SUNIX AI Accelerator Card User Manual

Disclaimer

SUNIX shall not be liable for any incidental or consequential damages resulting from the performance or use of this equipment.

SUNIX makes no representations or warranties regarding the contents of this manual. Information in this manual has been carefully checked for reliability; however, no guarantee is given as to the correctness of this content. In the interest of continued product improvement, this company reserves the right to revise the manual or include change in the specifications of the product described within it at any time without notice and without obligation to notify any person of such revision or changes. The information contained in this manual is provided for general use by the customers.

Safety Information

1. Keep this User's Manual for future reference.
2. Always read the safety information carefully.
3. Keep this equipment away from direct sunlight, or in humid or damp places.
4. Do not place this equipment in an unstable position, or on vibrating surface before setting it up.
5. Do not use or place this equipment near magnetic fields, televisions, or radios to avoid electronic interface that affects device performance.

Table of Contents

1. System Requirements

Hardware:

Host system CPU Architecture

- x86 or ARM based
- 16+ GB RAM (32+ GB recommended)

Software:

Pre-requisite

- Access to Hailo Developer Zone
- Ubuntu 20.04/22.04, 64 bit
  Hailo software package support：
  - Dataflow Compiler、
  - HailoRT
  - PCIe Driver
  - pyHailoRT
  - Integration Tool
  - Model Zoo
  - TAPPAS
- Microsoft Windows 10/11, 64 bit
  Hailo software package support：
  - HailoRT
  - PCIe Driver
  - pyHailoRT

AI frameworks support：

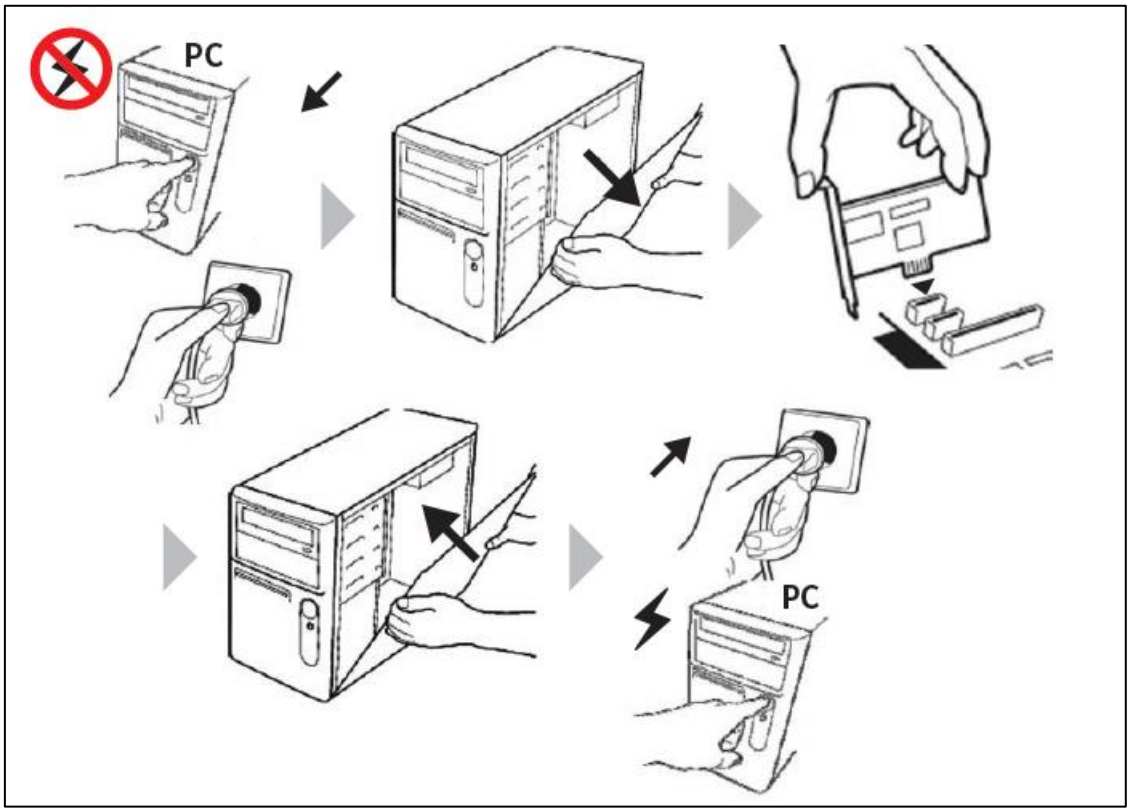
- TensorFlow
- TensorFlow Lite
- Keras
- PyTorch

2. Hardware Installation

The hardware installation of PCI Express AI accelerator board is easy to carry out. Before inserting the card into the PCIe bus, please follow the detailed steps given below to install the board in your computer.

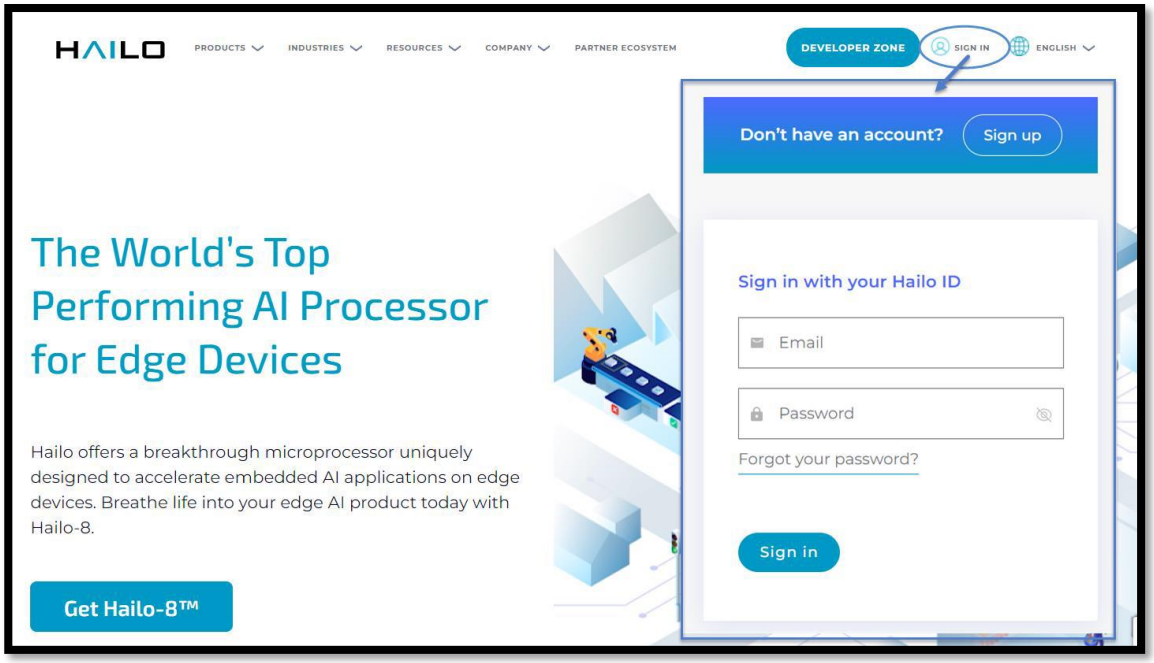
SAFETY FIRST To avoid damaging to the computer, make sure to remove any power connection before card installation.

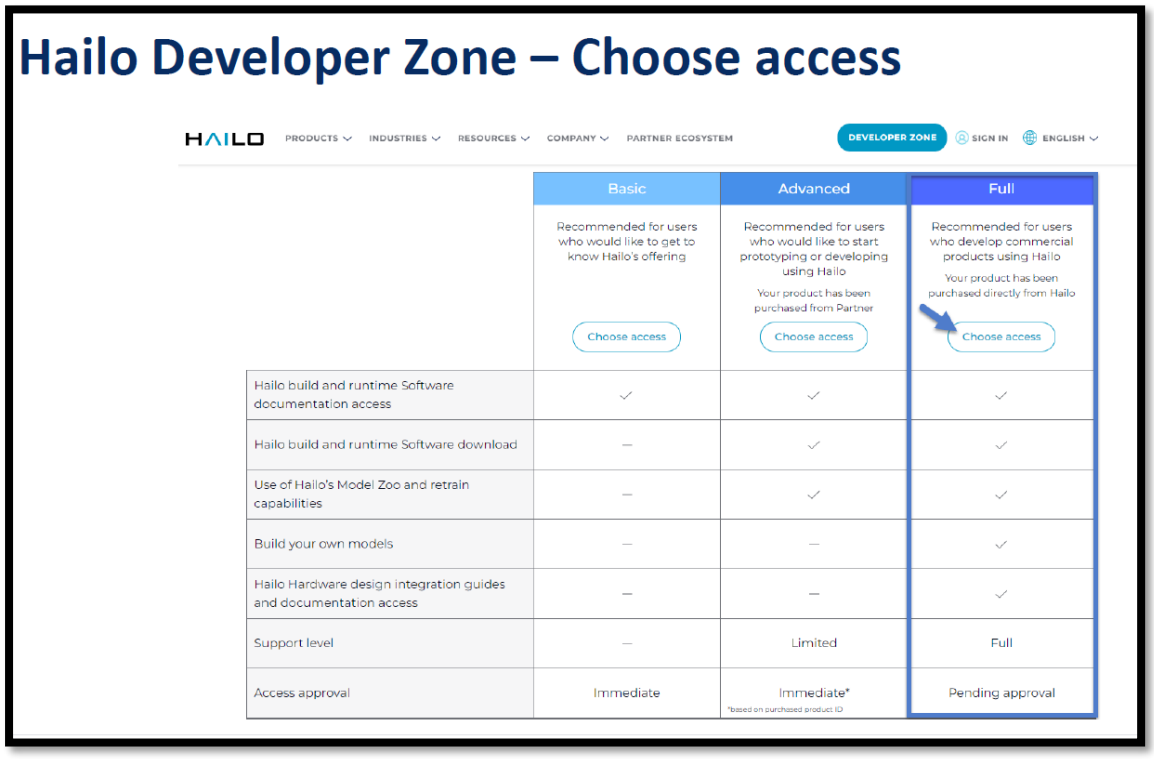
Step 1: Turn your PC's power off, and shut off the power to any peripheral.
Step 2: Remove the power plug from the plug socket.
Step 3: Remove the cover from the computer case.
Step 4: If fitted. Remove the metal cover plate on the rear of a free PCIe slot.
Step 5: Insert PCI Express AI accelerator board into the free PCIe slot
and screw it firmly on the bracket side.
Step 6: Place the cover back onto the computer.
Step 7: Insert the plug into the plug socket.



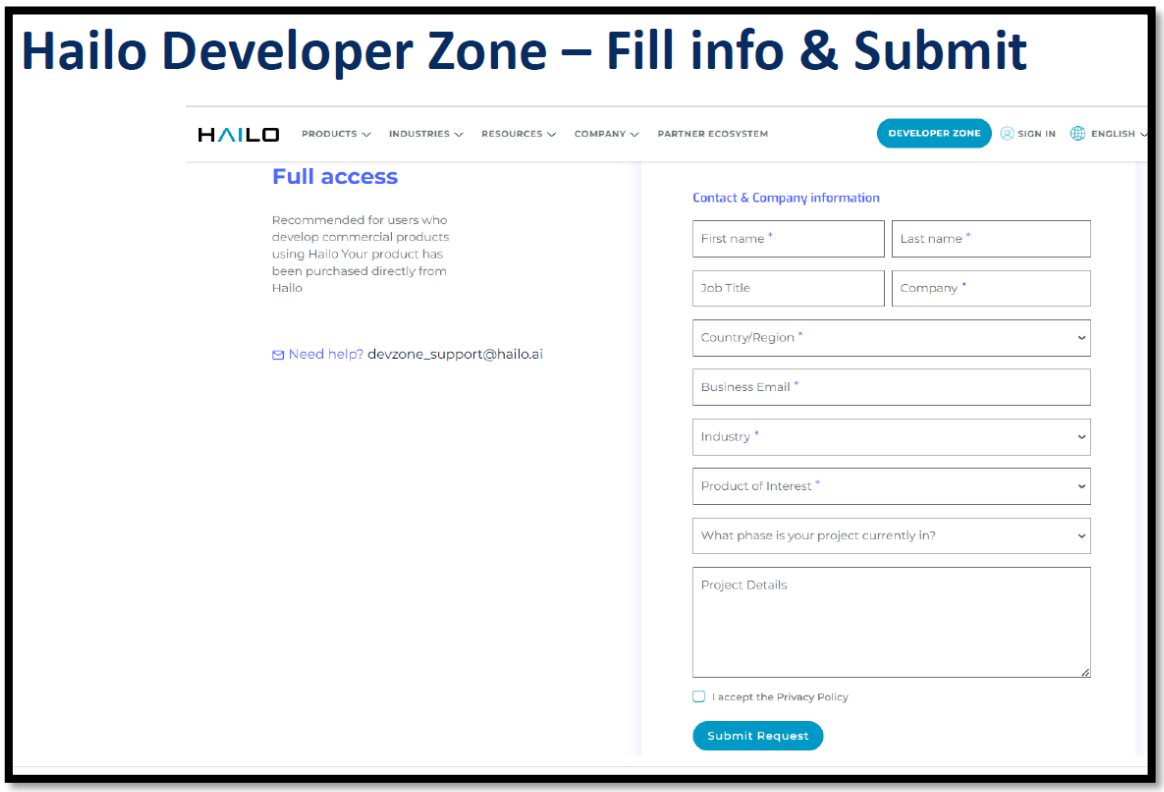
3. Hailo Developer Zone Sign Up

Visit hailo web site： https://hailo.ai

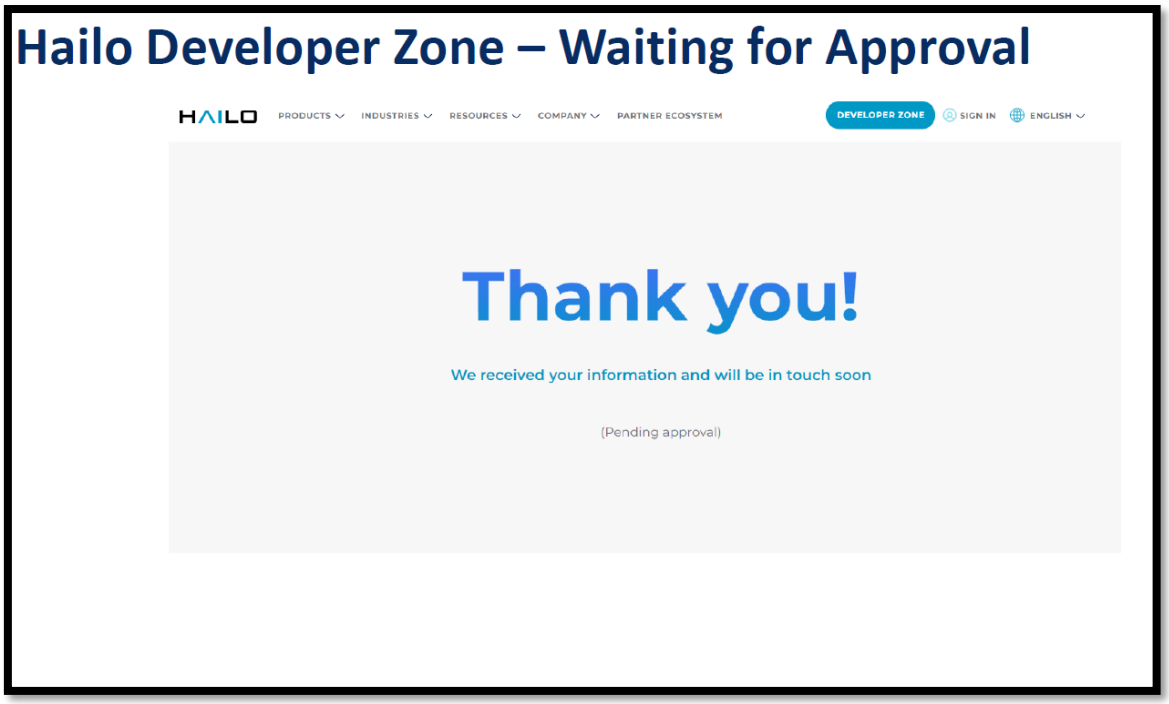




Hailo Developer Zone – Fill info & Submit

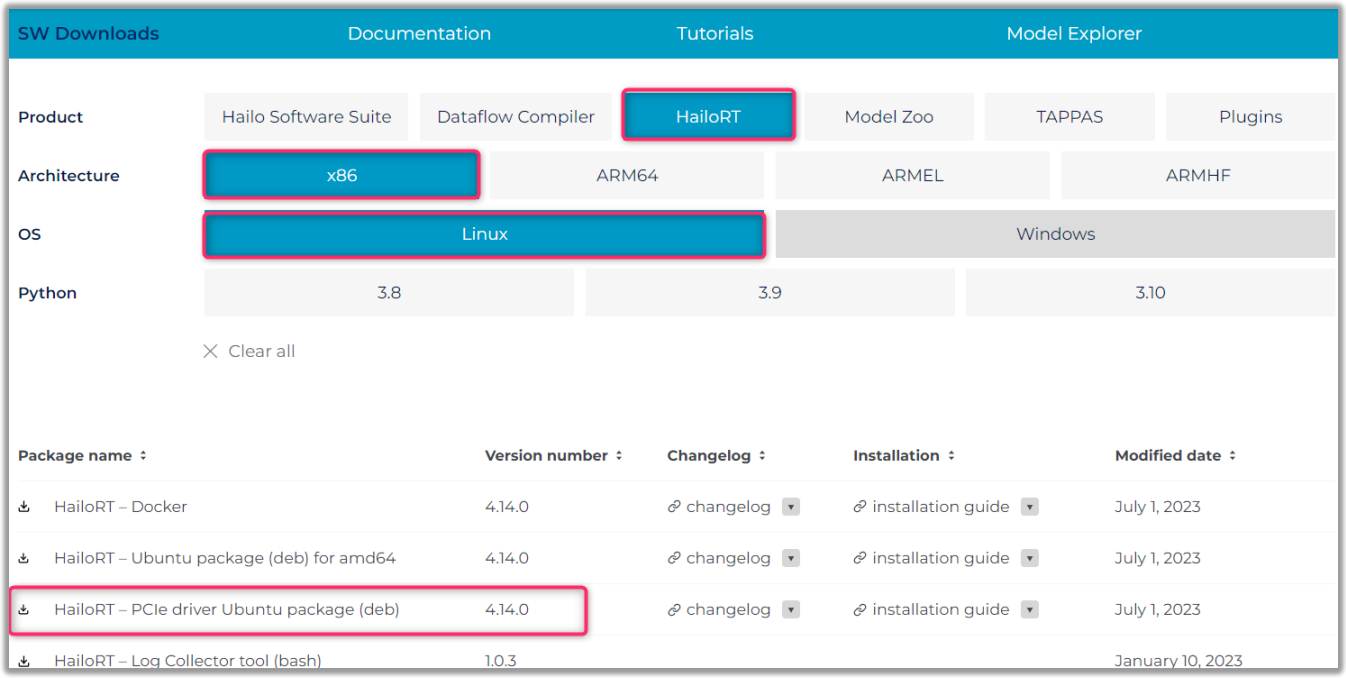


Hailo Developer Zone – Waiting for Approval



4. Driver Installation

(1) HailoRT, PCIe Driver Linux Packages Download



(2) HailoRT, PCI Driver Linux Installation

```
# install required system packages
sudo apt update
sudo apt install build-essential make cmake dpkg dkms
sudo apt install linux-headers-$(uname -r)
sudo apt install linux-image-$(uname -r)

# install driver - issuing below commands in Terminal
sudo dpkg --install hailort_pcie_driver_x.x.x_all.deb
# install HailoRT x86 64 (AMD 64 ) version
sudo dpkg --install hailort_x.x.x_amd64.deb
# reboot to apply driver installation
sudo reboot
```

5. Installation Verification

After successfully installing SUNIX AI Board into the network appliance, please follow the steps below to complete SUNIX PCIe enumeration verification process.

Step 1: Power ON the host system and log on to Linux.
Step 2: Verify that the SUNIX AI board processor is detected on the PCIe bus. From a Linux terminal, type:

Commands

(1) Check that your board is recognized by the host by running:

lspci | grep Co-processor

(2) Find Hailo virtual environment:
source hailo_platform_venv/bin/activate

(3) Verify (while in Hailo environment):

hailo fw-control identify

6. Hailo software suite overview

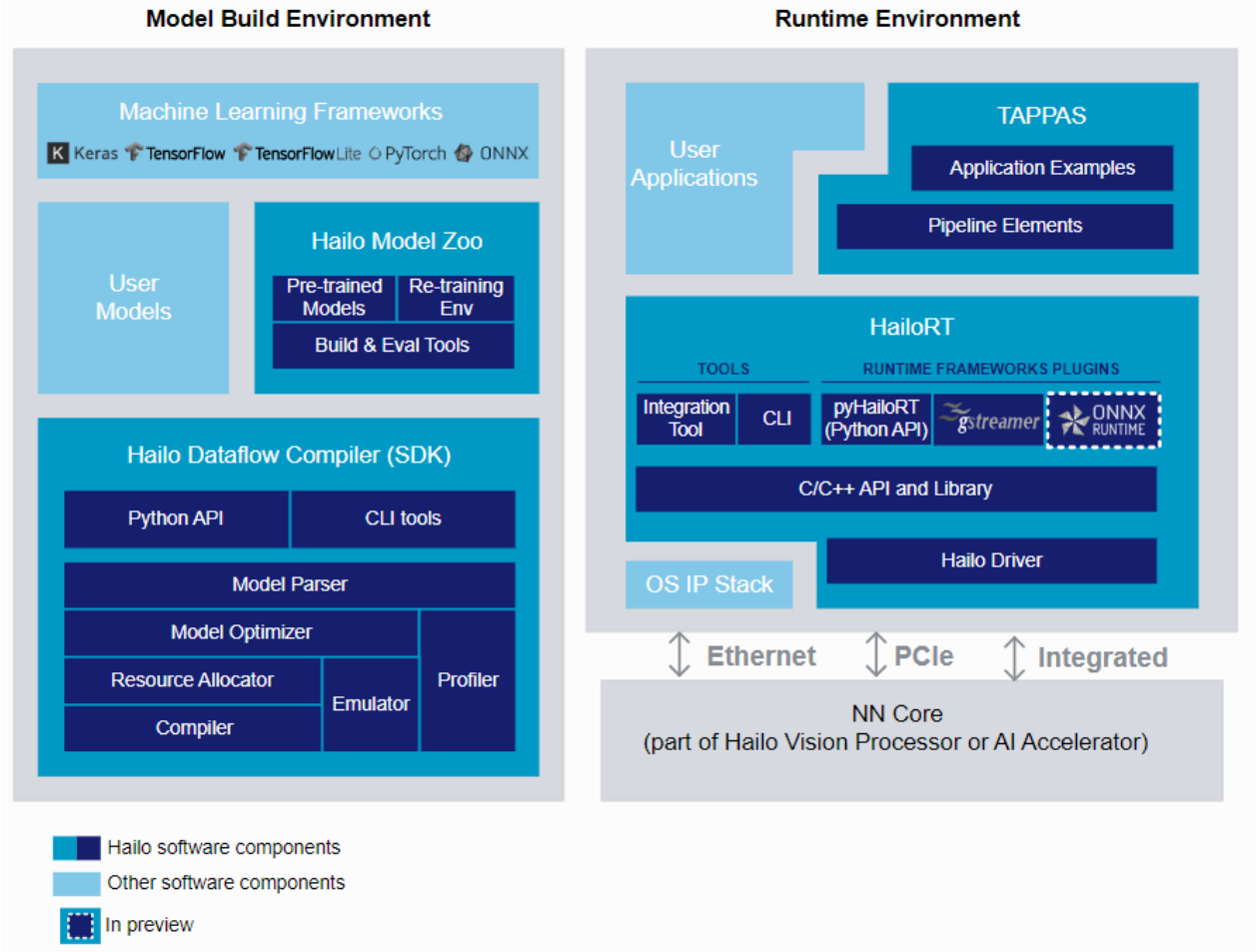
Getting Started Guide

Hailo SW products are set of frameworks and tools that enable you to compile, run and evaluate neural networks on Hailo devices:

1. **Dataflow Compiler** (Model conversion and compilation to Hailo binary format)
2. **HailoRT** (Runtime environment and driver for running networks and interacting with Hailo devices)
3. **Model Zoo** (Pre-trained models to run and evaluate on Hailo devices)
4. **TAPPAS** (Deployment framework, examples and multi-network pipelines)

Although you can install each product separately, Hailo releases a quarterly software suite in which all the product versions are aligned. Therefore, using Hailo SW Suites ensures the best compatibility.

Suite Components



Detailed block diagram of Hailo software packages

Hailo SW components are used in this manner:

- **On the Model build environment:**
  - Hailo Dataflow Compiler is used to compile a trained model to run on Hailo devices.
  - Hailo Model Zoo contains a large database of pre-trained models that are validated to work with best performance on Hailo devices.
    It also contains a retraining environment.

- **On the Runtime environment:**
  - HailoRT is used to load the compiled model to Hailo device and interact with it (using the PCIe driver).
  - TAPPAS includes complete examples and demos of using HailoRT to create full pipelines on top of Hailo devices.

Dataflow Compiler

The Dataflow Compiler API is used for compiling models to Hailo binaries. The input of the Dataflow Compiler is a trained Deep Learning model. The output is a binary file which is loaded to the Hailo device.

HailoRT

The HailoRT API is used for deploying the built model on the target device. This library is used by the runtime applications.
It implements a userspace C/C++ API that is called from the user's applications. It allows both to control the Hailo device and to send and receive data from it. It supports both the PCIe interface.
The HailoRT Python package wraps the C/C++ API and exposes a Python interface that allows to load models to the device and send and receive data from it.
It also includes a PCIe driver is required when working via the PCIe interface. It links the HailoRT library and the device. It also loads the device's firmware when working through this interface.

Finally, Hailo's Yocto layer allows the user to integrate Hailo's software into an existing Yocto environment. It includes recipes for the HailoRT library, Python package and the PCIe driver.

Hailo Model Zoo

Hailo Model Zoo provides pre-trained models for high-performance deep learning applications.

Using the Hailo Model Zoo you can measure the full precision accuracy of each model, the optimized accuracy using the Hailo Emulator and measure the accuracy on the Hailo-8 device.

Finally, you will be able to generate the Hailo Executable Format (HEF) binary file to speed-up development and generate high quality applications accelerated with Hailo-8.

The models are optimized for high accuracy on public datasets and can be used to benchmark the Hailo model optimization scheme.

TAPPAS

TAPPAS is Hailo's set of full application examples, implementing pipeline elements and pre-trained AI tasks.

Demonstrating Hailo's system integration scenario of specific use cases on predefined systems (software and Hardware platforms). It can be used for evaluations, reference code and demos:

- Accelerating time to market by reducing development time and deployment effort
- Simplifying integration with Hailo's runtime SW stack
- Providing a starting point for customers to fine-tune their applications

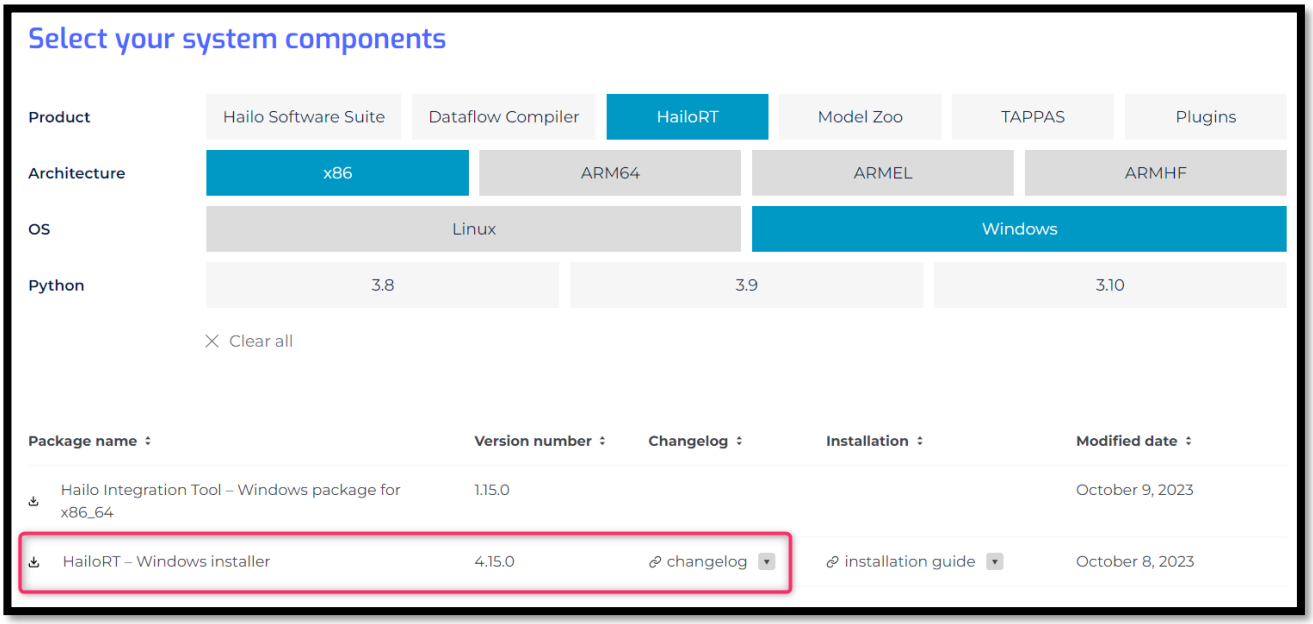
For the most updated versions, please visit Hailo developer site for the latest installation manuals and software installation packages.

7. HailoRT installation

7-1Packages download and installation

Windows OS platform

(1) Sign in Hailo's Developer Zone => https://developer.hailo.ai/developer-zone
Download windows installer

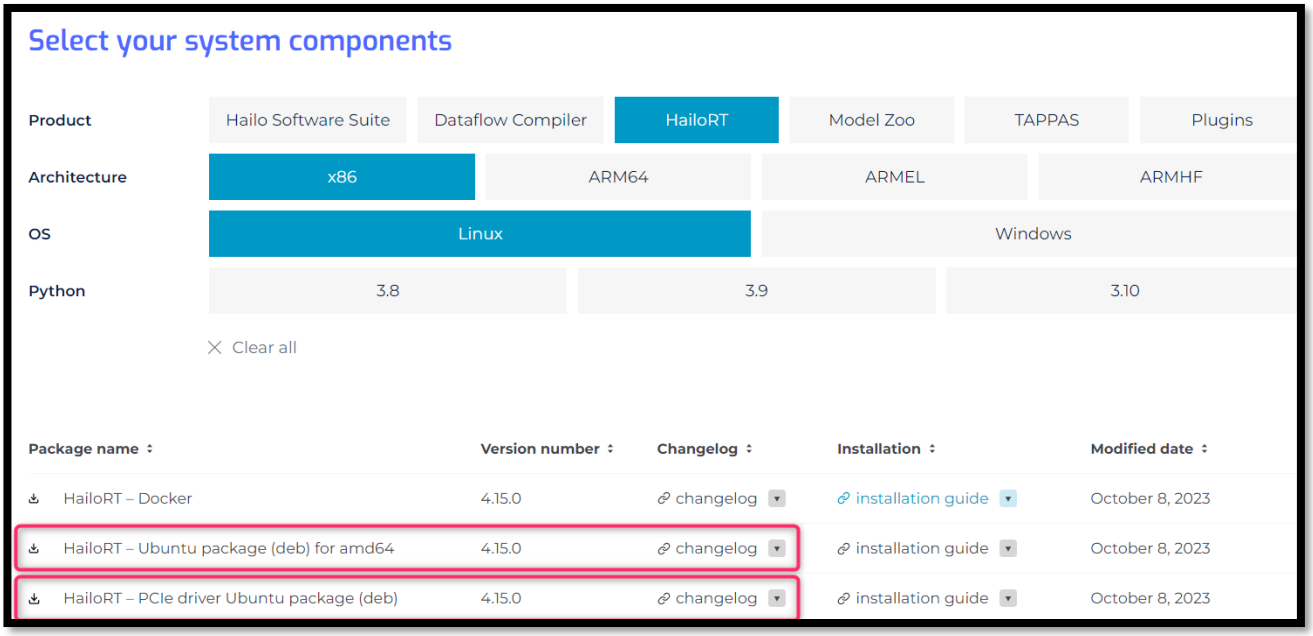


SW Products	Windows
HailoRT	Windows Installer
PCIe Driver pyHailoRT(*preview)	

(2) Execute hailort_x.x.x_windows_install.msi to install.

Linux OS platform

(1) Sign in Hailo's Developer Zone => https://developer.hailo.ai/developer-zone
Download HailoRT & PCIe-driver packages



(2) HailoRT, PCI-Driver：Linux Installation

```
# install required system packages
sudo apt update
sudo apt install build-essential make cmake dpkg dkms
sudo apt install linux-headers-$(uname -r)
sudo apt install linux-image-$(uname -r)

# install driver-issuing below commands in Terminal
sudo dpkg --install hailort-pcie-driver_x.x.x_all.deb

# install HailoRT x86 64 (AMD 64 ) version
sudo dpkg --install hailort_x.x.x_amd64.deb

# reboot to apply driver installation
sudo reboot

# After boot, continue next slice to verify the HailoRT & Driver installation
```

7-2 Device scan & identify

(1) HailoRT, Driver：Check The Installation

verify the driver installation

lspci | grep Co-processor

```
65:00.0 Co-processor: Hailo Technologies Ltd. Hailo-8 AI Processor (rev 01)
```

Figure 4: Results if PCIe IDs are up to date

```
04:00.0 Co-processor: Device 1e60:2864 (rev 01)
```

Figure 5: Results if PCIe IDs are not up to date

get help from hailortcli

hailortcli --help

```
hailo@hailo-abp3000:~/Downloads$ hailortcli --help
HailoRT CLI
Usage: hailortcli [OPTIONS] SUBCOMMAND

Options:
  -h,--help                   Print this help message and exit
  -v,--version                Print program version and exit

Subcommands:
  run                         Run a compiled network
  scan                        Shows all available devices
  benchmark                   Measure basic performance on compiled network
  measure-power               Measures power consumption
  sensor-config               Config sensor attached to the Hailo chip
  fw-config                   User firmware configuration tool
  fw-logger                   Download fw logs to a file
  fw-update                   Firmware update tool (only for flash based devices)
  ssb-update                  Second stage boot update command (only for flash based devices)
  udp-rate-limiter            Limit UDP rate
  parse-hef                   Parse HEF to get information about its components
  fw-control                  Useful firmware control operations
```

(2) HailoRT, Driver：Check The Installation (cont.)

show all available devices & the BDF (PCI Domain:Bus:Device.Function

hailortcli scan

```
abp3000@abp3000-ipc:~$ hailortcli scan
Hailo PCIe Devices:
[-] Device BDF: 0000:02:00.0
[-] Device BDF: 0000:03:00.0
abp3000@abp3000-ipc:~$
```

display device info (if only single device installed)

hailortcli fw-control identify

```
hailo@hailo-abp3000:~/Downloads$ hailortcli fw-control identify
Identifying board
Control Protocol Version: 2
Firmware Version: 4.8.1 (release,app)
Logger Version: 0
Board Name: Hailo-8
Device Architecture: HAILO8_B0
Serial Number: HLLWM20205200498
Part Number: HM218B1C2FA
Product Name: HAILO-8 AI ACCELERATOR M.2 M KEY MODULE
```

8. Compiling HailoRT

(1) Compiling from Sources-HailoRT (runtime library)

```
# install git and build packages
sudo apt update
sudo apt install build-essential make cmake dpkg dkms git curl

# download HailoRT
git clone https://github.com/hailo-ai/hailort.git

# compiling HailoRT source
cd hailort/
cmake -H. Bbuild -DCMAKE_BUILD_TYPE=Release -DHAILO_BUILD_EXAMPLES=1
sudo cmake --build build --config release --target install

# back to installation directory for HailoRT PCIe Driver installation
cd ../
```

(2) Compiling from Sources-HailoRT-PCIe Driver

```
# install git and build packages (skipped if continue from previous slice)
sudo apt update
sudo apt install build-essential make cmake dpkg dkms git curl

# download HailoRT PCI driver
git clone https://github.com/hailo-ai/hailort-drivers.git

# compiling PCI Driver source
cd hailort-drivers/ linux/pcie
make all && sudo make install
sudo modprobe hailo_pci
sudo cp 51-hailo-udev.rules /etc/udev/rules.d
sudo udevadm control –reload-rule && sudo udevadm trigger

# reboot system to apply the driver installation

sudo reboot
```

(3) **Compiling from Sources-HailoRT Example Codes**

```
# changing directory to libhailort /examples
cd hailort/hailort/libhailort/examples

# compiling HailoRT example codes
cmake -H. -Bbuild
cmake --build build

# Start from vstreams_example.c or vstreams_example.cc
# Find more example code detail in HailoRT User Guide
# 5.1 . C inference tutorial
# 5.2 . C++ inference tutorial
```