FLIK

Getting Started Guide

terasIC
www.terasic.com

Contents

Chapter 1

Overview

This chapter provides an overview of this guide.

1.1 General Description

This document will describe how to quickly run a face recognition demo using FLIK in a Linux environment. The content will include the driver, software that needs to be installed. It allows users to quickly experience the acceleration ability of deep learning inference brought by FPGA hardware platform:

1. Install Thunderbolt3 driver
2. Install Intel® PAC and the Intel® Programmable Acceleration Card Stack
3. Install Intel® Distribution of OpenVINO™ toolkit for Linux* with FPGA Support
4. Install and verify the Terasic FLIK demo package

Before you start using the FLIK, users need to prepare some Development and Target Platform. To use FLIK with Linux, the required PC hardware and Linux Operating System can be obtained from the link below.

https://docs.openvinotoolkit.org/latest/_docs_install_guides_installing_openvino_linux.html

It is important to note that because the transmission and communication interface between FLIK and host PC is **Thunderbolt 3**. User's PC must have Thunderbolt 3 interface to connect with FLIK.

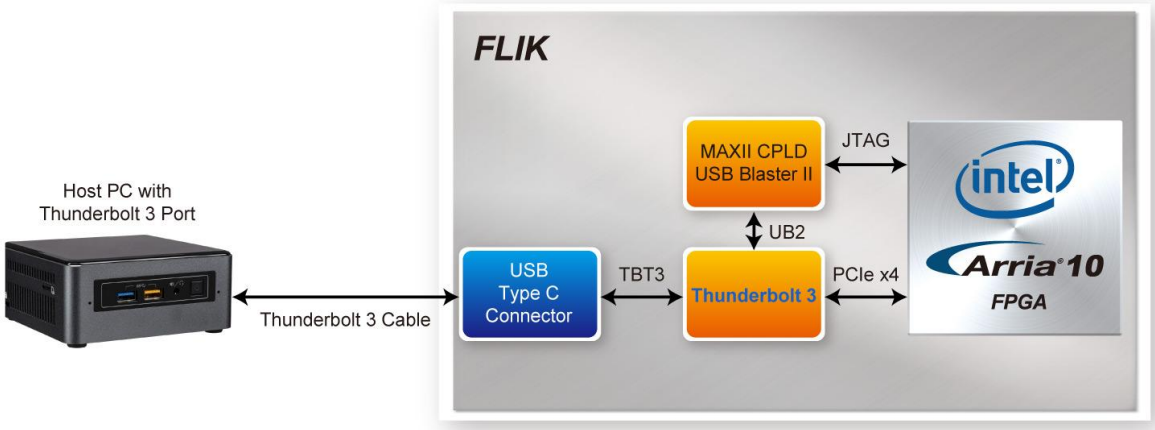
In addition, in this document, the Linux distribution we use is Ubuntu 16.04.6 LTS (64-bit). Note, for many Linux distributions of Ubuntu 16.04, Ubuntu 16.04.6 can install Thunderbolt3 driver normally, which is the ultimate system requirement for using FLIK.

Chapter 2

Install Thunderbolt3 driver

Thunderbolt 3 is unique communication path between the Host PC and FLIK. As shown in below, the FLIK use thunderbolt3 port to connect the PCIe and USB Blaster II interface to the FPGA. For the OpenCL and OpenVINO applications with the Intel FPGA, the PCIe bus is the main interface for communicating and transferring data with the host PC. The thunderbolt3 port on the FLIK allows the user to build such connection for the FPGA and the host PC via a thunderbolt3 cable.
In addition, the thunderbolt3 on the FLIK also provides an USB interface to connect to the USB Blaster II circuit on the board, providing a JTAG interface for the user to program the FPGA or configuration device through the host PC.

Therefore, a host PC equipmented with thunderbolt 3 port is required to work with the FLIK. This chapter will show user how to setup thunderbolt 3 connection between the Host PC and the FLIK for the first time.



2.1 Hardware Requirement

- A Host PC with Thunderbolt 3 Port is required to perform FLIK function tests. The PC should contain the features in below:

- Built-in Thunderbolt 3 Port or with Thunderbotl3 Card Installed.
- Linux Installed(In this document, we recommend you to install Ubuntu 16.04.6 LTS (64-bit))

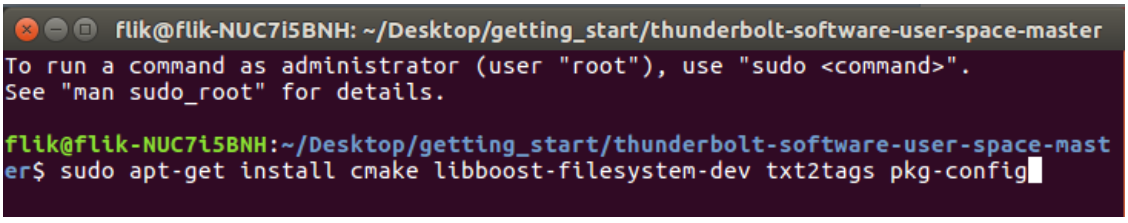
- A FLIK kit
- A Thunderbolt 3 Cable as shown in below.

2.2 Install Thunderbolt 3 Driver

1. Download the Thunderbolt user-space components in the link below and save it to your computer:
   https://github.com/intel/thunderbolt-software-user-space

2. You also need a C++ compiler with C++14 support. You can install the dependencies with the following command:

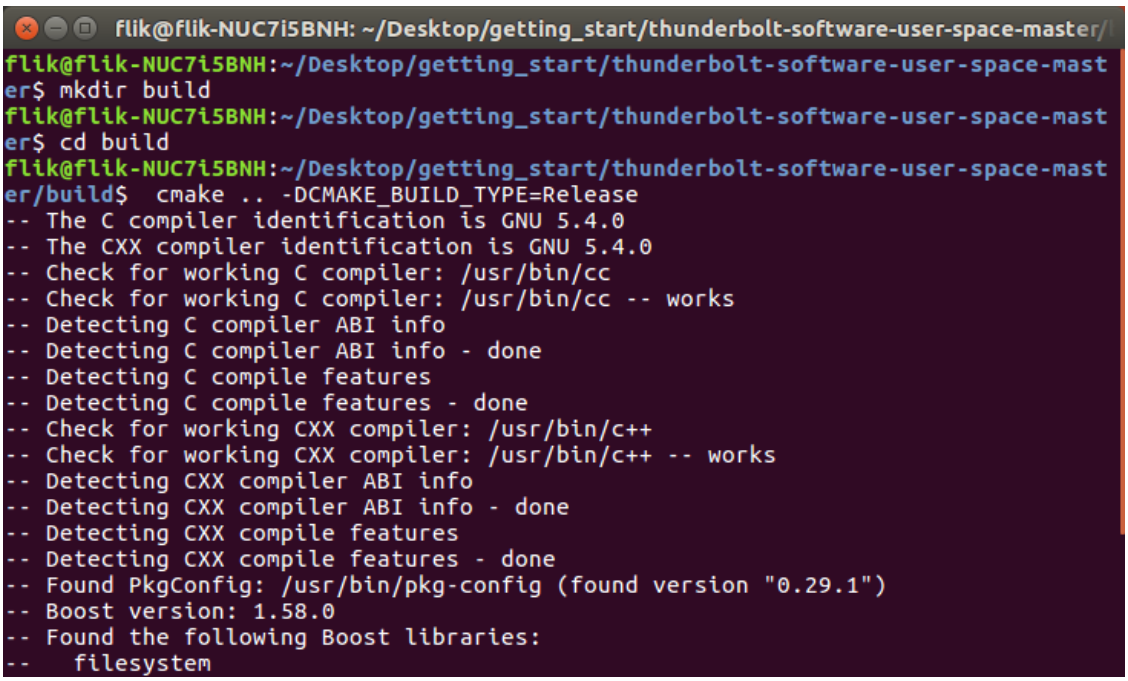
   ```
   sudo apt-get install cmake libboost-filesystem-dev txt2tags pkg-config
   ```



3. Unzip the downloaded file ***thunderbolt-software-user-space-master.zip***

4. Open terminal and locate the unzipped directory.
5. Execute the following command to install Thunderbolt 3 driver:

   - mkdir build
   - cd build
   - cmake .. -DCMAKE_BUILD_TYPE=Release
   - cmake --build .
   - sudo cmake --build . --target install

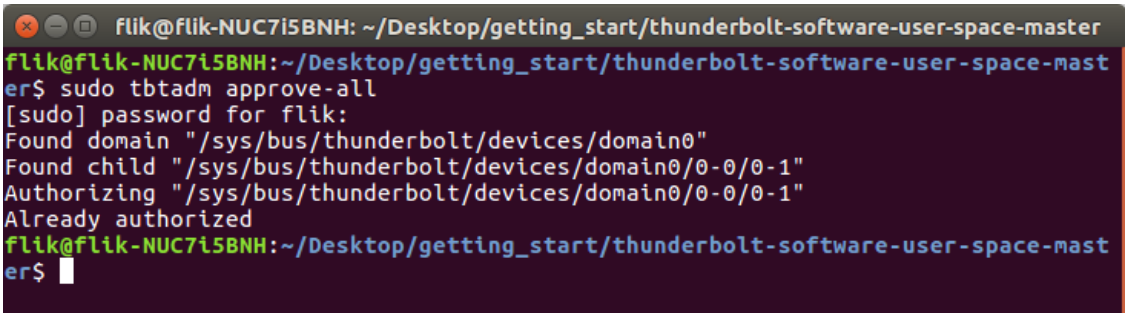


6. Next, verify whether the Host PC has detected the PCIe device after connecting to the FLIK via the Thunderbolt 3 cable.
7. Plug the DC 12V power adapter to the FLIK to power on the FLIK.

8. Connect the Host PC to the FLIK with a Thunderbolt3 cable.

9. Approve all currently connected Thunderbolt devices that aren't authorized yet and (if --once wasn't specified) add them to PC:

```
sudo tbtadm approve-all
```

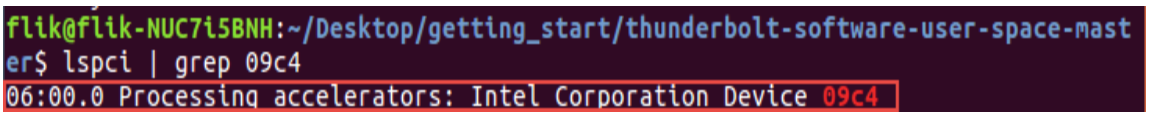


10. Execute the following command to verify whether the host PC has detected the PCIe device :

    lspci | grep 09c4

The result should be returned:

    06:00.0 Processing accelerators: Intel Corporation Device 09c4



That shows the driver of the Thunderbolt3 has been installed on your host PC correctly and the PCIe device in the FLIK is detected by the host PC.

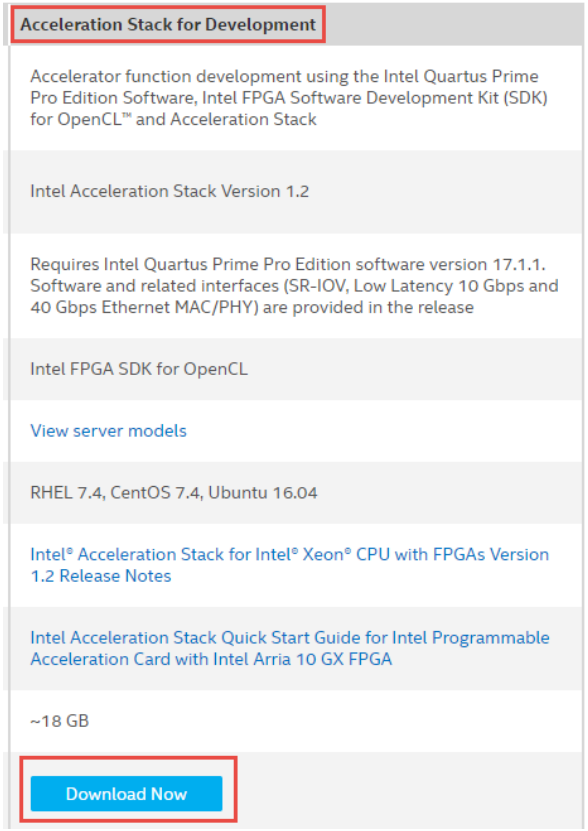
Chapter 3

Install Intel Acceleration Stack

This chapter describes how to install the Intel® Acceleration Stack.

1. Go to the link below download Intel® Acceleration Stack Version 1.2, please download "**Acceleration Stack for Development**". Users do not need to modify the factory code in the serial flash if the FLIK, or do not use the Quartus tool. You can choose "**Acceleration Stack for Runtime**" version to install. Note that in the following steps, you need to pay attention to the difference between the file name and the path name.

https://www.intel.com/content/www/us/en/programmable/products/boards_and_kits/dev-kits/altera/acceleration-card-arria-10-gx/getting-started.html



2. Go to the download file location and open terminal. Unzip the downloaded file:

a10_gx_pac_ias_1_2_pv_dev_installer.tar.gz

```
tar xvf a10_gx_pac_ias_1_2_pv_dev_installer.tar.gz
```

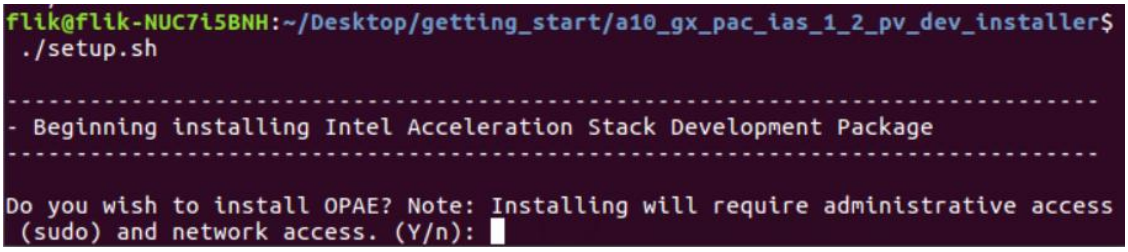
3. Change to the unzipped folder:

```
cd a10_gx_pac_ias_1_2_pv_dev_installer
```

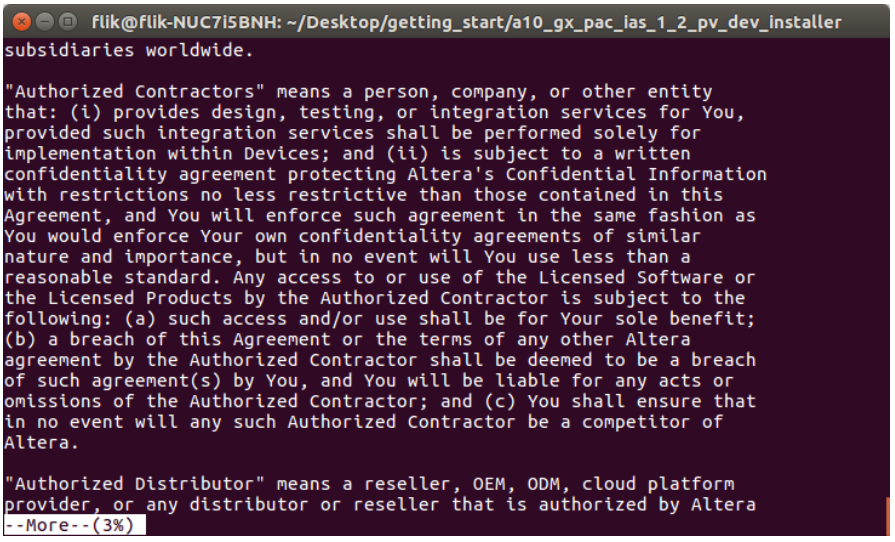
4. Run setup.sh

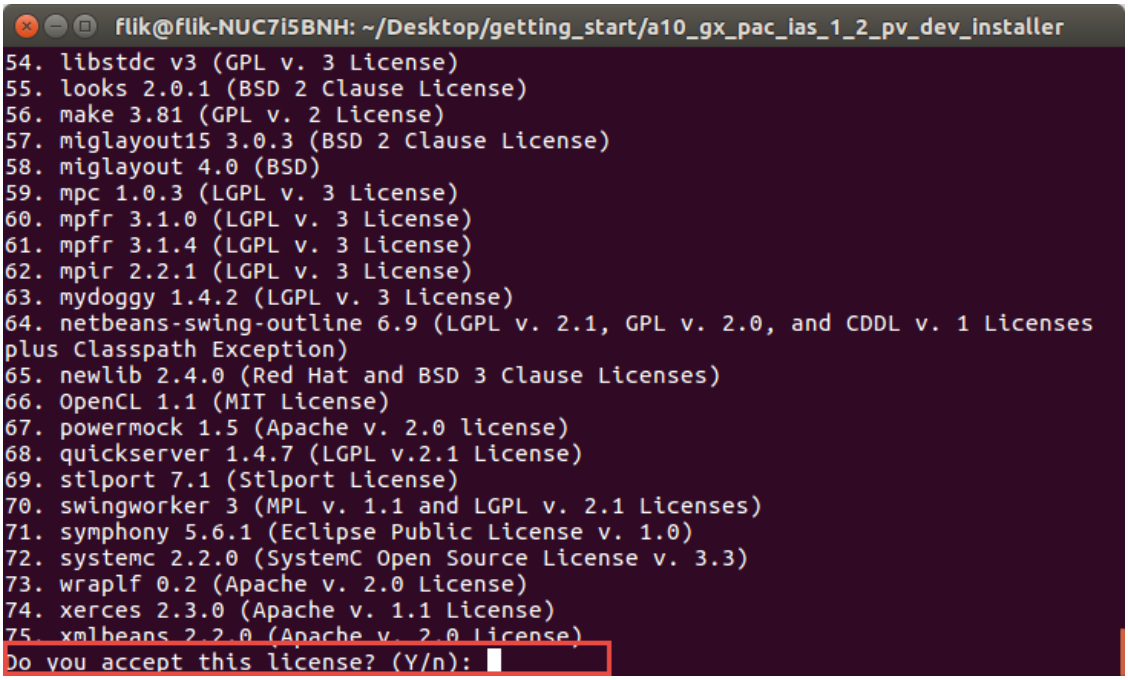
```
./setup.sh
```

5. You are prompted with the following question: **Do you wish to install OPAE?** Answer **Y.**

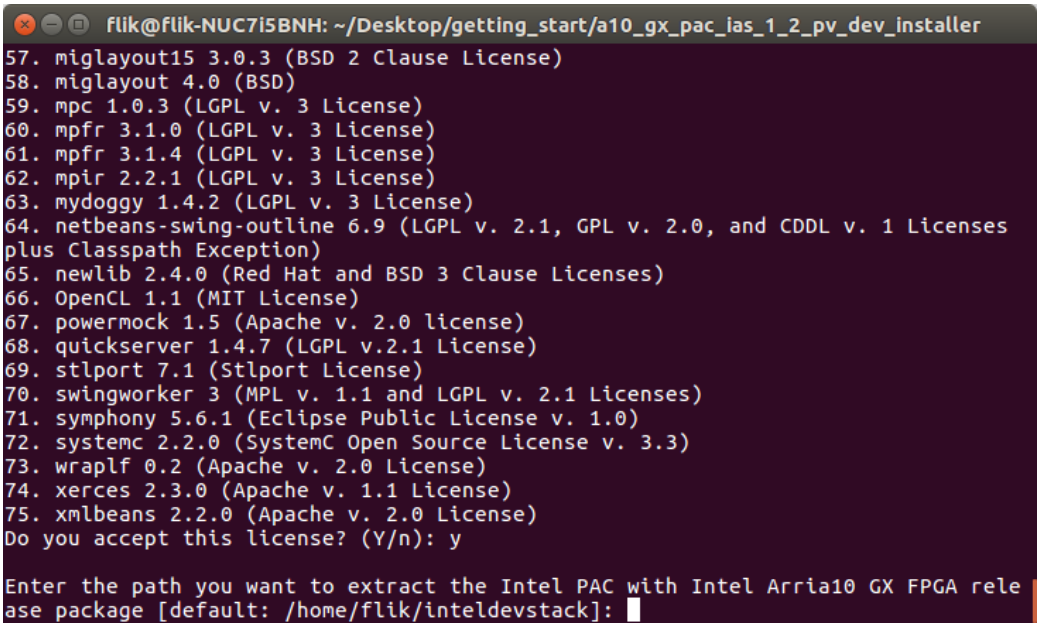


6. Accept the license.





7. When you receive an installation directory prompt, you can specify an install directory. Otherwise, the installer uses the default directory at /home/<username>/inteldevstack to install Intel® Quartus® Prime Pro Edition and OpenCL* SDK.



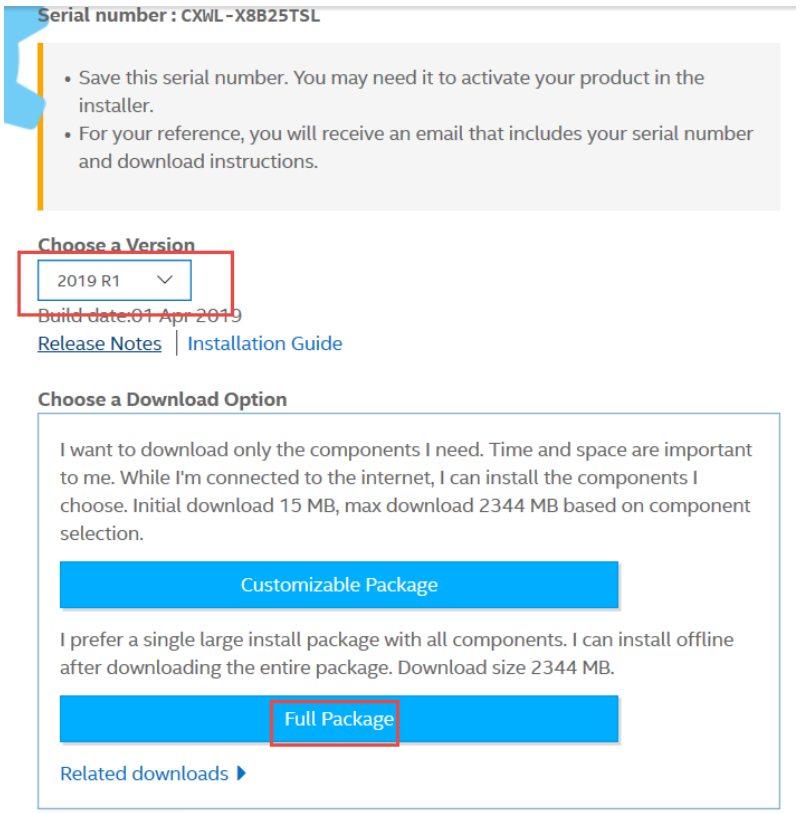
Chapter 4

Install Intel Distribution of OpenVINO toolkit

This chapter describes how to install the Intel Distribution of OpenVINO toolkit **for Linux* with FPGA support**. The following link is Intel's official installation guide, users can refer to the contents of this article for more information.
https://docs.openvinotoolkit.org/latest/_docs_install_guides_installing_openvino_linux_fpga.html

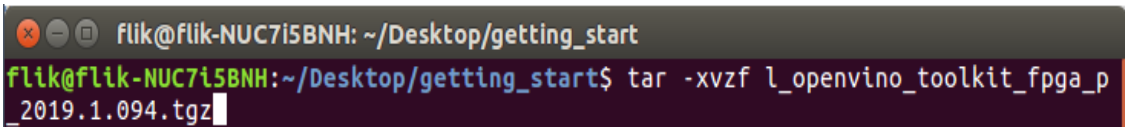
1. Go to the link below to download the Intel® Distribution of OpenVINO toolkit for Linux* with FPGA support:
   https://software.intel.com/en-us/openvino-toolkit/choose-download/free-download-linux-fpga

2. Please select **2019R1** version and select **Full Package** to download.



3. Open terminal and change directories to where you downloaded the Intel Distribution of OpenVINO toolkit for Linux* with FPGA support package file.
4. Unpack the .tgz file

   tar -xvzf l_openvino_toolkit_fpga_p_2019.1.094.tgz



The files are unpacked to the l_openvino_toolkit_fpga_p_<version> directory.

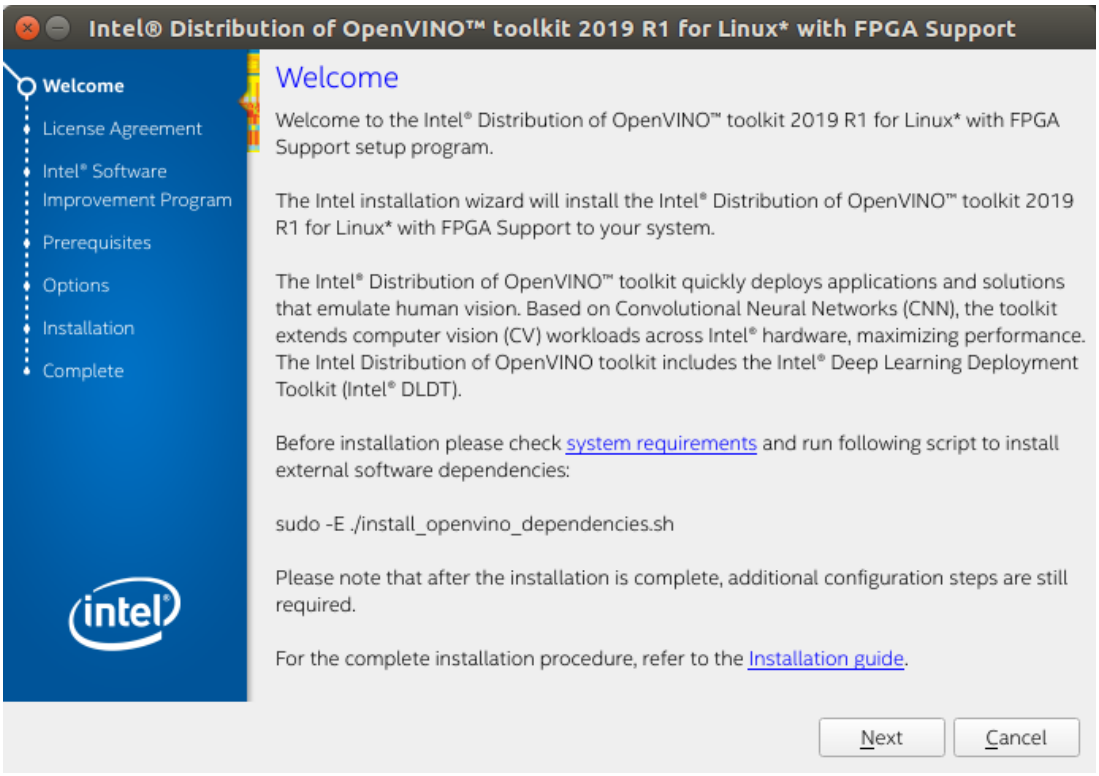
5. Go to the l_openvino_toolkit_fpga_p_<version> directory:

   cd l_openvino_toolkit_fpga_p_2019.1.094

6. Install the toolkit via GUI Installation Wizard.

   sudo ./install_GUI.sh

7. Follow the instructions to install the toolkit



8. When installed as root the default installation directory for the Intel Distribution of OpenVINO is /opt/intel/openvino_fpga_2019.1.094/. For simplicity, a symbolic link to the latest installation is also created: /opt/intel/openvino/.

9. Install external software dependencies, change to the install_dependencies directory:

```
cd /opt/intel/openvino/install_dependencies
```

10. Turn the script into an executable file.

```
sudo chmod +x install_openvino_dependencies.sh
```

11. Run a script to download and install the external software dependencies:

```
sudo -E ./install_openvino_dependencies.sh
```

Chapter 5

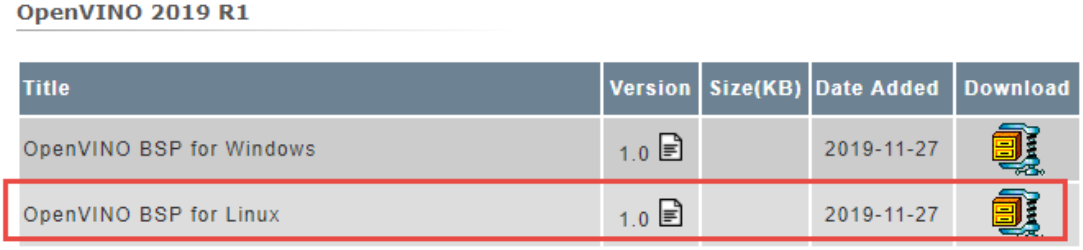
Install and verify the Terasic Demonstration of the FLIK

This chapter will show you how to quickly experience the Inference Engine demo on the FLIK.

5.1 Setup

This section describes how to quickly setup the environment for running an Inference Engine demonstration.

1. Users first need to download the OpenVINO BSP for Linux " **terasic_demo_flik_linux_2019R1.tar.gz** " in the link below. This file contains the board support package, environment variable settings and demonstration file.
   http://flik.terasic.com/cd



2. Open terminal and change directory to where you downloaded the board package file.
3. Extract the downloaded file "terasic_demo_flik_linux_2019R1.tar.gz ".

   ```
   tar -xvzf terasic_demo_flik_linux_2019R1.tar.gz
   ```

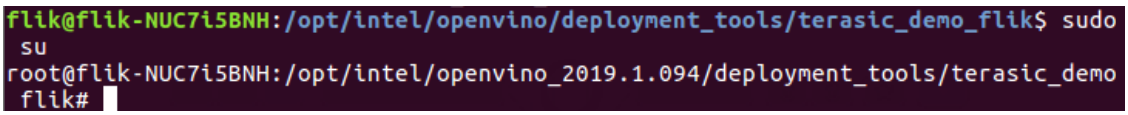
4. Copy the "terasic_demo_flik_linux_2019R1" folder to the path: /opt/intel/openvino/deployment_tools.

   ```
   sudo cp -r terasic_demo_flik_linux_2019R1 /opt/intel/openvino/deployment_tools
   ```

5. Go to the folder location you just copied

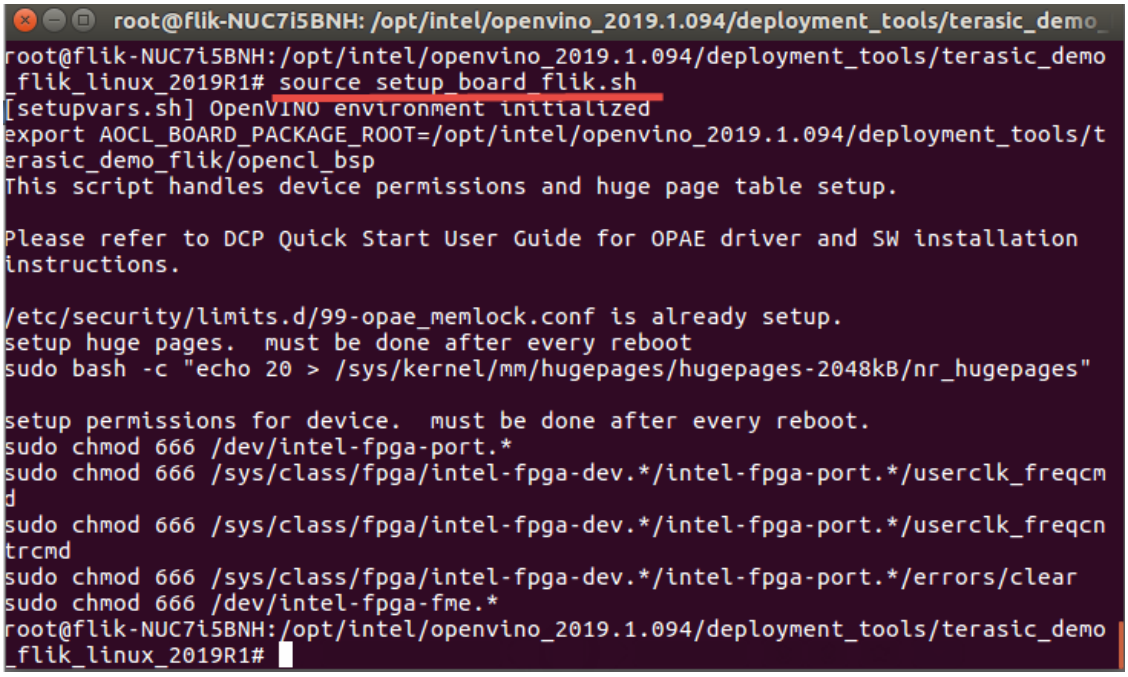
   ```
   cd /opt/intel/openvino/deployment_tools/terasic_demo_flik_linux_2019R1/
   ```

6. Switch to superuser
   ```
   sudo su
   ```



7. Execute the setting environment variable script.

   ```
   source setup_board_flik.sh
   ```



8. Make sure the FLIK is powered on and connected to the host PC via Thunderbolt 3 cable.
9. Execute the following command to verify whether the host PC has detected PCIe device in the FLIK.

   ```
   lspci | grep 09c4
   ```

   The result should be returned:

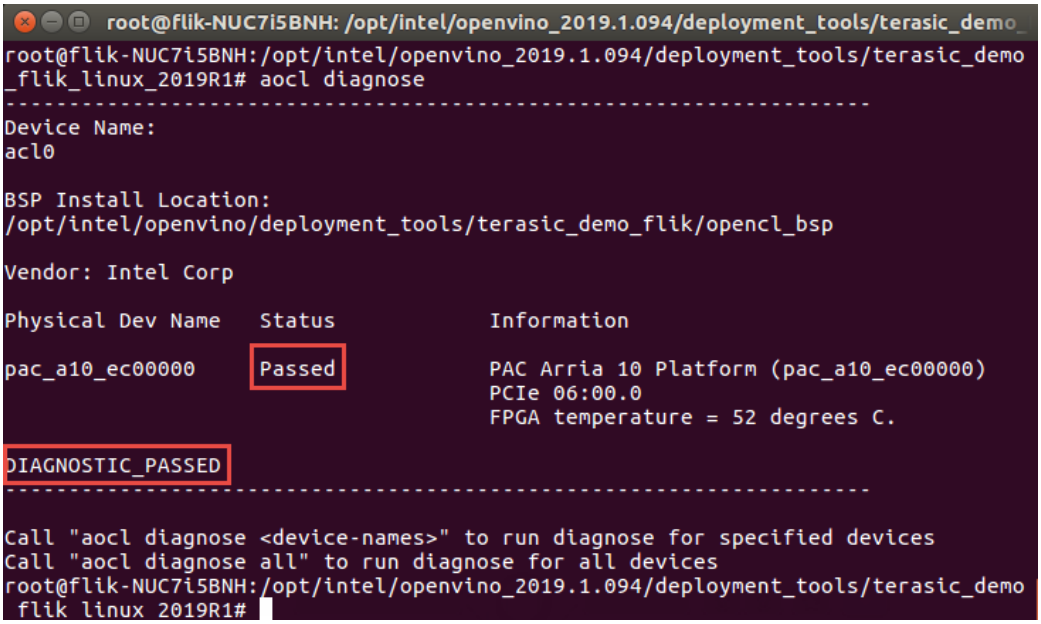
   06:00.0 Processing accelerators: Intel Corporation Device 09c4

10. Run the simple diagnostic utility:

    ```
    aocl diagnose
    ```

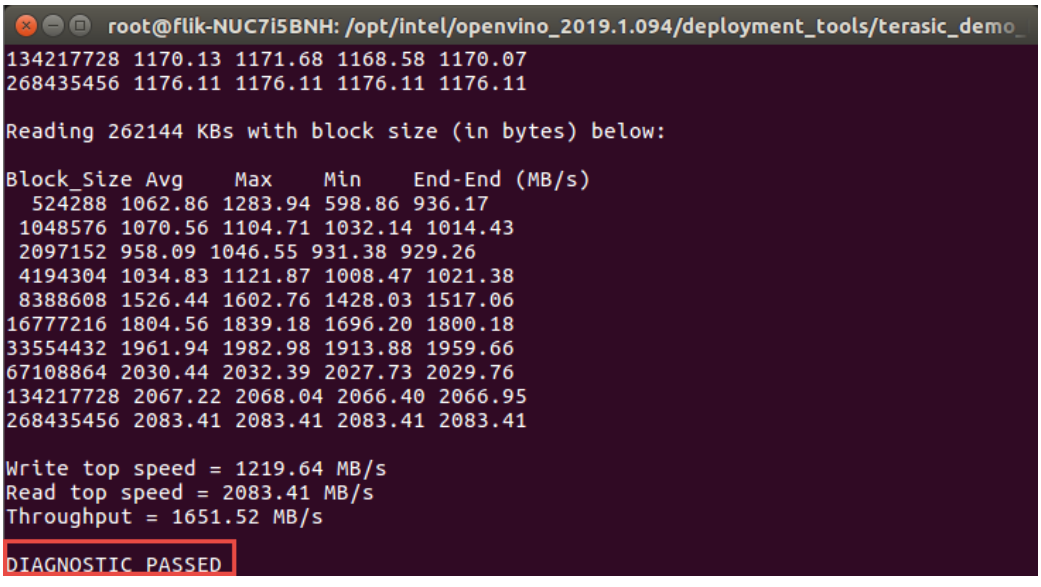
    The **aocl diagnose** command is useful to query the available PCIe device in a system. It will report device information and identifies issues.

    There is a diagnose report on the command shell as shown in below. If the command returns "**DIAGNOSTIC_PASSED**", it shows that the hardware configuration is successful.



11. The user can also do the test for the individual device connected to the host PC. The command is: aocl diagnose <device_name>. If the host PC is connected to only one device, <device_name> is usually named "acl0". So after entering the **aocl diagnose acl0** command, the memory transfer test will start and the test result will be returned.

    aocl diagnose acl0



Next, we will introduce how to use the squeezenet model with the FLIK to identify objects in the image.

In this demo, a SqueezeNet bitstream will be programmed into the FPGA on the FLIK and deploy the classification sample with a SqueezeNet model.

12. After installing the OpenVINO toolkit, there are several pre-compile bitstreams that can be directly used by Arria 10 FPGA on the FLIK. The location of these files is ***/opt/intel/openvino/bitstreams/a10_dcp_bitstreams*** directory. Please execute the following command to program the bitstream file into the FPGA of the FLIK.

    In this section, we will use 2019R1_RC_FP11_ResNet_SqueezeNet_VGG.aocx for programming bit stream.

    ```
    aocl program acl0 /opt/intel/openvino/bitstreams/a10_dcp_bitstreams/2019R1_RC_FP11_ResNet_SqueezeNet_VGG.aocx
    ```

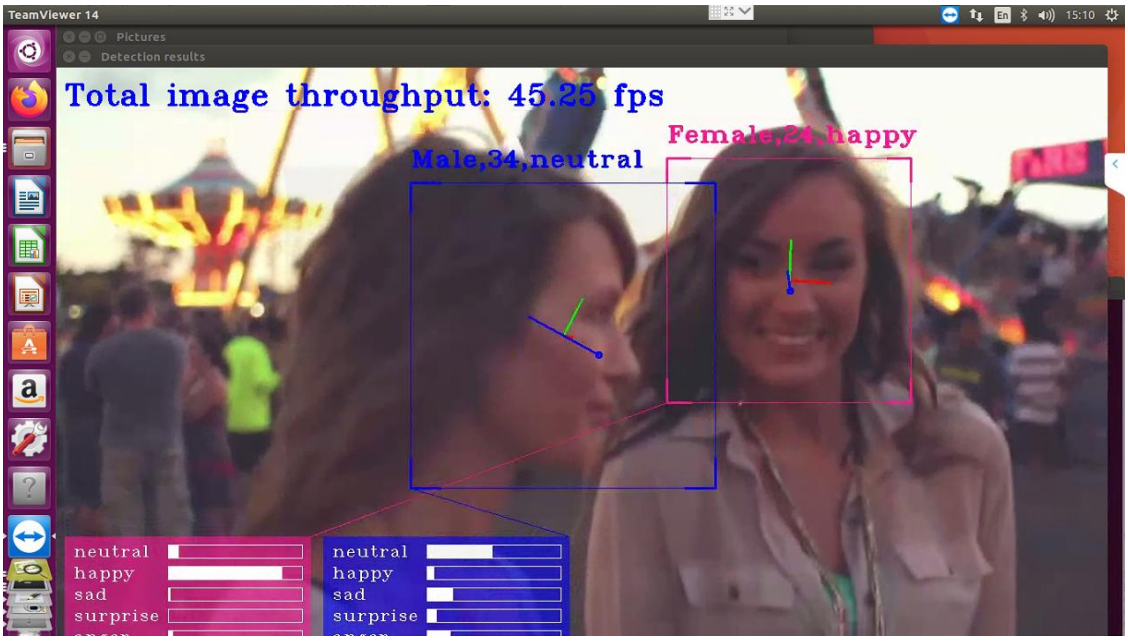
```
root@flik-NUC7i5BNH:/opt/intel/openvino_2019.1.094/deployment_tools/terasic_demo_flik_linux_2019R1# cd /opt/intel/openvino/bitstreams/a10_dcp_bitstreams
root@flik-NUC7i5BNH:/opt/intel/openvino/bitstreams/a10_dcp_bitstreams# aocl program acl0 /opt/intel/openvino/bitstreams/a10_dcp_bitstreams/2019R1_RC_FP11_ResNet_SqueezeNet_VGG.aocx
aocl program: Running program from /opt/intel/openvino/deployment_tools/terasic_demo_flik/opencl_bsp/linux64/libexec
Program succeed.
```

13. Go to the demo file location:

    ```
    cd /opt/intel/openvino/deployment_tools/terasic_demo_flik_linux_2019R1/demo
    ```

14. execute the demo batch file
    ```
    ./04_face_detection.sh video fpga
    ```

    You will see the face detection as shown in below.

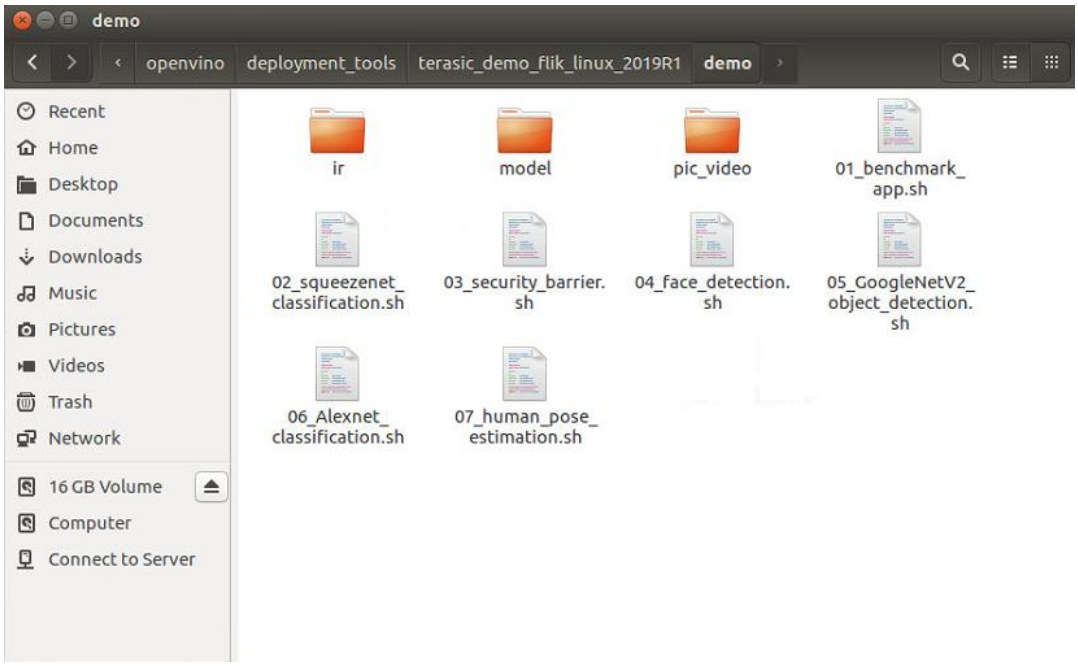


5.2 Other Demonstrations

This section describes how to run the reset of the demonstrations provided for the FLIK.

There are seven Inference Engine demos are provided in the "**terasic_demo_flik_linux_2019R1**" package. This section will briefly introduce the contents and results of the demo.

As shown in the figure below, there are some shell scripts in the terasic_demo_flik_linux_2019R1/demo folder. Below is the brief introduction of the demo folder.



1. How to use these Shell script files :
   Users can run any one of the shell script files with default parameters, and users can also use cpu , vpu or fpga to specify the target device to run the demo. There are also other parameters for using, users can run shell script file with '-h' for more details. And the default parameter is for cpu.

2. The images and video required by the demo are in the pic_video folder.

3. The Caffe model downloaded from internet is in the model folder. The folder includes:

   - Alexnet
   - Squeezenet1.1
   - GoogleNetV2

   Users can add Caffe model by referring to the writing rule of the script. Please pay attention to the path and name. Please refer to Converting a TensorFlow* Model to transfer the Tensorflow model

4. IR folder : While running the demo, the corresponding model IR file will be generated automatically if it's needed.

- The model generated under FP16 folder is used for FPGA
- The model generated under FP32 folder is used for CPU

Please note that each time you restart your terminal, please refer to steps 5~ 7 in section 5.1 to set the environment variable. Also, each time you restart your FLIK, please refer to steps 8~12 to set and verify the FPGA configuration in FLIK.

■ Benchmark Application

This topic demonstrates how to use the Benchmark Application to estimate deep learning inference performance on supported devices. The application outputs latency and throughput. The following are the steps to execute the demo.

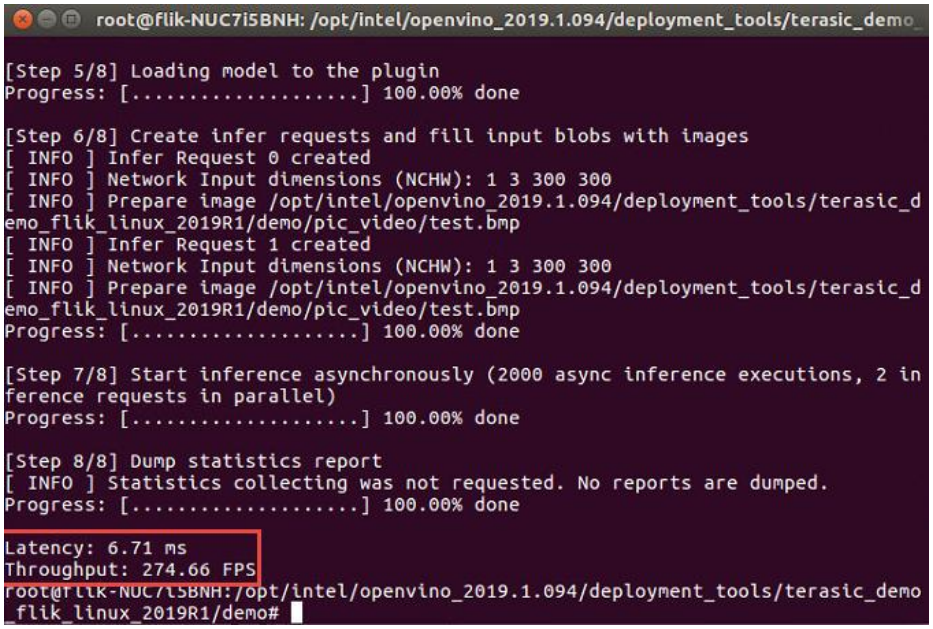
1. Go to the demo folder.

```
cd /opt/intel/openvino/deployment_tools/terasic_demo_flik_linux_2019R1/demo
```

2. Execute the demo batch file (run demo with FPGA).

```
./01_benchmark_app.sh fpga
```

3. The application outputs latency and throughput.



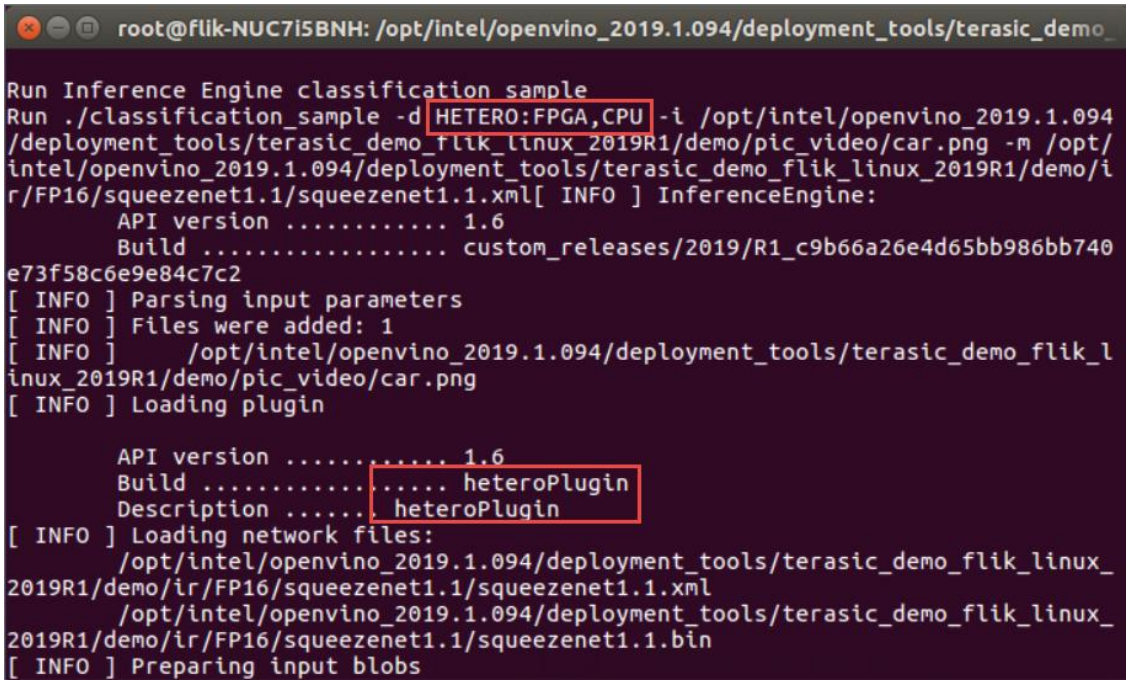
■ Squeezenet Classification

This demo can recognize the objects in the figure by using the squeezenet model

1. Execute the demo batch file (run demo with FPGA).

```
./02_squeezenet_classification.sh fpga
```

2. Users can see “HETERO:FPGA, CPU”, which prompts the DEMO is running on FPGA and CPU.



3. It prints out the top 10 results.

```
root@flik-NUC7i5BNH: /opt/intel/openvino_2019.1.094/deployment_tools/terasic_demo_

Top 10 results:

Image /opt/intel/openvino_2019.1.094/deployment_tools/terasic_demo_flik_linux_20
19R1/demo/pic_video/car.png

classid probability label
------- ----------- -----
817     0.8935490   sports car, sport car
511     0.0444872   convertible
479     0.0444872   car wheel
436     0.0060207   beach wagon, station wagon, wagon, estate car, beach waggon,
 station waggon, waggon
751     0.0060207   racer, race car, racing car
656     0.0022149   minivan
864     0.0008148   tow truck, tow car, wrecker
717     0.0008148   pickup, pickup truck
586     0.0008148   half track
581     0.0002998   grille, radiator grille


total inference time: 3.6003520
Average running time of one iteration: 3.6003520 ms
```

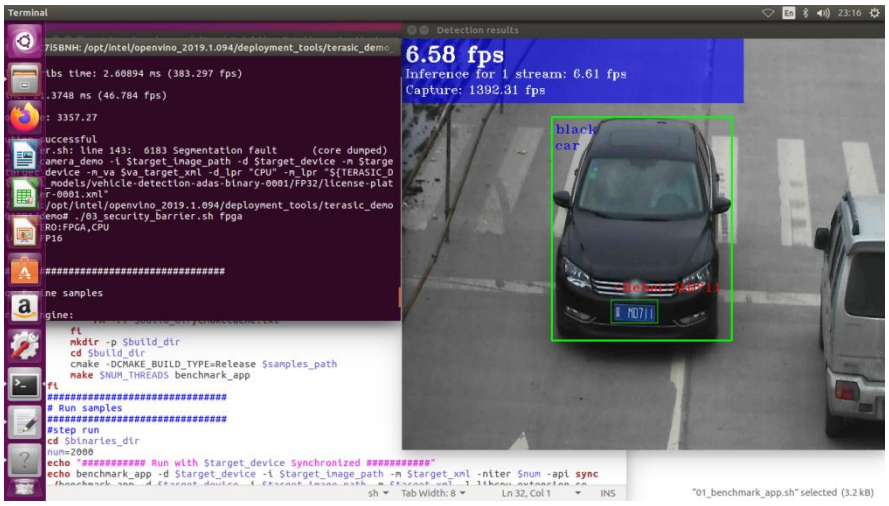
■ Security Barrier

This demo can recognize the car, car license number, and its location by using the three models.

1. Execute the demo batch file

```
./03_security_barrier.sh fpga
```

2. The result is shown in the figure below. Enter Ctrl+C to close the application.



■ Face Detection

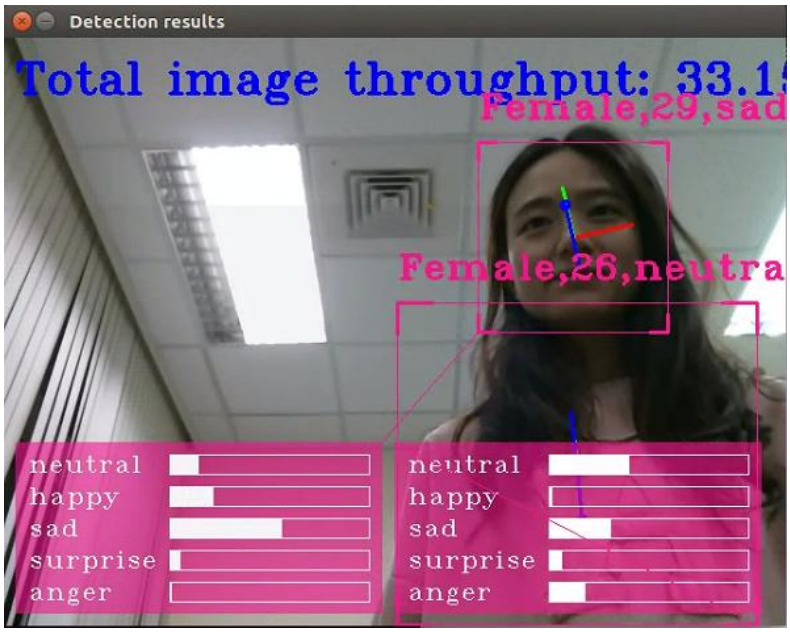
This demo uses four models and it can recognize human face position in the figure. It can also judge the human gender, age, expression, and head gesture according to the human face.

1. This video source of this demo can use USB Video Class(UVC) camera or video file in the demo folder. To use an UVC USB camera as the video source, plug a UVC USB camera to the host PC USB port.

2. Execute the demo batch file to run the demo with FPGA and USB camera.

```
./04_ face_detection.sh fpga
```

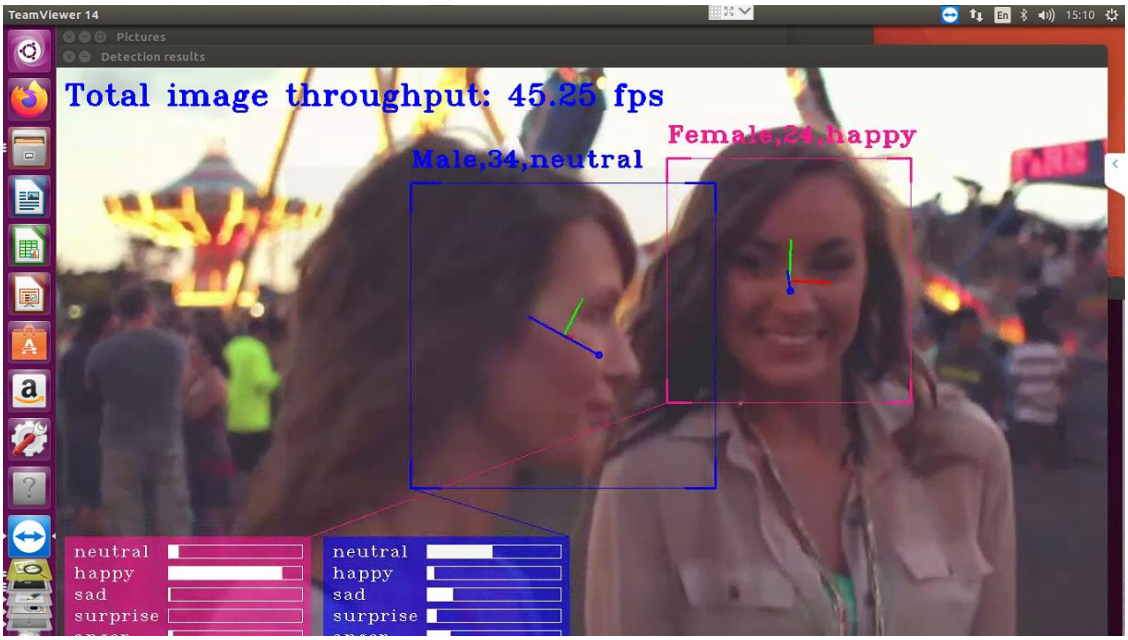
3. The result is shown in the figure below. Enter Ctrl+C to close the application.



4. To use video file as the video source of this demo. Please execute the command in below.

```
./04_ face_detection.sh video fpga
```

5. The result is shown in the figure below. Enter Ctrl+C to close the application.



■ GoogleNetV2 Object Detection

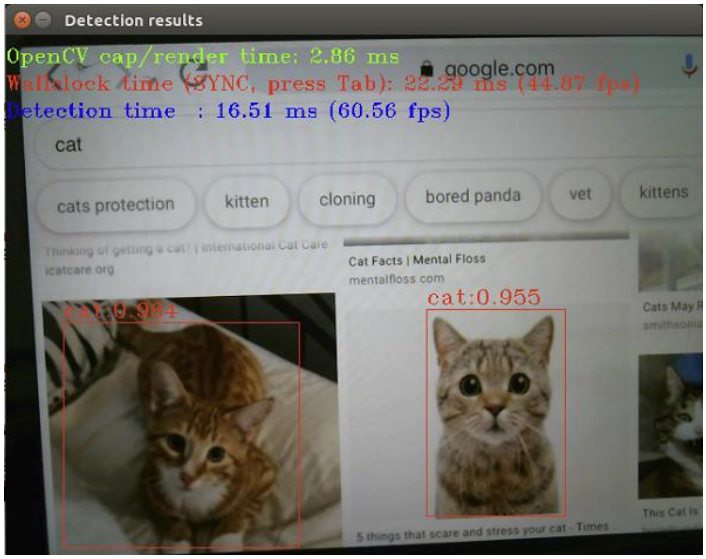
This demo can recognize the target object by using GoogleNetV2. The object tags are shown in the figure below:

```
1  aeroplane
2  bicycle
3  bird
4  boat
5  bottle
6  bus
7  car
8  cat
9  chair
10 cow
11 diningtable
12 dog
13 horse
14 motorbike
15 person
16 pottedplant
17 sheep
18 sofa
19 train
20 tvmonitor
```

1. Plug the UVC USB camera to the host PC USB port.
2. Execute the demo batch file to run the demo with FPGA.

```
./05_GoogleNetV2_object_detection.sh fpga
```

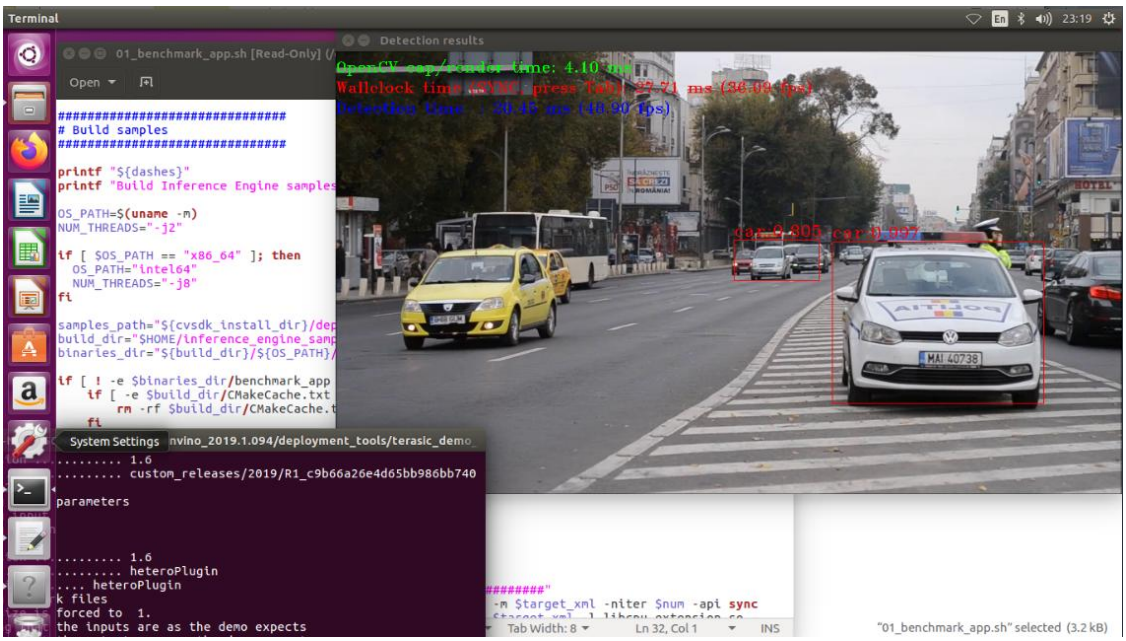
3. The results are shown in the figure below.



4. User can also use the video file as the input source of this demo.

```
./05_GoogleNetV2_object_detection.sh video fpga
```

5. The results are shown in the figure below.



■ Alexnet Classification

This demo can recognize the target objects (a car picture) by using Alexnet model and print out the top 10 information (the recognized result in top 10 probabilities).

1. Execute the demo batch file to run the demo with FPGA.

```
./06_Alexnet_classification.sh fpga
```

2. The results are shown in the figure below.

```
root@flik-NUC7i5BNH: /opt/intel/openvino_2019.1.094/deployment_tools/terasic_demo_

Top 10 results:

Image /opt/intel/openvino_2019.1.094/deployment_tools/terasic_demo_flik_linux_20
19R1/demo/pic_video/car.png

classid probability label
------- ----------- -----
817     0.8935490   sports car, sport car
511     0.0444872   convertible
479     0.0444872   car wheel
436     0.0060207   beach wagon, station wagon, wagon, estate car, beach waggon,
 station waggon, waggon
751     0.0060207   racer, race car, racing car
656     0.0022149   minivan
864     0.0008148   tow truck, tow car, wrecker
717     0.0008148   pickup, pickup truck
586     0.0008148   half track
581     0.0002998   grille, radiator grille


total inference time: 3.6003520
Average running time of one iteration: 3.6003520 ms
```

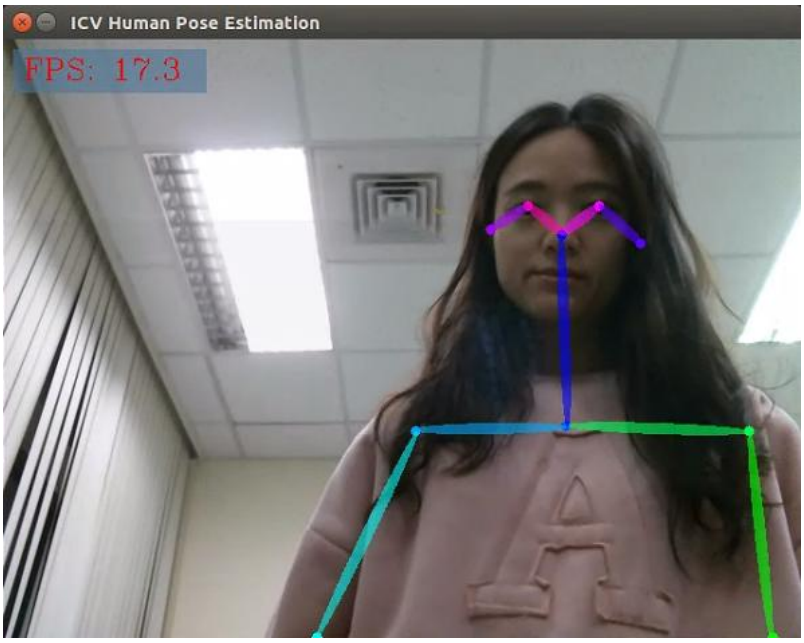
■ Human Pose Estimation

This demo can recognize human pose and display it.

1. Execute the demo batch file to run the demo with FPGA.

```
./07_human_pose_estimation.sh fpga
```

2. The result is shown in the figure below. Enter Ctrl+C to close the application.



Chapter 6

Factory Recovery

This chapter will show you how to recover the factory image file into the serial flash in the FLIK. When the FLIK is power on, the FPGA will read the factory configure file from the serial flash. The factory configuration allows the FPGA perform the acceleration task with the host PC. To program the serial flash file, user needs to connect the host PC to the FLIK via Thunderbolt 3. The configuration file is program to the serial flash from the host PC to the FLIK via the USB Blaster II interface.

Before referring to the steps below, please make sure that the user has already referenced the above chapters and has installed all the software and environment settings required.

- **Hardware Setup**

Power on the FLIK, and connect the host PC to the FLIK using the Thunderbolt 3 cable

- **Install USB Blaster II driver**

1. Please refer the link in below to install the Linux driver for USB Blaster II.
   https://rocketboards.org/foswiki/Documentation/UsingUSBBlasterUnderLinux

2. After setup USB Blaster II driver, user can enter the command "**lsusb**" to check if the system has detected the USB Blaster correctly. Normally you should see "**Altera**" appear on the device list.

   ```
   lsusb
   ```

```
flik@flik-NUC7i5BNH:~/Desktop/gs/thunderbolt-software-user-space-master$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 002: ID 09fb:6810 Altera
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 8087:0a2b Intel Corp.
Bus 001 Device 002: ID 046d:c52b Logitech, Inc. Unifying Receiver
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

3. Go to a path: ***/home/<username>/inteldevstack*** and execute the following command to run the initialization script to set the required environment variables.

   The environment variables setting will include the Quartus programmer. The Quartus programmer is a tool for downloading configuration files to serial flash and FPGA. This tool is available when you install Intel Acceleration Stack for **Development** version as described in Chapter3.

   Source init_env.sh

```
root@flik-NUC7i5BNH: /home/flik/inteldevstack
flik@flik-NUC7i5BNH:~$ cd /home/flik/inteldevstack/
flik@flik-NUC7i5BNH:~/inteldevstack$ sudo su
[sudo] password for flik:
root@flik-NUC7i5BNH:/home/flik/inteldevstack# source in
init_env.sh    intelFPGA_pro/
root@flik-NUC7i5BNH:/home/flik/inteldevstack# source init_env.sh
export QUARTUS_HOME=/home/flik/inteldevstack/intelFPGA_pro/quartus
export INTELFPGAOCLSDKROOT=/home/flik/inteldevstack/intelFPGA_pro/hld
Adding $QUARTUS_HOME/bin to PATH
export OPAE_PLATFORM_ROOT=/home/flik/inteldevstack/a10_gx_pac_ias_1_2_pv
export AOCL_BOARD_PACKAGE_ROOT=/home/flik/inteldevstack/a10_gx_pac_ias_1_2_pv/op
encl/opencl_bsp
source /home/flik/inteldevstack/a10_gx_pac_ias_1_2_pv/opencl/opencl_bsp/linux64/
libexec/setup_permissions.sh
Adding $OPAE_PLATFORM_ROOT/bin to PATH
sudo cp /home/flik/inteldevstack/a10_gx_pac_ias_1_2_pv/sw/fpgaflash /usr/bin/
sudo cp /home/flik/inteldevstack/a10_gx_pac_ias_1_2_pv/sw/afu_platform_info /usr
/bin/
find /home/flik/inteldevstack/a10_gx_pac_ias_1_2_pv/hw/samples/ -type d -name *S
10* -exec rm -r {} +
root@flik-NUC7i5BNH:/home/flik/inteldevstack#
```

4. You can use the Quartus programmer tool to check the JTAG chain status in the FLIK by entering the following command. The result will list the FPGA device status in the FLIK.

   jtagconfig

```
flik@flik-NUC7i5BNH:~/inteldevstack$ jtagconfig
1) FLIK [3-2]
  02E660DD   10AX115H1(.|E2|ES)/10AX115H2/..
  0318A0DD   10M04S(A|C)
```

5. Go to the bring up file path :

   cd /opt/intel/openvino/deployment_tools/terasic_demo_flik_linux_2019R1/bringup/

   Execute the script file to launch the function for programming the serial flash. Please note the subcommand ”flash”.

   ./bringup_board.sh flash

```
flik@flik-NUC7i5BNH: /opt/intel/openvino/deployment_tools/terasic_demo_flik_linux_20

flik@flik-NUC7i5BNH:/opt/intel/openvino/deployment_tools/terasic_demo_flik_linux
_2019R1/bringup$ ./bringup_board.sh flash
Programing startup image into flash
16M
Info: *******************************************************************
Info: Running Quartus Prime Programmer
    Info: Version 17.1.1 Build 273 12/19/2017 Patches 1.01dcp,1.02dcp,1.36,1.38
SJ Pro Edition
    Info: Copyright (C) 2017  Intel Corporation. All rights reserved.
    Info: Your use of Intel Corporation's design tools, logic functions
    Info: and other software and tools, and its AMPP partner logic
    Info: functions, and any output files from any of the foregoing
    Info: (including device programming or simulation files), and any
    Info: associated documentation or information are expressly subject
    Info: to the terms and conditions of the Intel Program License
    Info: Subscription Agreement, the Intel Quartus Prime License Agreement,
    Info: the Intel FPGA IP License Agreement, or other applicable license
    Info: agreement, including, without limitation, that your use is for
    Info: the sole purpose of programming logic devices manufactured by
    Info: Intel and sold by Intel or its authorized distributors.  Please
    Info: refer to the applicable agreement for further details.
    Info: Processing started: Tue Nov 12 15:59:06 2019
Info: Command: quartus_pgm -m jtag -c 1 -o p;Flik_sfl.sof
```

6. This procedure will first erase flash and then program it. It may take a few minutes.
7. After the program is complete, please power cycling the FLIK (Power off the FLIK first, then reopen the FLIK). Then reboot your system. Thus, the FLIK can restore to factory status for using.

Chapter 7

Additional Information

Here are the addresses where you can get help if you encounter problems:

■ Terasic Technologies

9F., No.176, Sec.2, Gongdao 5th Rd,
East Dist, HsinChu City, Taiwan, 30070
Email: support@terasic.com
Web: www.terasic.com
FLIK Web: flik.terasic.com

■ Revision History

Date	Version	Changes
2019.09	First publication	
2020.01	V1.1	Modify chapter 6