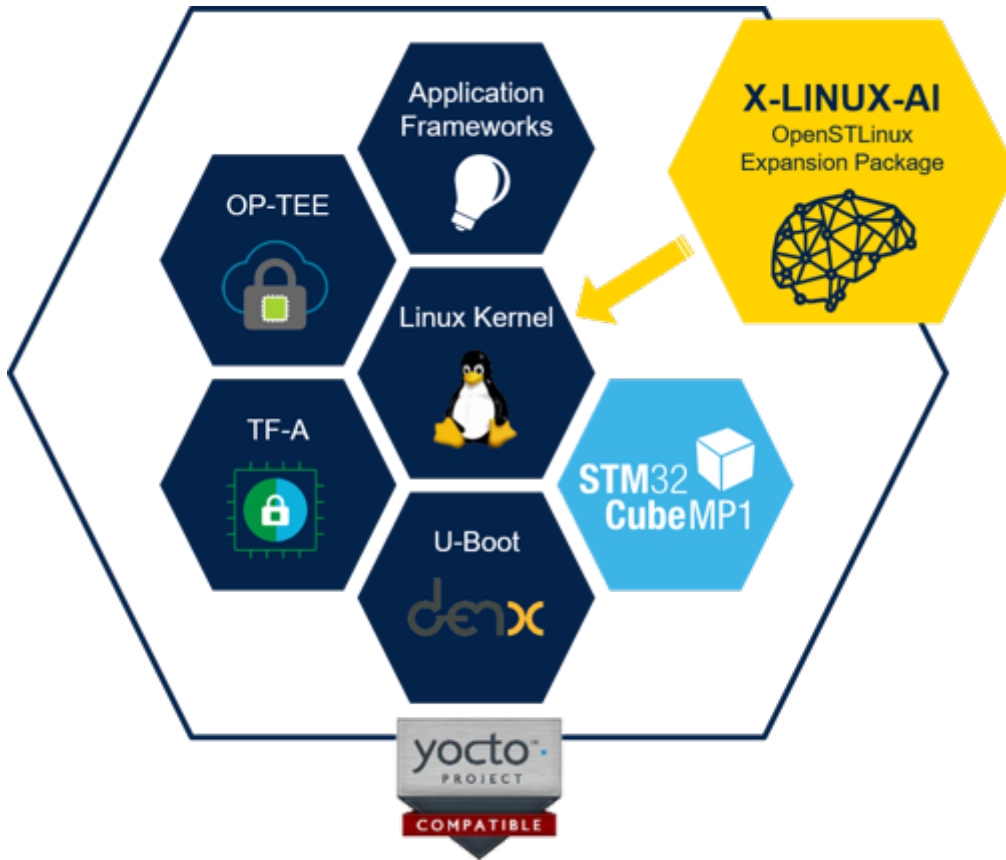




X-LINUX-AI OpenSTLinux Expansion Package

X-LINUX-AI OpenSTLinux Expansion Package



X-LINUX-AI is an STM32 MPU OpenSTLinux Expansion Package that targets artificial intelligence for STM32MP1 Series devices.

It contains Linux AI frameworks, as well as application examples to get started with some basic use cases such as computer vision (CV).

It is composed of an OpenEmbedded meta layer, named **meta-st-stm32mpu-ai**, to be added on top of the STM32MP1 Distribution Package.

It brings a complete and coherent easy-to-build / install environment to take advantage of AI on STM32MP1 Series devices.

Contents

1 Versions	4
1.1 X-LINUX-AI v2.1.0	4
1.1.1 Contents	4
1.1.2 Validated hardware	4
1.1.3 Software structure	5
1.2 X-LINUX-AI v2.0.0	6
1.2.1 Contents	6
1.2.2 Validated hardware	6
1.2.3 Software structure	7
2 Install from the OpenSTLinux AI package repository	8
2.1 Prerequisites	8
2.2 Configure the AI OpenSTLinux package repository	9



2.3 Install AI packages	9
2.3.1 Install all X-LINUX-AI packages	9
2.3.2 Install AI framework related packages	10
2.3.3 Install individual packages	10
3 Re-generate X-LINUX-AI OpenSTLinux distribution	17
3.1 Download the STM32MP1 Distribution Package	17
3.2 Install X-LINUX-AI environment for ST boards	17
3.3 Install X-LINUX-AI environment for Avenger96 board	17
3.4 Build the image	18
3.5 Flash the built image	18
4 How to use the X-LINUX-AI Expansion Package	19
4.1 Material needed	19
4.2 Boot the OpenSTlinux Starter Package	19
4.3 Install the X-LINUX-AI	19
4.4 Launch an AI application sample	20
4.5 Enjoy running your own NN models	20
5 References	21



1 Versions











1.1 X-LINUX-AI v2.1.0



This version is compatible with Yocto Project[®] build system Thud and Dunfell and has been validated against the OpenSTLinux ecosystem release v3.0.0 , ecosystem release v2.1.0, ecosystem release v2.0.0, ecosystem release v1.2.0 and validated on STM32MP157x-DKx, STM32MP157x-EV1 and STM32MP157 boards.

On Avenger96 ^[1]board the X-LINUX-AI expansion package has been validated up to OpenSTLinux ecosystem release v2.1.0

1.1.1 Contents

- Coral Edge TPU^[2] accelerator support
 -   libedgetpu 2.4.1 aligned with TensorFlow Lite 2.4.1 (built from source)
-   armNN^[3] 20.11
-   OpenCV^[4] 4.1.x
- Python^[5] 3.8.x (enabling Pillow module)
- Support STM32MP15x^[6] devices operating at up to 800MHz
- Application samples
 - C++ / Python image classification using TensorFlow Lite based on MobileNet v1 quantized model
 - C++ / Python object detection using TensorFlow Lite based on COCO SSD MobileNet v1 quantized model
 - C++ / Python image classification using Coral Edge TPU based on MobileNet v1 quantized model and compiled for the Coral Edge TPU
 - C++ / Python object detection using Coral Edge TPU based on COCO SSD MobileNet v1 quantized model and compiled for the Coral Edge TPU
 - C++ image classification using armNN TensorFlow Lite parser based on MobileNet v1 float model
 - C++ object detection using armNN TensorFlow Lite parser based on COCO SSD MobileNet v1 quantized model
-   TensorFlow Lite^[7] 2.4.1
 -   C++ face recognition using TensorFlow Lite models capable of recognizing the face of a known (enrolled) user (available on demand)



The face recognition binary is available on demand. Please contact the local STMicroelectronics support for more information about this application or send a request to edge.ai@st.com

1.1.2 Validated hardware

As any software expansion package, the X-LINUX-AI is supported on all STM32MP1 Series and it has been validated on the following boards:

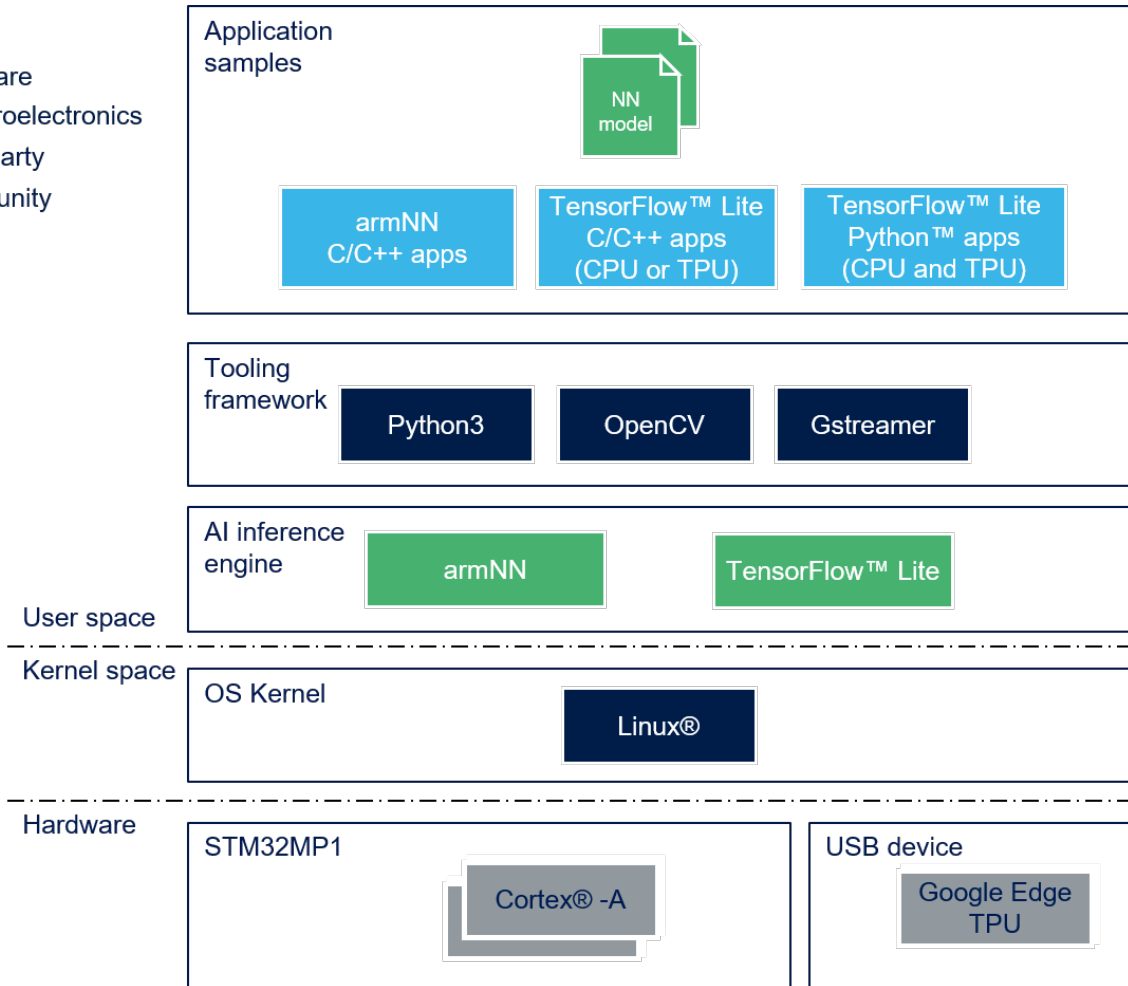
- STM32MP157C-DK2^[8]
- STM32MP157C-EV1^[9]
- STM32MP157A-EV1^[10]



- STM32MP157 Avenger96 board^[1]

1.1.3 Software structure

- Hardware
- STMicroelectronics
- Third party
- Community





1.2 X-LINUX-AI v2.0.0



This version has been validated against the OpenSTLinux ecosystem release v2.0.0 and validated on STM32MP157x-DKx and STM32MP157x-EV1 boards.

1.2.1 Contents

- TensorFlow Lite^[7] 2.2.0
- Coral Edge TPU^[2] accelerator support
- armNN^[3] 20.05
- OpenCV^[4] 4.1.x
- Python^[5] 3.8.x (enabling Pillow module)
- Support STM32MP15x^[6] devices operating at up to 800MHz
- Python and C++ application samples
 - Image classification using TensorFlow Lite based on MobileNet v1 quantized model
 - Object detection using TensorFlow Lite based on COCO SSD MobileNet v1 quantized model
 - Image classification using Coral Edge TPU based on MobileNet v1 quantized model and compiled for the Coral Edge TPU
 - Object detection using Coral Edge TPU based on COCO SSD MobileNet v1 quantized model and compiled for the Coral Edge TPU
 - Image classification using armNN TensorFlow Lite parser based on MobileNet v1 float model
 - Object detection using armNN TensorFlow Lite parser based on COCO SSD MobileNet v1 quantized model

1.2.2 Validated hardware

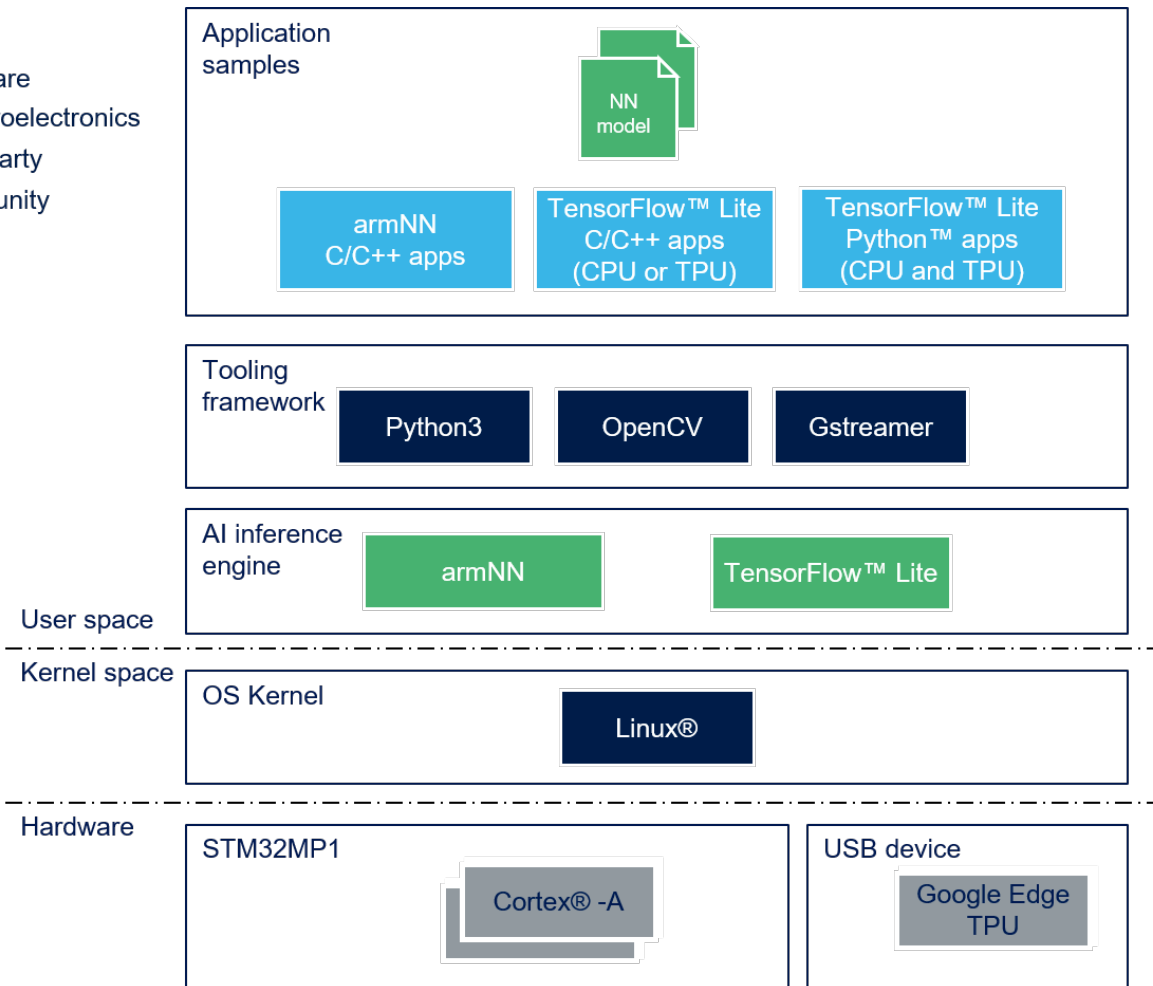
As any software expansion package, the X-LINUX-AI is supported on all STM32MP1 Series and it has been validated on the following boards:

- STM32MP157C-DK2^[8]
- STM32MP157C-EV1^[9]
- STM32MP157A-EV1^[10]



1.2.3 Software structure

- Hardware
- STMicroelectronics
- Third party
- Community





2 Install from the OpenSTLinux AI package repository



The STMicroelectronics packages repository service is provided for evaluation purposes only, its content may be updated at any time without notice and is therefore not approved for use in production.

All the generated X-LINUX-AI packages are available from the OpenSTLinux AI package repository service hosted at the non-browsable URL <http://extra.packages.openstlinux.st.com/AI>.

This repository contains AI packages that can be simply installed using **apt*** utilities, which the same as those used on a Debian system:

- the **main** group contains the selection of AI packages whose installation is automatically tested by STMicroelectronics
- the **updates** group is reserved for future uses such as package revision update.

You can install them individually or by package group.

2.1 Prerequisites

ST boards prerequisites:

- Flash the Starter Package on your SDCard
 - STM32MP157x-DKx Starter Package procedure
 - or
 - STM32MP157x-EV1 Starter Package procedure
- Your board has an internet connection either through the network cable or through a WiFi connection.



If your internet access depends on a proxy server, you should define the `http_proxy` environment variable with the following command before any `apt - *` commands:

```
Board $> export http_proxy='http://<proxy url>:<proxy port>/'
```

Avenger96 board prerequisites:

- The Avenger96 board starter image supporting OpenSTLinux v2.1.0 must be flashed on to your SD Card
 - OpenSTLinux-2.1 based on Yocto Dunfell LTS and Linux 5.4.56 - v6.5 Starter Image
- Your board has an internet connection either through the network cable or through a WiFi connection.



If your internet access depends on a proxy server, you should define the `http_proxy` environment variable with the following command before any `apt - *` commands:

```
Board $> export http_proxy='http://<proxy url>:<proxy port>/'
```




2.2 Configure the AI OpenSTLinux package repository

Once the board is booted, execute the following command in the console in order to configure the AI OpenSTLinux package repository:

For ecosystem release v3.0.0  :

```
Board $> wget http://extra.packages.openstlinux.st.com/AI/3.0/pool/config/a/apt-
openstlinux-ai/apt-openstlinux-ai_1.0_armhf.deb
Board $> dpkg -i apt-openstlinux-ai_1.0_armhf.deb
```

For ecosystem release v2.1.0 :

```
Board $> wget http://extra.packages.openstlinux.st.com/AI/2.1/pool/config/a/apt-
openstlinux-ai/apt-openstlinux-ai_1.0_armhf.deb
Board $> dpkg -i apt-openstlinux-ai_1.0_armhf.deb
```

For ecosystem release v2.0.0 :

```
Board $> wget http://extra.packages.openstlinux.st.com/AI/2.0/pool/config/a/apt-
openstlinux-ai/apt-openstlinux-ai_1.0_armhf.deb
Board $> dpkg -i apt-openstlinux-ai_1.0_armhf.deb
```

For ecosystem release v1.2.0 :

```
Board $> wget http://extra.packages.openstlinux.st.com/AI/1.2/pool/config/a/apt-
openstlinux-ai/apt-openstlinux-ai_1.0_armhf.deb
Board $> dpkg -i apt-openstlinux-ai_1.0_armhf.deb
```

Then synchronize the AI OpenSTLinux package repository.

```
Board $> apt-get update
```

2.3 Install AI packages



The software package is provided AS IS, and by downloading it, you agree to be bound to the terms of the software license agreement (SLA). The detailed content licenses can be found [here](#).

2.3.1 Install all X-LINUX-AI packages

Command	Description
	Install all the X-LINUX-AI packages



Command	Description
<code>apt-get install packagegroup-x-linux-ai</code>	(TensorFlow Lite, Edge TPU, armNN, application samples and tools)

2.3.2 Install AI framework related packages

Command	Description
<code>apt-get install packagegroup-x-linux-ai-tflite</code>	Install X-LINUX-AI packages related to TensorFlow Lite framework (including application samples)
<code>apt-get install packagegroup-x-linux-ai-tflite-edgetpu</code>	Install X-LINUX-AI packages related to the Edge TPU framework (including application samples)
<code>apt-get install packagegroup-x-linux-ai-armnn-tflite</code>	Install X-LINUX-AI packages related to the armNN framework (including application samples)

2.3.3 Install individual packages

X-LINUX-AI v2.1.0 packages

Command	Description
<code>apt-get install arm-compute-library</code>	Install Arm Compute Library (ACL)
	Install Arm Compute Library



Command	Description
<code>apt-get install arm-compute-library-tools</code>	utilities (graph examples and benchmarks)
<code>apt-get install armnn</code>	Install arm Neural Network SDK (armNN)
<code>apt-get install armnn-tensorflow-lite</code>	Install armNN TensorFlow Lite parser
<code>apt-get install armnn-tensorflow-lite-examples</code>	Install armNN TensorFlow Lite examples
<code>apt-get install armnn-tfl-cv-apps-image-classification-c++</code>	Install C++ image classification example using armNN TensorFlow Lite parser
<code>apt-get install armnn-tfl-cv-apps-object-detection-c++</code>	Install C++ object detection example using armNN TensorFlow Lite parser
<code>apt-get install armnn-tools</code>	Install armNN utilities such as unitary tests
<code>apt-get install python3-tensorflow-lite</code>	Install Python TensorFlow Lite inference engine
<code>apt-get install python3-tensorflow-lite-edgetpu</code>	Install Python TensorFlow Lite inference engine for Edge TPU
<code>apt-get install tensorflow-lite-edgetpu</code>	Install Edge TPU libraries and the USB rules



<pre>apt-get install tensorflow-lite-tools</pre>	<p>Install Tensorflow Lite utilities</p>
<pre>apt-get install tflite-cv-apps-edgetpu-image-classification-c++</pre>	<p>Install C++ image classification example using Coral Edge TPU TensorFlow Lite API</p>
<pre>apt-get install tflite-cv-apps-edgetpu-image-classification-python</pre>	<p>Install Python image classification example using Coral Edge TPU TensorFlow Lite API</p>
<pre>apt-get install tflite-cv-apps-edgetpu-object-detection-c++</pre>	<p>Install C++ object detection example using Coral Edge TPU TensorFlow Lite API</p>
<pre>apt-get install tflite-cv-apps-edgetpu-object-detection-python</pre>	<p>Install Python object detection example using Coral Edge TPU TensorFlow Lite API</p>
<pre>apt-get install tflite-cv-apps-image-classification-c++</pre>	<p>Install C++ image classification using TensorFlow Lite</p>
<pre>apt-get install tflite-cv-apps-image-classification-python</pre>	<p>Install Python image classification example using TensorFlow Lite</p>
<pre>apt-get install tflite-cv-apps-object-detection-c++</pre>	<p>Install C++ object detection example using TensorFlow</p>



Command	Description
	Lite
<code>apt-get install tflite-cv-apps-object-detection-python</code>	Install Python object detection example using TensorFlow Lite
<code>apt-get install tflite-edgetpu-benchmark</code>	Install benchmark application for Edge TPU models
<code>apt-get install tflite-models-coco-ssd-mobilenetv1</code>	Install TensorFlow Lite COCO SSD Mobilenetv1 model
<code>apt-get install tflite-models-coco-ssd-mobilenetv1-edgetpu</code>	Install TensorFlow Lite COCO SSD Mobilenetv1 model for Edge TPU
<code>apt-get install tflite-models-mobilenetv1</code>	Install TensorFlow Lite Mobilenetv1 model
<code>apt-get install tflite-models-mobilenetv1-edgetpu</code>	Install TensorFlow Lite Mobilenetv1 model for Edge TPU

X-LINUX-AI v2.0.0 packages

Command	Description
<code>apt-get install arm-compute-library</code>	Install Arm Compute Library (ACL)
<code>apt-get install arm-compute-library-tools</code>	Install Arm Compute Library utilities (graph examples and benchmarks)
	Install arm Neural Network SDK



Command	Description
<code>apt-get install armnn</code>	(armNN)
<code>apt-get install armnn-tensorflow-lite</code>	Install armNN TensorFlow Lite parser
<code>apt-get install armnn-tensorflow-lite-examples</code>	Install armNN TensorFlow Lite examples
<code>apt-get install armnn-tfl-benchmark</code>	Install armNN benchmark application for TensorFlow Lite models
<code>apt-get install armnn-tfl-cv-apps-image-classification-c++</code>	Install C++ image classification example using armNN TensorFlow Lite parser
<code>apt-get install armnn-tfl-cv-apps-object-detection-c++</code>	Install C++ object detection example using armNN TensorFlow Lite parser
<code>apt-get install armnn-tools</code>	Install armNN utilities such as unitary tests
<code>apt-get install libedgetpu1</code>	Install Edge TPU libraries and the USB rules
<code>apt-get install python3-tensorflow-lite</code>	Install Python TensorFlow Lite inference engine
	Install Python TensorFlow Lite inference engine



Command	Description
<code>apt-get install python3-tensorflow-lite-edgetpu</code>	for Edge TPU
<code>apt-get install tensorflow-lite-tools</code>	Install Tensorflow Lite utilities
<code>apt-get install tflite-cv-apps-edgetpu-image-classification-c++</code>	Install C++ image classification example using Coral Edge TPU TensorFlow Lite A PI
<code>apt-get install tflite-cv-apps-edgetpu-image-classification-python</code>	Install Python image classification example using Coral Edge TPU TensorFlow Lite A PI
<code>apt-get install tflite-cv-apps-edgetpu-object-detection-c++</code>	Install C++ object detection example using Coral Edge TPU TensorFlow Lite API
<code>apt-get install tflite-cv-apps-edgetpu-object-detection-python</code>	Install Python object detection example using Coral Edge TPU TensorFlow Lite A PI
<code>apt-get install tflite-cv-apps-image-classification-c++</code>	Install C++ image classification using TensorFlow Lite
<code>apt-get install tflite-cv-apps-image-classification-python</code>	Install Python image classification example using TensorFlow Lite
	Install C++ object



Command	Description
<code>apt-get install tflite-cv-apps-object-detection-c++</code>	detection example using TensorFlow Lite
<code>apt-get install tflite-cv-apps-object-detection-python</code>	Install Python object detection example using TensorFlow Lite
<code>apt-get install tflite-edgetpu-benchmark</code>	Install benchmark application for Edge TPU models
<code>apt-get install tflite-models-coco-ssd-mobilenetv1</code>	Install TensorFlow Lite COCO SSD Mobilenetv1 model
<code>apt-get install tflite-models-coco-ssd-mobilenetv1-edgetpu</code>	Install TensorFlow Lite COCO SSD Mobilenetv1 model for Edge TPU
<code>apt-get install tflite-models-mobilenetv1</code>	Install TensorFlow Lite Mobilenetv1 model
<code>apt-get install tflite-models-mobilenetv1-edgetpu</code>	Install TensorFlow Lite Mobilenetv1 model for Edge TPU



If you need more information about how to use apt-* utilities check the [Package repository for OpenSTLinux distribution](#) article.



3 Re-generate X-LINUX-AI OpenSTLinux distribution

With the following procedure, you can re-generate the complete distribution enabling the X-LINUX-AI expansion package. This procedure is mandatory if you want to update frameworks by yourself, or if you want to modify the application samples. For further details, please expand the contents...

3.1 Download the STM32MP1 Distribution Package

For ecosystem release v2.1.0 :

Install the STM32MP1 Distribution Package v2.1.0, **but do not initialize the OpenEmbedded environment (do not source the envsetup.sh).**

For ecosystem release v2.0.0 :

Install the STM32MP1 Distribution Package v2.0.0, **but do not initialize the OpenEmbedded environment (do not source the envsetup.sh).**

For ecosystem release v1.2.0 :

Install the STM32MP1 Distribution Package v1.2.0, **but do not initialize the OpenEmbedded environment (do not source the envsetup.sh).**

3.2 Install X-LINUX-AI environment for ST boards

- Clone the meta-st-stm32mpu-ai git repositories



The software package is provided AS IS, and by downloading it, you agree to be bound to the terms of the software license agreement (SLA). The detailed content licenses can be found [here](#).

For X-LINUX-AI v2.1.0:

```
PC $> cd <Distribution Package installation directory>/layers/meta-st
PC $> git clone https://github.com/STMicroelectronics/meta-st-stm32mpu-ai.git -b v2.1.0
```

For X-LINUX-AI v2.0.0:

```
PC $> cd <Distribution Package installation directory>/layers/meta-st
PC $> git clone https://github.com/STMicroelectronics/meta-st-stm32mpu-ai.git -b v2.0.0
```

- Set up the build environment

```
PC $> cd ../../
PC $> DISTRO=openstlinux-weston MACHINE=stm32mp1 BSP_DEPENDENCY='layers/meta-st/meta-st-stm32mpu-ai' source layers/meta-st/scripts/envsetup.sh
```

3.3 Install X-LINUX-AI environment for Avenger96 board

- Clone the meta-av96 and meta-st-stm32mpu-ai git repositories



The software package is provided AS IS, and by downloading it, you agree to be bound to the terms of the software license agreement (SLA). The detailed content licenses can be found [here](#).

For X-LINUX-AI v2.1.0:

```
PC $> cd <Distribution Package installation directory>/layers
PC $> git clone https://github.com/dh-electronics/meta-av96.git -b av96_v65
PC $> cd <Distribution Package installation directory>/layers/meta-st
PC $> git clone https://github.com/STMicroelectronics/meta-st-stm32mpu-ai.git -b v2.1.0
```

For X-LINUX-AI v2.0.0:

```
PC $> git clone https://github.com/dh-electronics/meta-av96.git -b av96_v62
PC $> cd <Distribution Package installation directory>/layers/meta-st
PC $> git clone https://github.com/STMicroelectronics/meta-st-stm32mpu-ai.git -b v2.0.0
```

- Set up the build environment

```
PC $> cd ../../
PC $> META_LAYER_ROOT=layers DISTRO=openstlinux-weston MACHINE=stm32mp1-av96 BSP_DEPENDENC
Y='layers/meta-st/meta-st-stm32mp-addons layers/meta-st/meta-st-stm32mpu-ai' source layers
/meta-st/scripts/envsetup.sh
```

3.4 Build the image

```
PC $> bitbake st-image-ai
```



Note that building the image could take long time depending on the host computer performance.

3.5 Flash the built image

Follow this [link](#) to see how to flash the built image.



4 How to use the X-LINUX-AI Expansion Package

4.1 Material needed

To use the X-LINUX-AI OpenSTLinux Expansion Package, choose one of the following materials:

- STM32MP157C-DK2^[8] + an UVC USB WebCam
- STM32MP157C-EV1^[9] with the built in OV5640 parallel camera
- STM32MP157A-EV1^[10] with the built in OV5640 parallel camera
- STM32MP157 Avenger96 board^[1] + an UVC USB WebCam or the OV5640 CSI Camera mezzanine board^[11]

Optional:

- Coral USB Edge TPU^[2] accelerator

4.2 Boot the OpenSTlinux Starter Package

At the end of the boot sequence, the demo launcher application appears on the screen.



4.3 Install the X-LINUX-AI



The software package is provided AS IS, and by downloading it, you agree to be bound to the terms of the software license agreement (SLA). The detailed content licenses can be found here.

After having configured the AI OpenSTLinux package you can install the X-LINUX-AI components.

```
Board $> apt-get install packagegroup-x-linux-ai
```

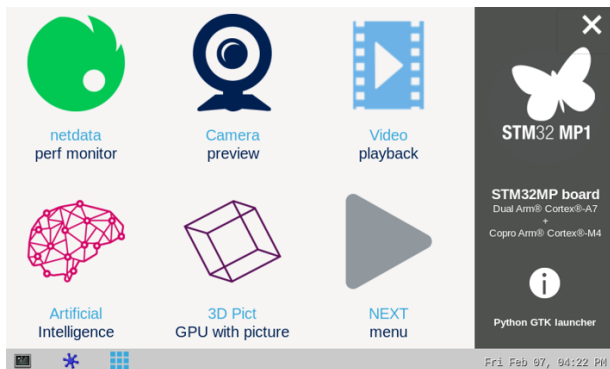
And restart the demo launcher

```
Board $> systemctl restart weston@root
```

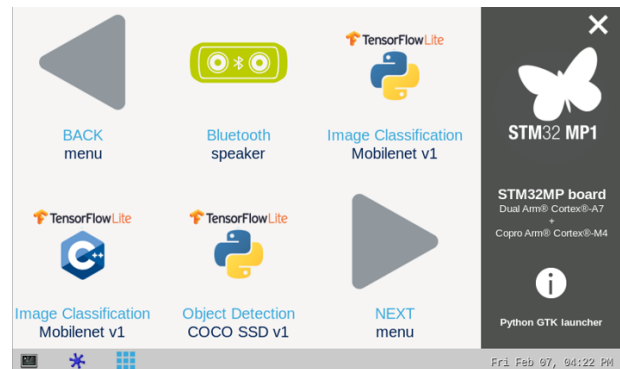


4.4 Launch an AI application sample

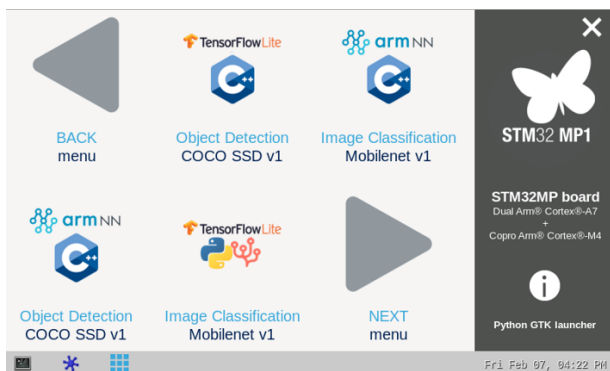
Once the demo launcher is restarted, notice that it is slightly different because new AI application samples have been installed. The demo launcher has the following appearance, and you can navigate into the different screens by using the **NEXT** or **BACK** buttons.



screen #1



screen #2



screen #3



screen #4

Screens 2, 3 and 4 contain AI application samples that are described within dedicated article available in the [X-LINUX-AI application samples zoo](#) page.

4.5 Enjoy running your own NN models

The above examples provide application samples to demonstrate how to execute models easily on the STM32MP1.

You are free to update the C/C++ application or Python scripts for your own purposes, using your own NN models.

Source code locations are provided in application sample pages.



5 References

- 1.01.11.2 Avenger96
- 2.02.12.2 Coral Edge TPU
- 3.03.1 armNN
- 4.04.1 OpenCV
- 5.05.1 Python
- 6.06.1 STM32MP1 series
- 7.07.1 TensorFlow Lite
- 8.08.18.2 STM32MP157C-DK2
- 9.09.19.2 STM32MP157C-EV1
- 10.010.110.2 STM32MP157A-EV1
- OV5640 CSI D3Camera board

Artificial Intelligence

Microprocessor Unit

Linux® is a registered trademark of Linus Torvalds.

Computer Vision

The Linux Foundation® and Yocto Project® are registered trademarks of the Linux Foundation. Linux® is a registered trademark of Linus Torvalds

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere. 

Automatic current limit (LCD power improvement solution)

Software development kit (A programming package that enables a programmer to develop applications for a specific platform.)

Application programming interface

Board support package

USB Video Class

Multi Speed Internal oscillator (STM32 clock source)

Neural Network