Quick Start Guide for Running MLPerf-Tiny Benchmark on AndesCore™-embedded Platforms

1. Prerequisite

- One PC installed with Windows 10 and a Linux virtual machine of Ubuntu-18.04 LTS
  - Unbuntu-18.04 – for building benchmark executables
  - Windows 10 – for running MLPerf Tiny benchmarks

- How to get the FPGA board for evaluation
  - Please contact sales@andestech.com for purchase the FPGA board. Detailed information is described in JSON file (path: https://github.com/mlcommons/submissions_tiny_0.7/tree/main/closed/Andes/systems)

- EnergyRunner, the benchmark framework used by MLPerf Tiny
  With a license for EnergyRunner, download the benchmark runner `EnergyRunner_MLPerfTiny_WIN10_3.0.10.exe` from the EEMBC website and install it on Windows 10. Please refer to the introduction on the website: https://github.com/eembc/energyrunner. After EnergyRunner installed, edit `.eembc.ini` in your home directory as follows:

```
root=C:\Users\YOUR_NAME
dut-baud=115200
dut-boot-mv=3000
default-timeout-ms=5000
emon-drop-thresh-pct=0.1
timestamp-hold-us=50
umount-on-error=true
use-crlf=true
use-visa=false
n6705-set-vio=false
disable-mute=false
```

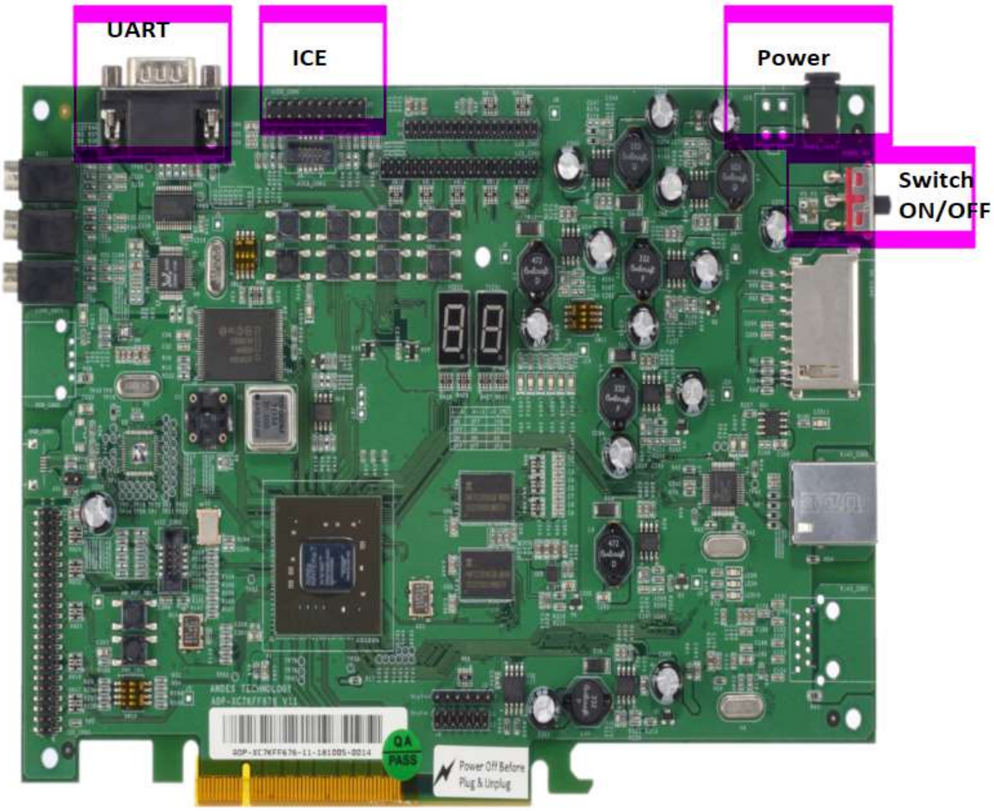
2. Tutorial

This tutorial contains the following instructions on running MLPerf Tiny benchmarks on platforms integrated with AndesCore™ D25F/D45/NX27V:

1. Setting up the hardware, including the target board and Andes ICE box
2. Downloading build scripts and datasets for MLPerf Tiny benchmarks
3. Compiling the benchmark code
4. Loading the benchmark executable to the target board and executing it

2.1. Setting up the hardware

AndeShape™ ADP-XC7K160/410 pre-integrated with AndesCore D25F/D45 is used as an example here. Please take note of the following sockets on the development board first.



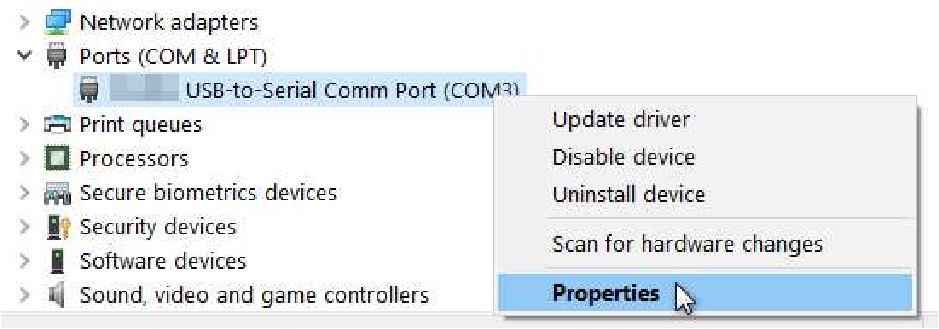
Then, proceed as follows:

Step 1 Prepare a 5v adapter and connect it to the socket.

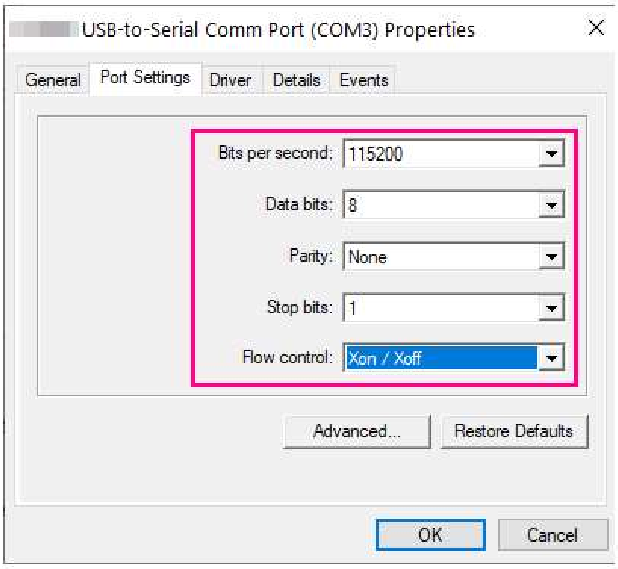
Step 2 Switch on the toggle and see the on-board LED light is turned on.

Step 3 Set up a COM port for the UART outputs.

1. Prepare a USB-to-RS232 cable and install its driver on Windows 10.
2. Use the cable to connect a USB port on your PC to the UART socket of the evaluation board.
3. The Windows Device Manager shows the newly-installed device appears as a USB-to-Serial Port (COM#). Right-click on the COM port and select “Properties” from the pull-down menu.



4. Select the Port Settings tab on the invoked dialog to specify communication parameters as follows and click “OK”.



Bits per second: 115200

Data bits: 8

Parity: None

Stop bits: 1

Flow control: Xon / Xoff

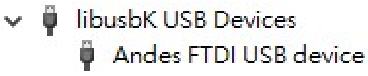
The COM port is ready for testing.

Step 4 Set up Andes ICE box, install its driver and run the controller program, ICEman:

1. Use a USB-to-ICE cable to connect a USB port on your PC to the ICE socket on the evaluation board.

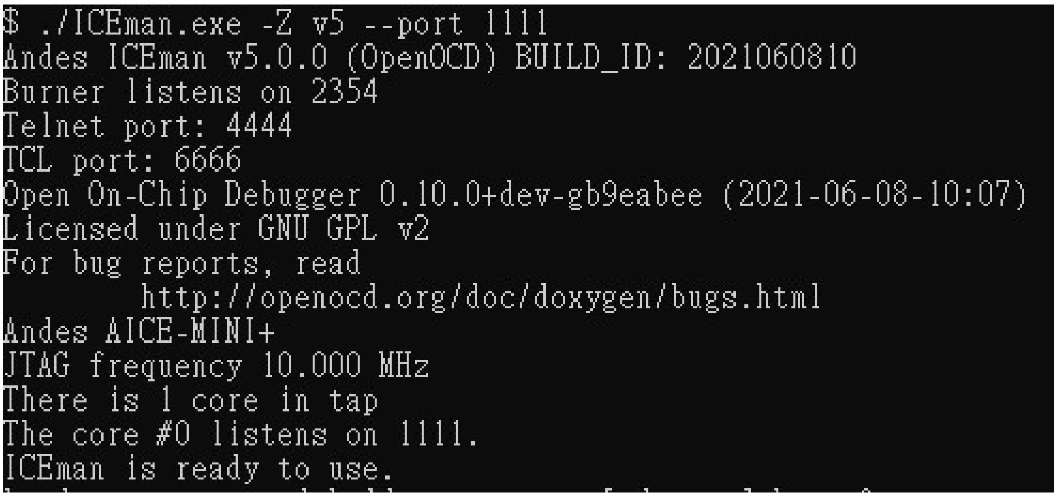
2. Download from Andes GitHub: https://github.com/andestech/Andes-Development-Kit/releases and unzip ice.zip on Windows 10. This is to obtain Andes ICE driver in the package.

3. Install Andes ICE driver by running the execution file install_driver.exe under unzip_path\ice\libusb-AICE-driver; The Andes ICE connected to your board now appears as Andes FTDI USB device like below on the Device Manager.



```
libusbK USB Devices
    Andes FTDI USB device
```

4. In the terminal, change the directory to unzip_path\ice; Next, execute Andes ICE controller program (ICEman.exe) and specify a port for GDB connection like below.



```
$ ./ICEman.exe -Z v5 --port 1111
Andes ICEman v5.0.0 (OpenOCD) BUILD_ID: 2021060810
Burner listens on 2354
Telnet port: 4444
TCL port: 6666
Open On-Chip Debugger 0.10.0+dev-gb9eabee (2021-06-08-10:07)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
Andes AICE-MINI+
JTAG frequency 10.000 MHz
There is 1 core in tap
The core #0 listens on 1111.
ICEman is ready to use.
```

ICEman is ready to use at this point.

2.2. Downloading build scripts for benchmarks

The following steps need to be performed in Linux environment (Ubuntu 18.04).

Step 1 Issue as follows to download the latest MLPerf Tiny suite.

```
git clone https://github.com/mlcommons/submissions_tiny_0.7
```

The destination folder is designated as MLPERF_ROOT in subsequent steps.

Step 2 Find Andes submission in MLPERF_ROOT/…/Andes/code. The directory includes four benchmark folders (i.e., anomaly_detection, image_classification, keyword_spotting and person_detection). A script for downloading and building benchmark files is provided in /script folder under respective benchmark folders.

Step 3 Refer to MLCommons Tiny GitHub to obtain the datasets kws01, ad01, ic01 and vww01 for benchmarks Andes supports (keyword spotting, anomaly detection, image classification, and person detection) and copy them to the working directory of EnergyRunner such as the folder `C:\Users\YOUR_NAME\eembc\runner\benchmarks\ulp-mlperf\datasets`.

2.3. Compiling the benchmark code

The following steps need to be performed in Linux environment (Ubuntu 18.04).

Step 1 Prepare the build environment:

1. Issue as follows to install python packages required for the build.

```
pip install -r GitHub_PATH/tensorflow/tflite-micro/blob/main/third_party/requirements.txt
pip install -r GitHub_PATH/tensorflow/tflite-micro/blob/main/tensorflow/lite/micro/examples/magic_wand/train/requirements.txt
```

2. Download the toolchain for your AndesCore (`nds32le-elf-newlib-v5d.txz` for D25F/D45 and `nds64le-elf-newlib-v5d.txz` for NX27V) from the GitHub page of Andes Linux toolchains: https://github.com/andestech/Andes-Development-Kit/releases/tag/ast-v5_0_0-release-linux or https://github.com/andestech/nds-toolchain/releases/tag/ast-v5_0_0-release-linux and issue as follows to decompress it:

```
cd /MLPERF_ROOT
xz -d [nds32le-elf-newlib-v5d.txz|nds64le-elf-newlib-v5d.txz]
tar xvf [nds32le-elf-newlib-v5d.tar|nds64le-elf-newlib-v5d.tar]
```

3. Find `config.ini` in `MLPERF_ROOT/…/Andes/code` and modify the following options to configure the benchmark code for your target platform:

   - `opt`
     Set it to `1` to enable optimization in compilation.
   - `cpu`
     The available parameters include `d25`, `d45` and `nx27v`. Be sure to specify with lowercase letters.
   - `run_mode`
     The available modes include `release` and `debug`. Be sure to

use lowercase letters when specifying.

- GITHUB_TFLITE_PATH

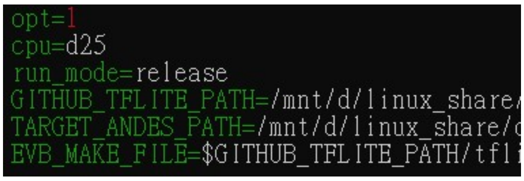
  This is to specify a destination directory for the download of TfLite-micro source from GitHub.

- TARGET_ANDES_PATH

  This is to specify the location of Andes TfLite-micro code (andescore_d25.tgz/andescore_d45.tgz/andescore_nx27v.tgz), which will be merged with the downloaded TfLite-micro source from GitHub. The default path to the Andes code is MLPERF_ROOT/…/Andes/code (i.e. Andes submission folder).

- EVB_MAKE_FILE

  No need to care about this option



```
opt=1
cpu=d25
run_mode=release
GITHUB_TFLITE_PATH=/mnt/d/linux_share/
TARGET_ANDES_PATH=/mnt/d/linux_share/c
EVB_MAKE_FILE=$GITHUB_TFLITE_PATH/tfli
```

4. Set environment variables by exporting the toolchain directory to $PATH.

   export PATH=$PATH:/MLPERF_ROOT/[nds32le-elf-newlib-v5d | nds64le-elf-newlib-v5d]/bin

Step 2 Execute the script for the desired benchmark to build the benchmark executable.

1. Change the current directory to the script folder of the desired benchmark (MLPERF_ROOT/…/Andes/code/[anomaly_detection|image_classification|keywordspotting|person_detection]/script).

2. Run the script in the folder to download and build files for the benchmark.

   bash ./[build_ad.sh|build_ic.sh|build_kws.sh|build_vww.sh]

3. After the compilation is done, an execution file “mlperf_libnn” will be generated for the desired benchmark.

   For example, with config.ini specified as follows,

```
opt=1
cpu=d25
run_mode=debug
GITHUB_TFLITE_PATH=/mnt/d/user_path/tflite-micro-path-evb
…
```

the executable “mlperf_libnn” will be generated in /mnt/d/user_path/tflite-micro-path-evb/tflite-micro/tensorflow/lite/micro/tools/make/gen/andes_evb_rv32p_debug/bin.

2.4. Load and run the benchmark executable

Step 1 Reference below to Load the benchmark executable to your target board in Linux environment (ubuntu-18.04).

```
GDB
target remote WIN10_IP:PORT_NUMER
reset-and-hold
file ./MLPERF_LIBNN
load
continue
```

[Where]

GDB is riscv32-elf-gdb for D25F/D45 and riscv64-elf-gdb for NX27V.

WIN10_IP is the IP address of your Windows 10 PC.

PORT_NUMBER is the port number designated for GDB connection when initiating ICEman (see Section 2.1, Step 4).

MLPERF_LIBNN is the benchmark executable file

Step 2 Run EnergyRunner on Windows 10 and refer to the instructions on its GitHub page: https://github.com/eembc/energyrunner to test the benchmark on the target platform integrated with AndesCore D25F/D45/NX27V.