Software manual for setting up the HSI-v4 and ZED X mini stereo camera with the Jetson Orin Nano.

**Flashing the Orin Nano to SSD (not SD!)**

The setup is partly replicated from JetsonHacks:

[Jetson Orin Nano Tutorial: SSD Install, Boot, and JetPack Setup - Full Guide!](https://www.youtube.com/watch?v=q4fGac-nrTI)

[Jetson Orin Nano Tutorial: SSD Install, Boot, and JetPack Setup - JetsonHacks](https://jetsonhacks.com/2023/05/30/jetson-orin-nano-tutorial-ssd-install-boot-and-jetpack-setup/)

With the caveat that I used NVIDIAS SDK manager, instead of his command-line solution, as described by:

[Nvidia Jetson Orin Nano unboxing & SSD install](https://www.youtube.com/watch?v=FX2exKW_20E&t=367s)

The flashing assumes that you have:

* A Jetson orin nano development kit and all that is in the box
* A data capable USB-c to USB cable (for flashing)
* A female-female jumper cable for putting jetson in “force recovery mode”
* An M.2 PCIe gen 3 SSD, I used *Samsung 970 EVO Plus 1TB PCIe SSD NVMe M.2.* The Orin Nano supports PCIe gen 3 (not gen 4) with form factor M.2.
* A linux host machine (x86) for running the NVIDIA SDK manager. As we need Jetpack 5.1.2, you need a machine with 18.04 or 20.04
* A screwdriver for the SSD retaining screw

I’d advice following the bottom Youtube tutorial as it is concise. Make sure to double check pin when doing the “force recovery mode” with the jumper cables. The relevant pins are not on the 40 GPIO header, but on the horizontal pins under the fan. The two pins are nr 9 and 10. Be careful to do things in the right order, as specified by the tutorial. It is worth noting that the SDK manager downloads and installs 1) Jetson Linux and 2) Jetson Runtime components and 3) Jetson SDK components. Only 1) is essential, and my experience is that 2) and 3) may fail. However, these can be installed on the jetson via internet as long as the flashing of Linux 1) is successful.

In the case that steps 2) and/or 3) failed, while 1) was successful, connect the Jetson to HDMI, keyboard, mouse and log in. Open a terminal and run

>sudo apt update

>sudo apt install nvidia-jetpack

This will take 1 hour or so. See for more

[How to Install JetPack](https://docs.nvidia.com/jetson/jetpack/install-jetpack/index.html)

When it comes to OS distribution, make an informed decision based on your payload hardware. For our case, ZED SDK 4.1 supports only Jetpack 5.1.2 and 5.1.1 for the Jetson Orin*.* The HSI v4 currently has an old IDS camera, and the peak SDK (most user friendly) and we choose the

Device info:

username/password: ubuntu/ubuntu

IP for SSH: 192.168.1.100

**Getting the ZED X Mini running on Orin**

Required hardware/software:

* ZED X Mini: [ZED X Mini Stereo Camera | Stereolabs](https://store.stereolabs.com/en-eu/products/zed-x-mini-stereo-camera)
  + 1x ZED X Mini
  + 1x 1.5m GMSL2 Female to Female cable (FAKRA cable)
* ZED Link Mono Capture Card [ZED Link Mono Capture Card | Stereolabs](https://store.stereolabs.com/en-eu/products/gmsl2-capture-card-mono) :
  + 1x GMSL2 Capture Card
  + 1x JST to jack connector
  + 1x CSI 22pin 0.5mm FFC cable (OP)
* Jetson Orin Nano flashed with Jetpack 5.1.1 or **5.1.2**. You must install the full runtime package incl CUDA 11.4.
* Power supply Capture Card 12-19 V that fits the power jack **(Must be acquired!!!)**. Double check that the positive pose is in the middle! The current requirement is low, so all 12V-19V power suppliers normally satisfy it.

The reader is referred to [Setting up ZED X on Orin Nano / NX Developer Kits - Stereolabs](https://www.stereolabs.com/docs/get-started-with-zed-x/jetson-orin-devkit-setup)

Describing how to get started. See also [ZED X Datasheet Mar2023](https://alcom.be/uploads/zed-x-datasheet-march-2023.pdf)

To connect the hardware, do the following:

* Please make sure the DevKit and the capture card are powered **OFF** and not connected to a power source before the installation.
* Connect MIPI/CSI cable as shown in one of the images here: [ZED Link Mono Capture Card | Stereolabs](https://store.stereolabs.com/en-eu/products/gmsl2-capture-card-mono) (device is card and host is Orin cam0 port). Be careful with the MIPI clamps and google how to do it, do not use force.
* Connect the female-female [GMSL2 Fakra Cable for Cameras | Stereolabs](https://store.stereolabs.com/products/gmsl2-fakra-cables) to the camera and capture card. You should hear a click when they are securely connected.
* Connect the power to the capture card
* Connect the jetson and login with screen/keyboard/mouse
* Download ZED SDK 4.1 ZED SDK for JetPack 5.1.2 (L4T 35.4) from [ZED SDK 4.1 - Download | Stereolabs](https://www.stereolabs.com/developers/release) (ZED\_SDK\_Tegra\_L4T35.4\_v4.1.0.zstd.run)

cd ~/Downloads/

chmod +x ZED\_SDK\_Tegra\_L4T35.4\_v4.1.0.zstd.run

./ZED\_SDK\_Tegra\_L4T35.4\_v4.1.0.zstd.run

Go through installation Yes/No questions

* Download driver under ZED X, ZED link mono, 5.1.2 (stereolabs-zedx\_1.0.0-ZED-LINK-MONO-L4T35.4.1\_arm64.deb)

sudo dpkg -i stereolabs-zedx\_1.0.0-ZED-LINK-MONO-L4T35.4.1\_arm64.deb

sudo apt install libqt5core5a

reboot

* Test the driver:

sudo: dmesg | grep zedx

PS Everytime there is a hardware change, a reboot or following is needed.

sudo systemctl restart zed\_x\_daemon

* Test the depth viewer (GUI app) through:

cd /usr/local/zed/tools/

ZED\_Depth\_Viewer

Software development (here in python API):

* I Recommend using VSCode with the Python extension.
* Sample scripts are found under /usr/local/zed/samples for doing all that the ZED SDK can do.
* (Optional): Installing conda/mamba is convenient and can be done from:

<https://github.com/conda-forge/miniforge?tab=readme-ov-file>

If so you need to follow the description in:  
<https://github.com/stereolabs/zed-python-api/tree/master>

For how to install.

* Scripts generally run as they are supposed to, except that you need to modify the recording parameter for compression:  
  recording\_param = sl.RecordingParameters(opt.output\_svo\_file, sl.SVO\_COMPRESSION\_MODE.LOSSLESS)

And potentially the

* Additional documentation:

<https://www.stereolabs.com/docs/api/python/index.html>

* For AutoComplete follow /[community.stereolabs.com/t/no-auto-suggest-in-vscode-for-pyzed-sl/1633](http://community.stereolabs.com/t/no-auto-suggest-in-vscode-for-pyzed-sl/1633) where you download the \*.pyx from:  
  <https://github.com/stereolabs/zed-python-api/tree/master/src>

A note on the SVO format. It is ZED’s video recording format and it makes sense to record in this one. For our non-realtime purposes we can use playback to do many of the interesting things. The following is needed for proof-of-concept:

* Record data: The script record\_svo.py records an SVO file with image data and imu data. We process this in the playback mode for our purposes (although realtime options exist).
* Timestamped images: The images themselves are not timestamped, but the csv generated from the poses give a pose per time stamp. The script export\_
* Timestamped poses: The script /position\_post.py —input\_svo\_file “path/to/your/file.svo” reads the position tracking data from the camera and calculates a trajectory.
* Mesh generation:

**THE NEXT STEP IS TO RUN THE ZED SDK python scripts for recording SVO, positional tracking, and mesh-generation (spatial\_mapping) timestamps are essential.**

**HSI-v4 on Jetson Orin**

Required hardware/software:

* HSI-v4 (see [Do it yourself hyperspectral imager for handheld to airborne operations](https://opg.optica.org/oe/fulltext.cfm?uri=oe-26-5-6021&id=382214) )
  + 1x HSI v4 with IDS camera
  + 1x USB3 cable
* Jetson Orin Nano flashed with Jetpack **5.1.2**.

The goal is to install IDS’s SDK components including ueye.

Step 1. Go to [The comprehensive download center for all IDS products](https://en.ids-imaging.com/downloads.html) and search for the camera model we are using “UI-3060CP-M-GL Rev.2”. You will need to create an account where username is your email address. You will see that you need IDS peak 2.8 (or higher with transport layer/support for **UI models**). The simplest installation is via the Debian repository server. Installation for Windows is very simple, while linux desktop is similar to Debian.

Step 2: Run the commands below, and see <https://en.ids-imaging.com/files/downloads/ids-peak/readme/ids-peak-linux-readme-2.8.0_EN.html> and section

“Installation of IDS Peak->Debian packages (Debian repository server)” for counciling.

* The first step is to download the gpg key. Just open the following link:

<https://repo.ids-imaging.com/ids.pgp>

* Rename the file to gpg.key with: mv ids.pgp gpg.key
* gpg --dearmor ./gpg.key
* sudo mv ./gpg.key.gpg /etc/apt/trusted.gpg.d/ids.gpg
* sudo chown root:root /etc/apt/trusted.gpg.d/ids.gpg
* sudo chmod ugo+r /etc/apt/trusted.gpg.d/ids.gpg
* sudo chmod go-w /etc/apt/trusted.gpg.d/ids.gpg
* sudo touch /etc/apt/auth.conf.d/ids-peak-repo.conf
* sudo sh -c 'echo "machine repo.ids-imaging.com/ubuntu login <IDS webstore user name> password <IDS webstore password>" > /etc/apt/auth.conf.d/ids-peak-repo.conf'
* sudo touch /etc/apt/sources.list.d/ids-peak-repo.list
* sudo sh -c 'echo "deb [signed-by=/etc/apt/trusted.gpg.d/ids.gpg] https://repo.ids-imaging.com/ubuntu <distribution, see list above> main" > /etc/apt/sources.list.d/ids-peak-repo.list'
* sudo apt-get update && sudo apt-get install ids-peak-with-ueyetl

To check that the camera is working, connect a USB and open the GUI application IDS peak Cockpit. A frame display for the camera should show.

For programmetric control, check out havardlovas’s fork of openhsi

<https://github.com/havardlovas/openhsi/blob/master/nbs/api/cameras/ids_peak.ipynb> (Parser/interface between IDS cameras and openhsi. Can also do recordings here)

<https://github.com/havardlovas/uw-openhsi/blob/main/uw-openhsi/main.py> (Function for running hsi)

The ids\_peak.ipynb recording builds on ids\_peak and pip installable wheels for python 3.7-3.11 will be located under “usr/local/share/ids/bindings/python”. Install all three modules for your python version, e.g. pip install \**cp310\*.* The reference for the SDK is given at:  
<https://www.1stvision.com/cameras/IDS/IDS-manuals/en/>

The openhsi repo is built around "Notebook development" (<https://nbdev.fast.ai/>). This means that if you make a change to the cell in the camera parser "ids\_peak.ipynb" that starts with

#| export cameras

and then run "nbdev\_export" in the terminal (from openhsi top dir) then the cell (typically a class) will be exported to openhsi/cameras.py which is an autogenerated python module that can be imported externally via

from openhsi.cameras import IDSCamera

A bit out of the ordinary as a dev-strategy, but surprisingly convenient. I find that the simplest option for dev is to pip install --editable path/to/havardlovas/fork/of/openhsi). Notably you need openhsi to run the main.py in uw-openhsi. Currently, the easiest way of running it, is through using the main.py function from “uw-openhsi”.