB.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Trademarks

NVIDIA, the NVIDIA logo, Jetson AGX Xavier, and Quadro are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2021 NVIDIA Corporation. All rights reserved.