XMOS

AN02017: Getting Started with the xcore.ai Vision Development Kit

Publication Date: 2024/10/16
Document Number: XM-015125-AN v0.1.0

IN THIS DOCUMENT

1 Introduction

The `xcore.ai Vision Development Kit` is a board based on the `xcore.ai` chip with a small form factor (65x37.5mm) for developing Vision and Audio applications.

The `xcore.ai Vision Development Kit` includes the following features:

- Chip: `xcore.ai` (XU316-1024-TQ128-C24).
- Camera: One built-in 8MPx camera sensor (IMX219).
- Microphones: Two digital MEMS microphones (IM72D128V01).
- XTAG: Integrated XTAG4.
- Flash Memory: 64Mbit QSPI Flash (W25Q64JV)
- GPIO Capabilities: Buttons, LED, 3V3 IO among other features.
- Ports: 2x Micro USB ports: one for debug and power, and one dedicated to USB.

This section explains how to set up the environment to run the application, from the requirements to a quick start guide for running the provided source code.

This application note is targeted to run on the `xcore.ai Vision Development Kit`, although it can be easily adapted to run on other boards.

1.1 Required Hardware

- 1x `xcore.ai Vision Development Kit`
- 1x Micro USB cable (for Power supply and xTag)

1.2 Required Software

- XTC tools: 15.3.0 XTC tools.
- Python: 3.10 or later Python.
- CMake: 3.21 or later CMAKE.

1.3 Hardware Setup

- Plug the 1x Micro USB cable to both the host computer and the **DEBUG** Micro USB port.
- Face the `xcore.ai Vision Development Kit` horizontally, with the camera connector looking down.

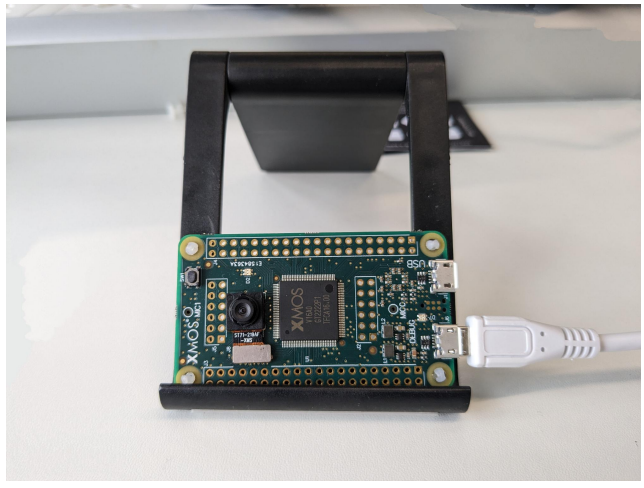

Fig. 1: Hardware Setup for the `xcore.ai Vision Development Kit`

1.4 Software Setup Example

Before running the example a Software setup is needed to ensure that all needed libraries are correctly imported. Software setup will depend on the actual application that is being developed. Make sure `XTC tools` are installed and activated in your development environment (XTC tools).

Once done, run then run the following commands:

```
# Get the Application Note file. Unzip them if needed.
# Create a Python venv
python -m venv .venv
# Activate venv (Windows)
call .venv/Scripts/activate.bat
# Activate venv (Mac OS, Linux)
source .venv/bin/activate
```

This will create a virtual environment to install the necessary packages. This process just needs to be run once, although the Python environment needs to be active while working on this project (`Activate` section on the previous block).

1.5 Hello World

Once the Hardware and Software are ready, the next step is building and running the application. This will be dependent as well on the Application that we are going to build. In this example, we will build a Hello World program just to ensure that the board runs correctly.

To install the necessary packages, generate the necessary build files and run the application, run the following command from the top-level directory:

```
# go to AN repo
cd an02017

# Python deps (if any)
pip install -r requirements.txt

# build
cmake -G "Unix Makefiles" -B build
xmake -C build

# run
xrun --io bin/proj_main.xe
```

This will print in the terminal:

```
Hello World!
```

The setup process is now complete. For further examples, refer to the following Application Notes:

- AN02005: *Xmos Logo Detector*
- AN02009: *Object Classification Demo*
- AN02010: *Face Identification on xcore.ai*
- AN02013: *FaceId And Keyword Spotting*

Those App Notes explore how to build complete applications and take advantage of the main resources of the board. It covers Vision, Audio, AI models and GPIO capabilities.

2 References

- XMOS XTC Tools User Guide: XTC tools.
- XMOS XTC Tools Installing Guide: XTC Installing Guide.

3 Support

For all support issues please visit XMOS Support.