

HAPS-SX VU19P 1F Deep Learning Accelerator



Last Updates: 14th Sep 2020
E-Elements Technology Co., Ltd

Contents

Introduction	2
Equipment Requirements	2
Quick start guide	2
Check HAPS SMF SD card content	2
Connecting everything	3
Insert HAPS SMF SD card.	3
Insert HAPS-SX Project by USB or USB reader.	3
Switch 000001	3
Connecting HAPS-SX power cable.	4
Connecting FireFly loopback cable assembly.	4
Connecting mini DP to HDMI monitor.	4
Connecting Jtag to PC.	5
Power ON HAPS-SX and check DONE light.	5
Connecting HAPS-SMF power supply.	6
Open MobaXterm and connect to SMF	6
Log in	6
Select demo to run	7

1. Introduction

HAPS is a product, with the fastest performance, highest debug productivity and enterprise scalability to accelerate software development, system validation and verification.

Today, we will use HAPS-SX to demo Deep Learning Accelerator (DLA). There are two labs in this section. One is classification, and the other is object detection.

2. Equipment Requirements

Machine

- ☐ [HAPS-SX](#)
- ☐ USB reader
- ☐ SD card *2 (burn already)
- ☐ screen
- ☐ PC

Daughter board

- ☐ HAPS-SMF

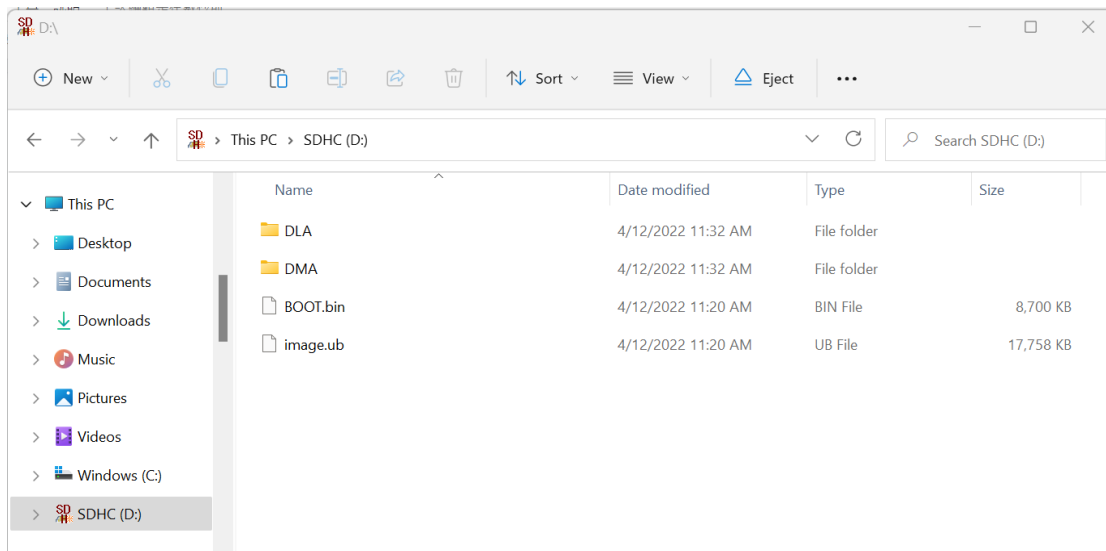
Cable

- ☐ [HAPS-SX power cable](#)
- ☐ [FireFly loopback cable assembly *2](#)
- ☐ [mini DP to HDMI](#)
- ☐ [micro USB to USB](#)
- ☐ SMF 4EV power supply

3. Quick start guide

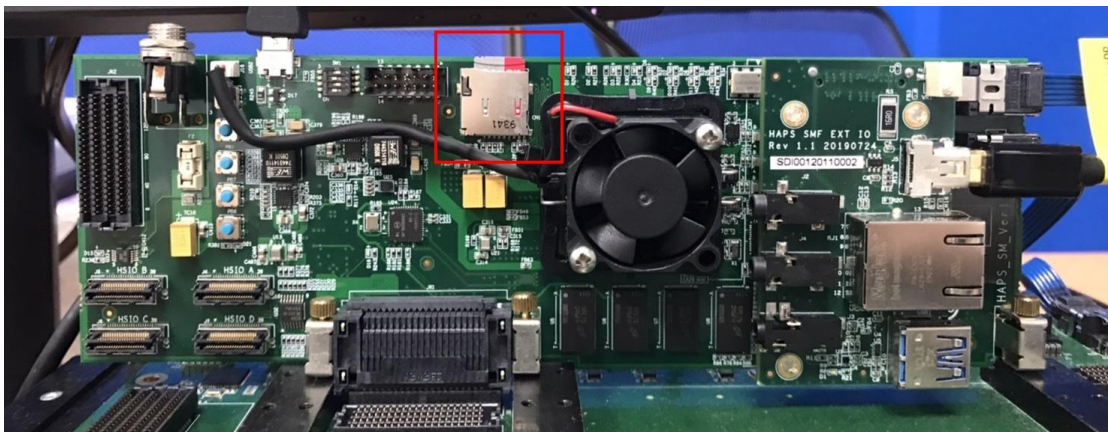
a. Check HAPS SMF SD card content

Copy content in the DLA folder to the root file.



b. Connecting everything

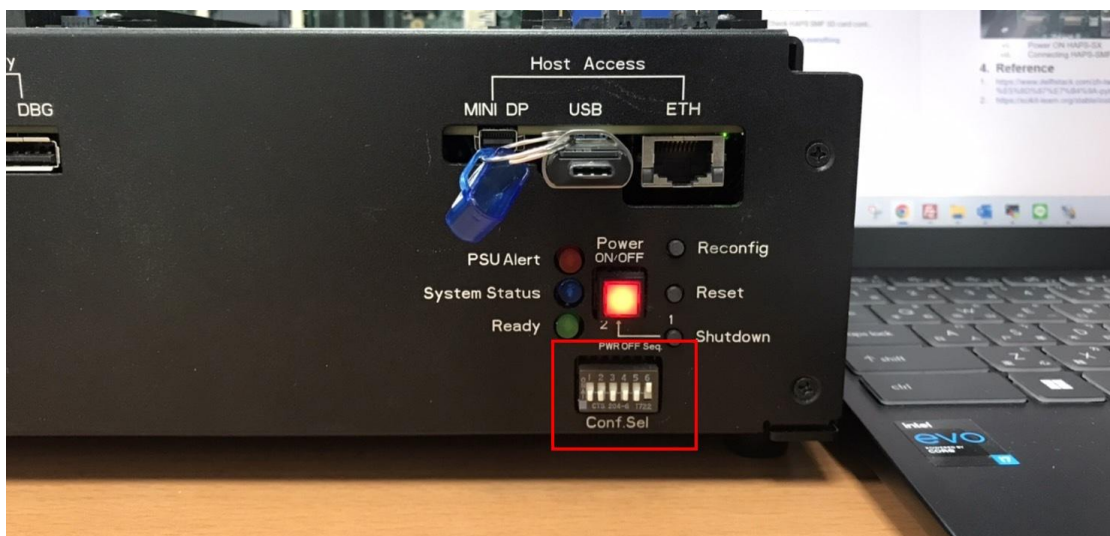
i. Insert HAPS SMF SD card into JX12.



ii. Insert HAPS-SX Project by USB or USB reader.



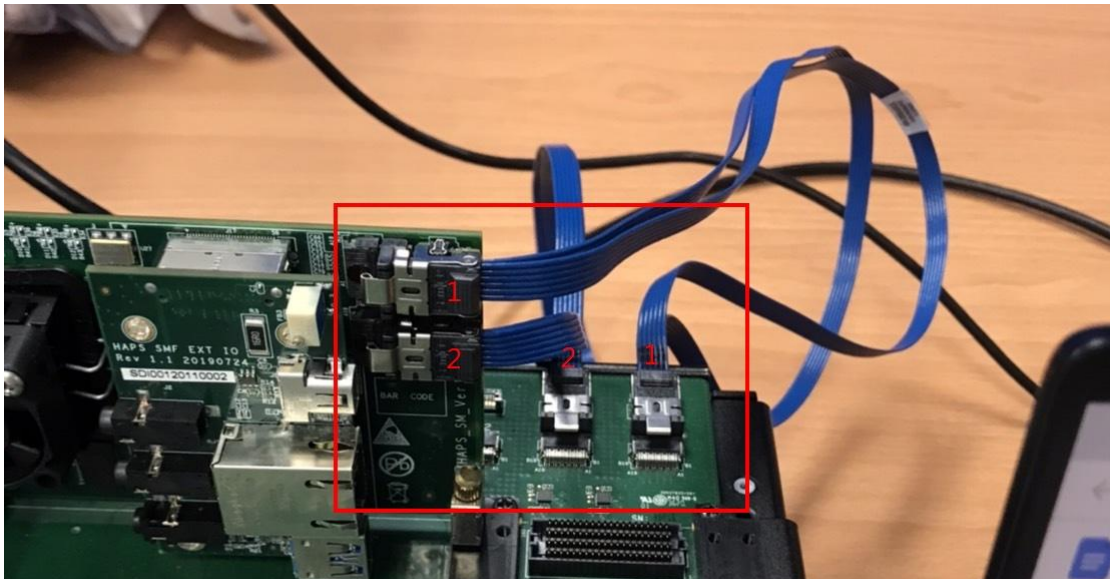
iii. Switch 000001



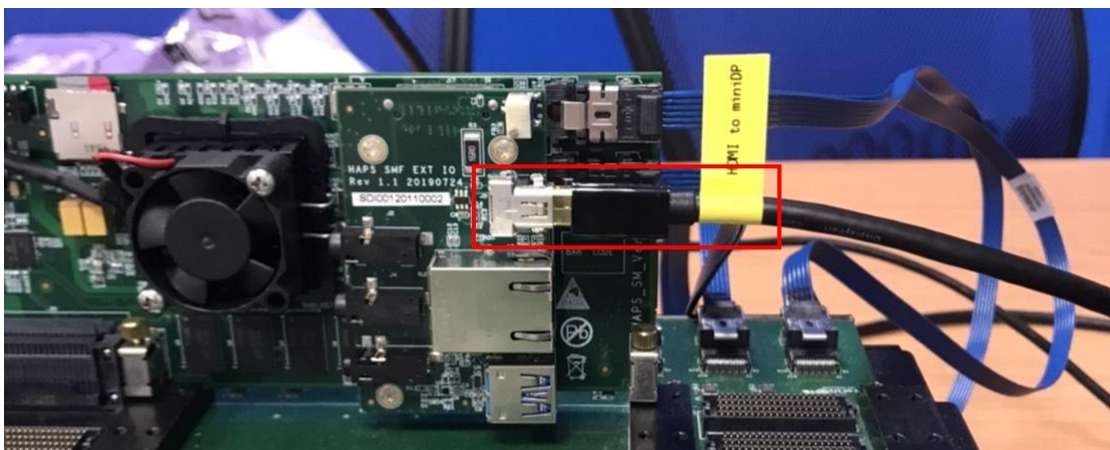
iv. Connecting HAPS-SX power cable.



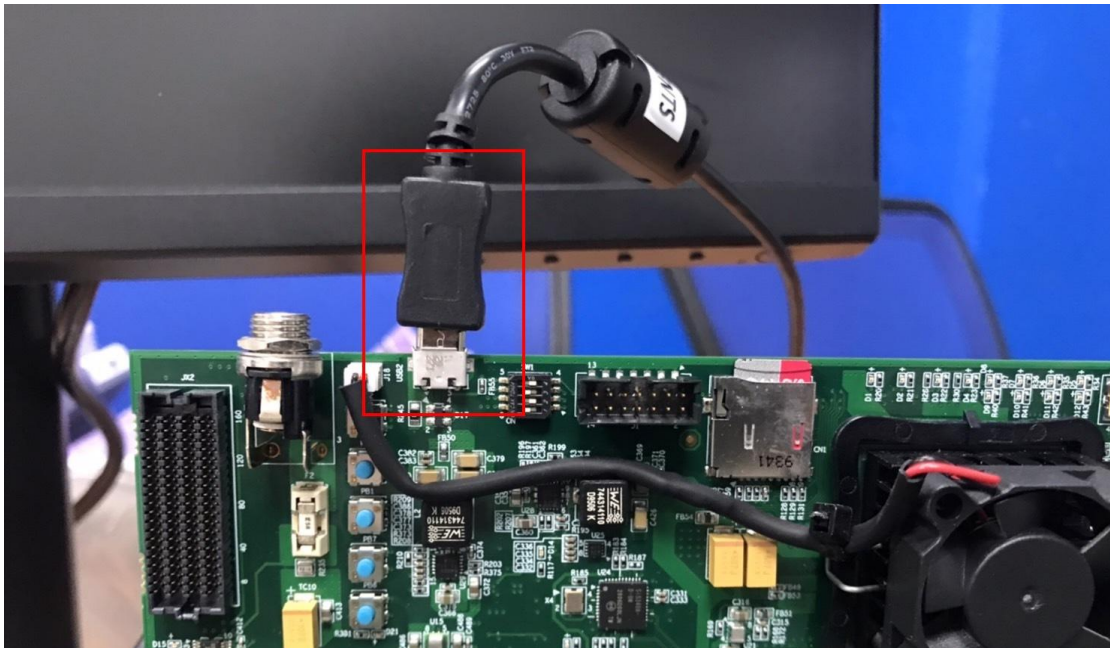
v. Connecting FireFly loopback cable assembly.



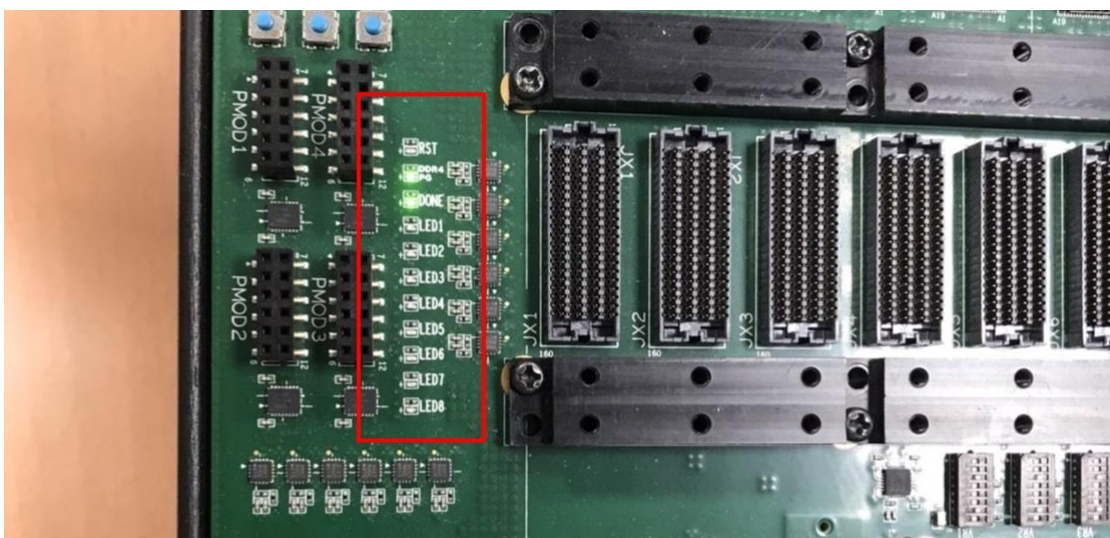
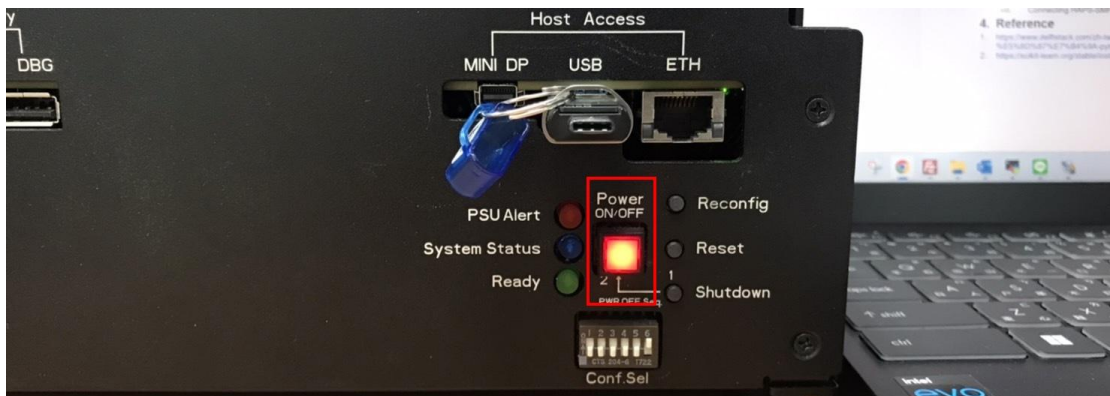
vi. Connecting mini DP to HDMI monitor.



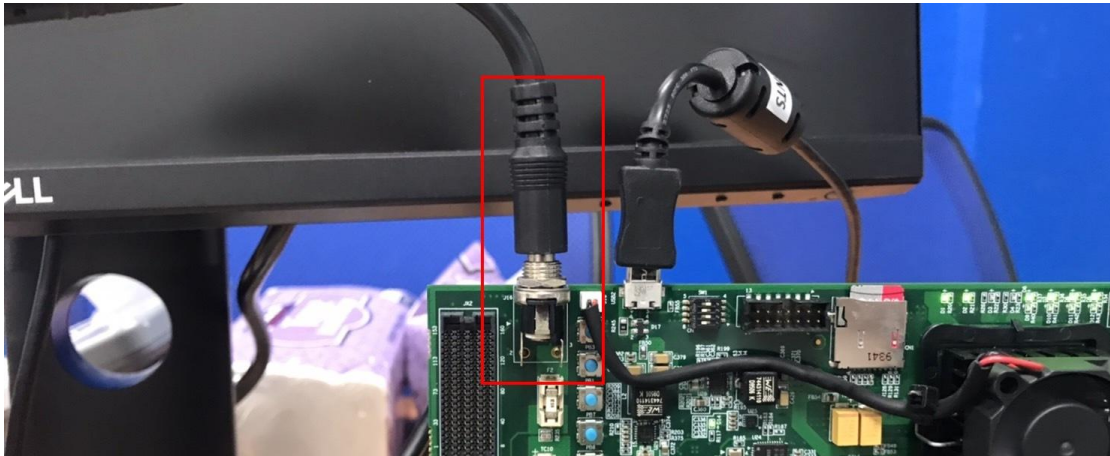
vii. Connecting Jtag to PC.



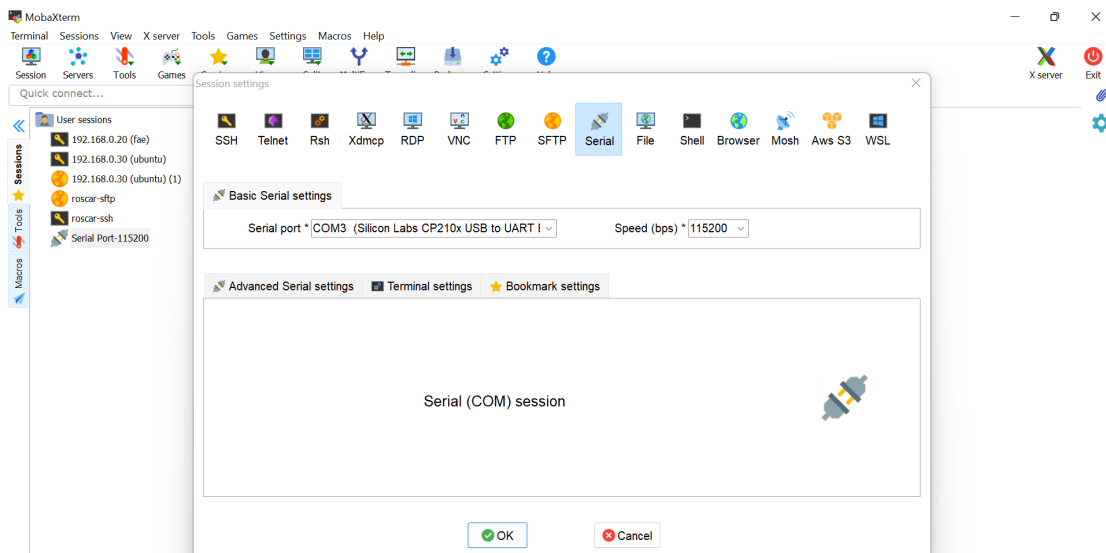
viii. Power ON HAPS-SX and check DONE light.



ix. Connecting HAPS-SMF power supply.



c. Open MobaXterm and connect to SMF



d. Log in
login: root
password: root


```

PetaLinux 2019.1 xilinx-zcu104-2019_1 /dev/ttyPS0
xilinx-zcu104-2019_1 login: root
Password:
[ 202.644158] [drm] Initialized nvidia_small 0.0.0 20190122 for a0030000.nv_small_64_v07 on minor 1
[ 202.656133] nvidia_drm_probe, allocation physical continuous memory@600000000 done, 0
root@xilinx-zcu104-2019_1[/work/nvdla_test]# ffserver version 3.4.6 Copyright (c) 2000-2019 the FFmpeg developers
built with gcc 7.4.1 (Linaro GCC 7.4-2019.02) 20181213 [linaro-7.4-2019.02 revision 56ec6f0b99cc167ff0c2f8e1a2eed33b1edc05d4]
configuration: --prefix=/home/akio/work/zcu104_nvdla_demo/zcu104_install --cross-prefix=/opt/aarch64-toolchains/linaro-7.4.1/bin/aarch64-linux-gnu- --enable-cross-compile --enable-rpath --enable-shared --enable-network --disable-doc --disable-htmlpages --disable-manpages --disable-podpages --disable-txtpages --disable-iconv --disable-xlib --disable-libopenh264 --disable-run-time-cpudetect --disable-avdevice --disable-swresample --disable-swscale --disable-avfilter --disable-avresample --disable-devices --disable-encoders --disable-hwaccels --disable-v4l2_m2m --disable-iconv --disable-sdl2 --disable-xlib --disable-cuda --disable-cuvid --disable-nvenc --disable-libxcb --disable-libxcb-shm --disable-libxcb-shape --disable-libxcb-xfixes --disable-decoders --enable-decoder-h263 --enable-decoder-h264 --enable-decoder-hevc --enable-decoder-mpeg4 --enable-decoder-mjpeg --disable-muxers --enable-muxer-rtp --enable-muxer-rtsp --enable-muxer-h263 --enable-muxer-h264 --enable-muxer-m4v --enable-muxer-mjpg --enable-demuxers --enable-demuxer-ftp --enable-demuxer-rtsp --enable-demuxer-h263 --enable-demuxer-h264 --enable-demuxer-m4v --enable-demuxer-mjpg --enable-parser-h263 --enable-parser-h264 --enable-parser-hevc --enable-parser-mpeg4 --enable-parser-mjpeg

```

e. Select demo to run

- i. `./classification.sh`
- ii. `./yolov1.sh`

