AVNET i.MX 8M Plus Linux-Yocto Edge AI Kit User Manual

AVNET®
Reach Further™

i.MX 8M Plus Linux-Yocto Edge AI Kit
User Manual

Contents

i.MX 8M Plus Linux-Yocto Edge AI Kitc

Copyright Statement:

Disclaimer:

- Avnet Manufacturing Services does not take warranty of any kind, either expressed or implied, as to the program source code, software and documents provided along with the products, and including, but not limited to, warranties of fitness for a particular purpose; The entire risk as to the quality or performance of the program is with the user of products.

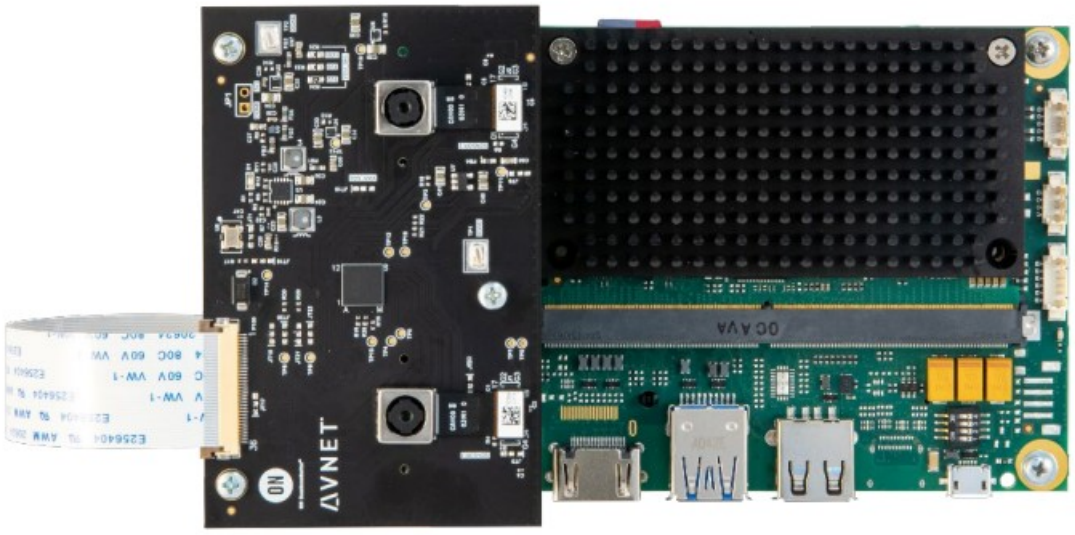
Revision History

Rev.	Description	Author	Date
v1.0	Initial version	Monica	2022/12/20
vii	Added Network boot instructions and support for the AR0830 image sensor	Monica	2022/3/17
V1.2	Revised to incorporate information about elQ that was left ou t in version 1.1	Monica	2022/4/18

Introduction

1.1 Target Board

Edge AI Kit is a development board developed by Avnet, based on the i.MX 8M+ processor from NXP.



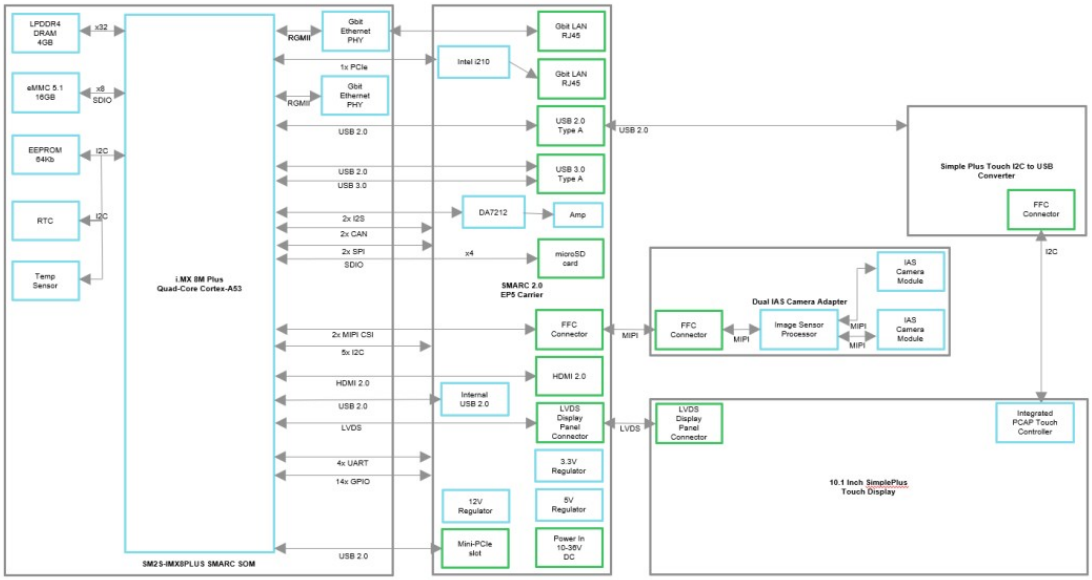
1.2 Introduction

This document provides a guide to prepare Edge AI Kit to boot up with the Verified Linux Package and introduces how to use the supported functions.

1.3 Feature List

- Yocto version: Handknit
- U-Boot version: 2021.04
- Kernel version: 5.10.35
- Evaluation image: Yocto Image
- Remote Ethernet boot / eMMC boot
- Device-tree overlay support
- Desktop (Weston 9.0)
- 2 x Gigabit Ethernet (IEEE 1588, 1x with TSN)
- 2 x USB 3.0 Host + 2 x USB 2.0 Host + 1 x USB 2.0 OTG

- 4 UART (TTL) including debug port
- External interfaces (I2C, UART, SPI, CAN, RS232 and GPIO)
- LVDS Display, HDMI 2.0
- I2S Audio
- MIPI-CSI Camera
- H.265/4 encode / decode



System Boot-Up

2.1 Preparation

2.1.1 Required Hardware
I addition to the kit hardware, you will need:

- SD card (8GB-64GB)

For debugging and interacting with the board:

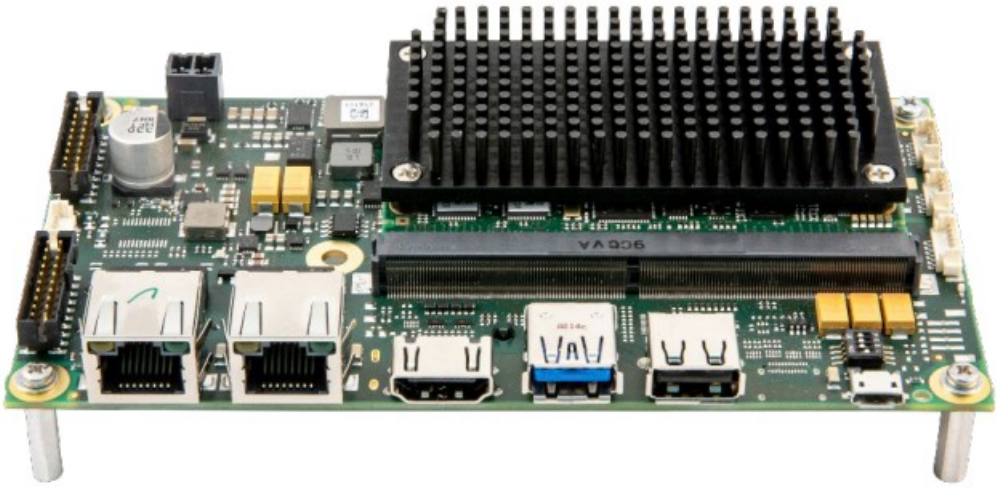
- A D-SUB 9-pin “null-modem” (Tx/Rx crossed) cable
- An RS-232 to USB cable (if your PC doesn’t have a native serial port) Alternatively, you can use a USB keyboard and mouse and an HDMI display.

2.1.2 Hardware Preparation

You’ll need a small phillips screwdriver to complete the following steps:

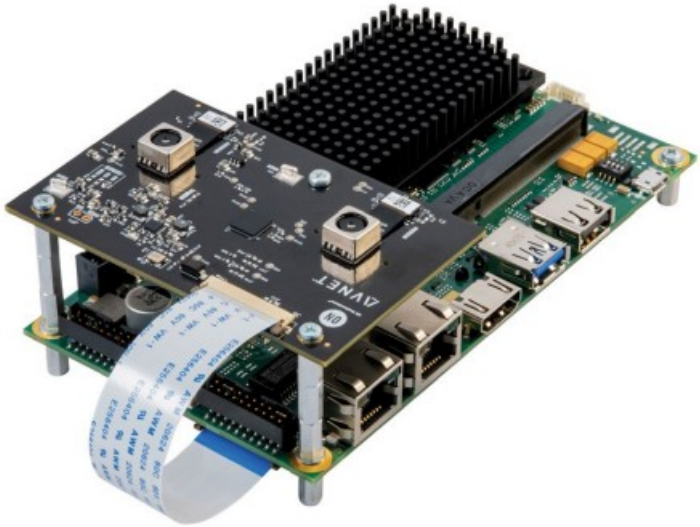
1. Plug the Avnet i.MX 8M Plus SMARC SOM into the EP5 carrier board at an angle, and then push down so it rests flat on the standoffs.

2. Use the provided screws to attach the heat sink to the Avnet i.MX 8M Plus SMARC



SOM.

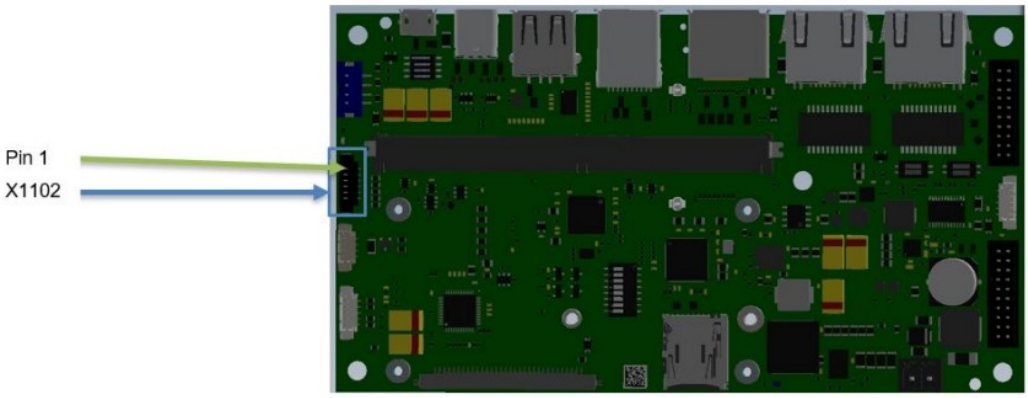
3. Connect the 12 volt, 3 amp power connector to the board, but do not plug it in yet.
4. Carefully, use the flat foil cable (FFC) to connect the SMARC EP5 carrier board to the Avnet Dual IAS Camera adapter. Use provided standoffs to mount the camera adapter.



2.1.3 Set up debug using RS-232 cable

If debugging using the RS-232 cable:

- Connect the RS-232 breakout cable to X1102 on the Edge AI Kit

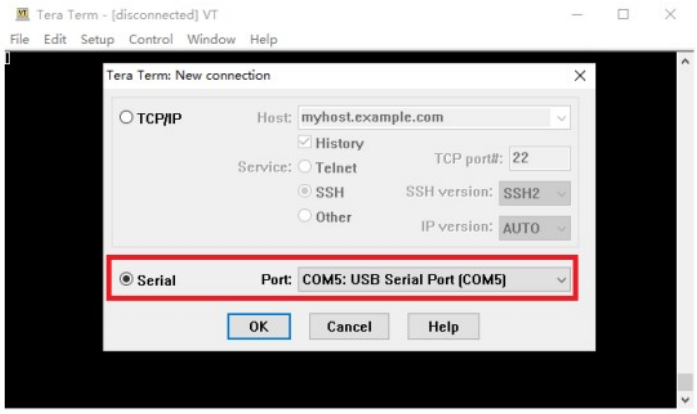


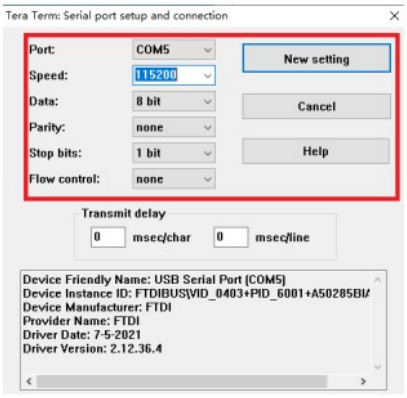
- Connect the D-SUB 9-pin “null-modem” cable to the RS-232 breakout cable on the EP5 side
- The other end of the D-SUB 9-pin cable goes to your development computer
- If your PC doesn’t have a native Serial Port, you can use a USB-to-Serial adapter.

2.1.4 Software Tools Preparation

Install Tera Term terminal software

- For Windows-based command-line debug output and command entry, the use of Tera Term terminal software is recommended
- Download and install teraterm-***.exe and configure the relevant COM port as shown below:



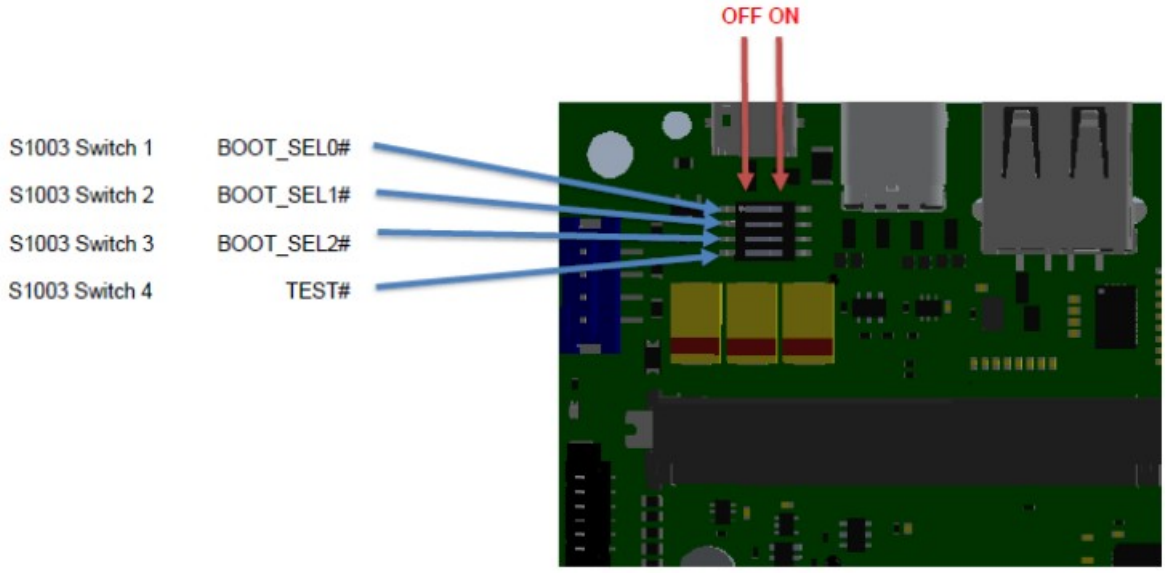


2.2 Booting the Edge AI Kit

Edge AI Kit supports Linux boot from SD card, Remote ethernet Boot, eMMC or USB Mass Storage.

2.2.1 Boot mode selection

The desired boot mode can be selected via a 4 pole Dipswitch located next to the microUSB port on the SMARC Carrier.



The following table describes the available boot options for the Edge AI Kit. See the i.MX 8M Plus Applications Processor Reference Manual from NXP for a complete description.

Boot Source	BOOT_SEL2#	BOOT_SEL1#	BOOT_SEL0#
Carrier SD Card	ON (GND)	ON (GND)	OFF (Float)
Remote Device (ethernet boot)	OFF (Float)	ON (GND)	OFF (Float)
module eMMC Flash	OFF (Float)	OFF (Float)	ON (GND)
USB Mass Storage	OFF (Float)	OFF (Float)	OFF (Float)

For DIP #4 (TEST#) — Booting SPL (secondary program loader)/U-Boot:
If TEST# = HIGH (DIP Switch = OFF), the i.MX8M Plus Boot ROM code uses the module eMMC flash as primary and the Carrier SD card as secondary (fallback) boot media. The fall back media is always selected, when booting from primary media is not possible (empty, corrupted, etc.)
If TEST# = LOW (DIP Switch = ON), the i.MX8M Plus Boot ROM code uses the Carrier SD card as boot media regardless of whether the module eMMC flash contains a properly programmed system image or not.

2.2.2 Boot from SD Card

2.2.2.1 Downloading operating system images

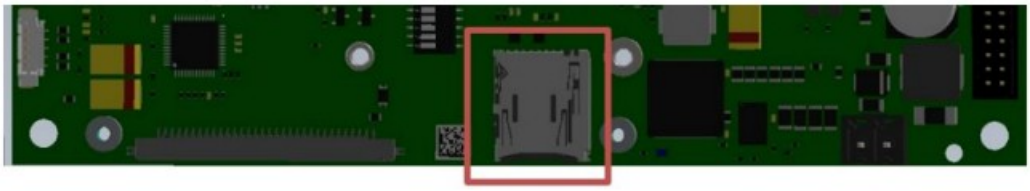
- Visit the downloads page: **http://avnet.me/imx8mplus-edgeai** to download the out of box Yocto image

2.2.2.2 Flash the SD Card

- You will need a 16+GB microSD card + adapter
- Download and install the flash tool – Etcher – on your host PC from: **https://www.balena.io/etcher**
- Flash the image to the SD card with Etcher

2.2.2.3 Boot the board

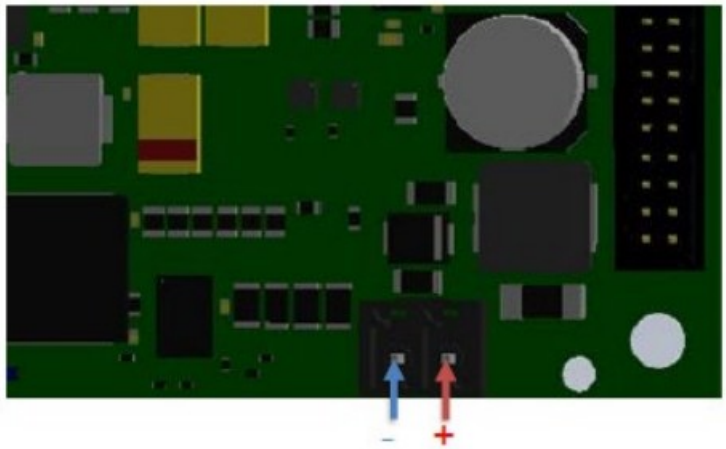
Once the image has been written to the micro SD Card, remove it and place it in the micro SD card slot on the bottom of the SMARC carrier.



- Make sure the DIP Switch is set to the following settings:

Switch 1	off
Switch 2	on
Switch 3	on
Switch 4	on

- On the host PC, open TeraTerm or another serial terminal with the settings defined in section 2.1.3.
- Connect the power supply to the SMARC Carrier via the 2pin Würth connector (X1401)
-



When the system boots-up, the TeraTerm will print the following:

```
Avnet Embedded Strudel Distro 0.1.0-ff2970817ef8e5172d4b6593e31cd82758a1477f sm2s-
imx8mp ttymxc1
sm2s-imx8mp login:
```

- Enter username as “root”, no password.
- The Linux system interface also supports directly attached keyboard and mouse
- Users can also use keyboard and mouse connected to Edge AI Kit to login to Linux.

Feature Configuration

3.1 Device Tree Overlays

The main concept of device trees and device tree overlays is to abstract the different hardware configurations while using the same kernel. There is a binary base blob for the SOM, which describes the properties, installed hardware components and features of a module, and activates them if possible and useful.
In addition, there are a few overlay blobs that modify the binary base blob depending on the desired usage

scenario of the overall system, e.g. to activate support for a specific baseboard, LVDS panel, HDMI interface, camera sensor, etc. Since overlays can be combined, it is very easy to realize many application scenarios without having to create a specific device tree for each of them, as was the case in the past.

3.1.1 Available Device Tree overlays

These are the overlays for the devices included in the Edge AI Kit. For overlays for additional compatible devices, see the MSC-LDK Manual from Avnet Embedded.

SOM overlay: msc-sm2s-imx8mp-24N0600I-module.dtb

Carrier overlay: overlay-baseboard-ep5.dtb

Display Overlays: overlay-hdmi.dtb

overlay-lvds0-ama-101a01.dtb

Camera Overlays:

File	Comment
overlay-caml-ap1302-ar1335-single.dtbo	support for the single onsemi AR1335 image sensor on CSI-0
overlay-caml-ap1302-ar1335-dual.dtbo	support for dual onsemi AR1335 image sensor on CSI-0
overlay-caml-ap1302-ar0144-single.dtbo	support for the single onsemi AR0144 image sensor on CSI-0
overlay-caml-ap1302-ar0144-dual.dtbo	support for the dual onsemi AR0144 image sensors on CSI-0

overlay-caml -ap1302-ar0830-single.dtbo	support for the single onsemi AR00830 image sensor on CSI-0
overlay-cam1-ap1302-ar0830-dual.dtb	support for the dual onsemi AR0830 image sensors on CSI-0

3.1.2 Selecting the device tree blobs in U-Boot

Two environment variables in the U-Boot-env are used for this purpose:

- fdt_module
- fdt_overlay

The variable fdtmodule specifies the base device tree of the SoM variant to be used, in this case fdt_module=msc-sm2s-imx8mp-24N0600I-module.dtb. The variable fdt_overlay is the desired overlay(s). To set the environment variables use the U-Boot commands.

To enter U-Boot, with TeraTerm open on your host PC, reboot the Edge AI Kit from the console:

```
root@sm2s-imx8mp:~# reboot
```

To interrupt boot process, hit esc, then hit any key.

```
Hit any key to stop autoboot: 0
u-boot=>
```

For example, to set the LVDS overlay, along with dual AR1335 cameras, use the following commands:

```
setenv fdt_overlay 'overlay-cam1-ap1302-ar1335-dual.dtbo overlay-lvds0-ama-101a01.dtb'
saveenv
```

Finally, use the printenv command to make sure the environment saved correctly:

```
u-boot=> printenv fdt_overlay
fdt_overlay=overlay-cam1-ap1302-ar1335-single.dtbo overlay-lvds0-ama-101a01.dtb
```

To reset to the default environment, use the command:

```
env default -a -f; saveenv
```

Continue boot with the command “boot.”

3.2 Display Output

Edge AI Kit supports both LVDS and HDMI displays.

3.2.1 HDMI Display

By default, both HDMI and LVDS are selected for the fdt_overlay, so HDMI should work out of the box. In U-Boot, use the printenv command to check:

```
u-boot=> printenv fdt_overlay
fdt_overlay=overlay-lvds0-ama-101a01.dtb overlay-hdmi.dtb overlay-cam1-ap1302-ar1335-dual.dtbo
```

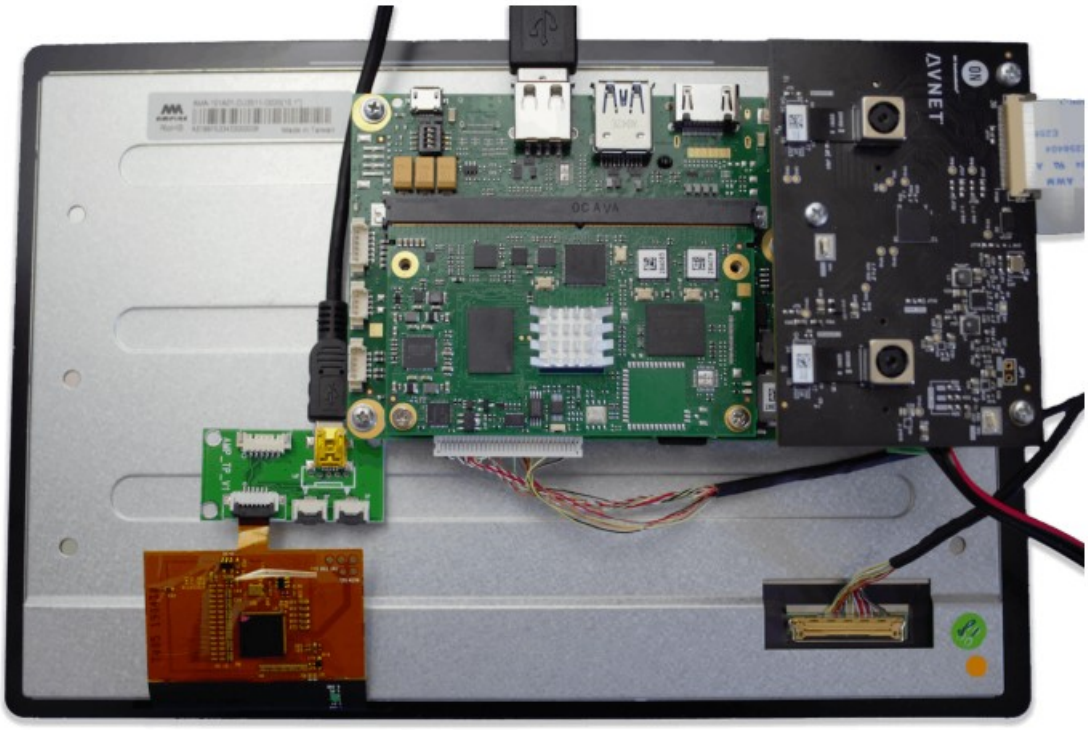
If you wish to use only HDMI and not LVDS, set the environment like this:

```
setenv fdt_overlay 'overlay-hdmi.dtb overlay-cam1-ap1302.dtb'
saveenv
```

3.2.2 LVDS Display

The 10.1” LVDS Touch Display is included with the Edge AI Kit.

To set up the 10.1” Touch Display, use the LVDS and backlight cable to connect it to the EP5 carrier. Connect one end of the USB cable to mini-USB cable to the Touch Adapter Board, and the other end to a USB port on the EP5 Carrier. Connect the Touch Adapter Board’s J5 pin to the orange FFC cable on the display.



3.3 AP1302 and Cameras

The AP1302 is NXP’s Image Sensor Processor (ISP). It processes the raw images from the SMARC Dual Camera Adapter.

The Dual Camera Adapter features two AR1335 IAS Camera modules, which are 1/3.2-inch CMOS active-pixel digital image sensors with a pixel array of 4208H x 3120V. The modules are manufactured by Rapyrus. The AR1335 digital image sensor features 1.1µm pixel technology that delivers superior low-light image quality through leading sensitivity, quantum efficiency and linear full well.

For more detailed information regarding the camera modules and image sensor processor, please see the AR1335 IAS Module and AP1302 Image Signal Processor pages from onsemi.

3.3.1 Default Camera Streaming

By default, the ap1302-stream.service runs the ap1302-stream.sh which displays the AP1302 video stream on the weston desktop. It is set to auto start and auto restart every 5 seconds on failure.

To stop streaming, run the command:

```
systemctl stop ap1302-stream
```

To start it again:

```
systemctl start ap1302-stream
```

3.3.2 SMARC AP1302 Configuration

By default, the Edge AI kit supports the AR1335 IAS modules on the SMARC Dual Camera Adapter. The image has default boot configuration for dual AR1335.

The ap1302-cfg.sh script can show and change the system configuration. To view possible configurations, run:

```
root@sm2s-imx8mp:~# ap1302-cfg.sh -h
Usage /bin/ap1302-cfg.sh [-h|help|status|<config>]
        -h|help : print this help
        status : show current overlay config
        available config
                ar1335-single : set ar1335 single config
                ar1335-dual     : set ar1335 dual config
        extended config - hardware dependent
                ar0144-single : set ar0144 single config
                ar0144-dual      : set ar0144 dual config
                ar0830-single : set ar0830 single config
                ar0830-dual     : set ar0830 dual config
```

3.3.3 Configure Dual vs Single Image Sensors

The Edge AI Kit currently supports both single and dual camera configurations.

Dual or single cameras can be configured using device tree overlays from U-Boot, like so:

```
setenv fdt_overlay 'overlay-cam1-ap1302-ar1335-single.dtbo overlay-lvds0-ama-
101a01.dtb'
saveenv
```

They can also be configured using the ap1302-cfg.sh script as shown below.

To use a single camera, use the following command:

```
root@sm2s-imx8mp:~# ap1302-cfg.sh ar1335-single
Cannot read environment, using default
U-Boot Env overlay configuration
fdt_overlay=overlay-lvds0-ama-101a01.dtb overlay-hdmi.dtb overlay-cam1-ap1302-ar1335-
single.dtbo
```

Reboot the board from the console with reboot or poweroff:

```
root@sm2s-imx8mp:~# poweroff
```

After the reboot:

```
root@sm2s-imx8mp:~# ap1302-cfg.sh status
U-Boot Env overlay configuration
fdt_overlay=overlay-lvds0-ama-101a01.dtb overlay-hdmi.dtb overlay-cam1-ap1302-ar1335-
single.dtbo
```

To switch back to dual cameras, use the command:

```
ap1302-cfg.sh ar1335-dual
```

3.3.4 Configure other onsemi image sensors

In addition to the included AR1335, the onsemi AR0144 image sensor is currently supported. These can be configured using U-Boot, E.G.:

```
setenv fdt_overlay 'overlay-cam1-ar0144-dual.dtbo overlay-lvds0-ama-101a01.dtb'
saveenv
```

They can also be configured using using the AP1302 config script. To configure a single AR0144, use the following command and then reboot:

```
ap1302-cfg.sh ar1335-single
```

To configure dual AR0144, use this command and then reboot:

```
ap1302-cfg.sh ar0144-dual
```

3.3.5 Troubleshooting Cameras

You can check that both cameras boot with the status command:

```
root@sm2s-imx8mp:~# ap1302-cfg.sh status
U-Boot Env overlay configuration
Cannot read environment, using default
fdt_overlay=overlay-lvds0-ama-101a01.dtb overlay-hdmi.dtb overlay-cam1-ap1302-ar1335-dual.dtbo
```

The AP1302 should be correctly enumerated:

```
root@sm2s-imx8mp:~# dmesg | grep ap1302
[    0.000000] Kernel command line: console=ttymxc1,115200 root=/dev/mmcblk1p2 rootwait rw dtb-module=msc-sm2s-imx8mp-24N0600I-module.dtb dtb-overlay=overlay-lvds0-ama-101a01.dtb overlay-hdmi.dtb overlay-cam1-ap1302-ar1335-dual.dtbo logl7
[    5.572050] ap1302: loading out-of-tree module taints kernel.
[    5.781162] ap1302 4-003c: Firmware header version : 1
[    5.803647] ap1302 4-003c: Firmware description : Release 404 @ 8-13-2021 0:50:37 - AP1302 Dual AR1335 on shared I2C bus (IAS1MOD_AR1335CSSC080110)
[    6.017682] ap1302 4-003c: AP1302 revision 0.2.6 detected
[    8.962058] mx8-img-md: Registered sensor subdevice: ap1302.4-003c (1)
[    8.962084] mx8-img-md: created link [ap1302.4-003c] => [mxc-mipi-csi2.1]
```

The AP1302 sends picture via CSI and images are processed by the ISI. Processing can be checked with the interrupt:

```
root@sm2s-imx8mp:~# cat /proc/interrupts  | grep -e csi -e isi
 56:       5594          0          0          0     GICv3  48 Level     32e00000.isi
 57:      11189          0          0          0     GICv3 112 Level     32e50000.csi
```

The numbers will increase at the framerate, i.e 30fps. If there is no interrupt for a module, there is an issue. Check the V4L system:

```
root@sm2s-imx8mp:~# v4l2-ctl --list-devices
 ():
        /dev/v4l-subdev0
FSL Capture Media Device (platform:mxc-md):
        /dev/media0
vsi_v4l2dec (platform:vsi_v4l2dec):
        /dev/video1
vsi_v4l2enc (platform:vsi_v4l2enc):
```

Check the CSI interface:

```
root@sm2s-imx8mp:~# media-ctl -p
```

Check the AP1302. Its log can be obtained by:

```
root@sm2s-imx8mp:~# v4l2-ctl -d $(media-ctl -e ap1302.4-003c) --log
```

When working, the frame counters increase :

```
> [   366.858966] ap1302 4-003c: Frame counters: ICP 8328, HINF 65, BRAC 1
> [   378.555293] ap1302 4-003c: Frame counters: ICP 8599, HINF 80, BRAC 1
```

The last command that can be run is the status/log of the CSI interface to check for ECC errors:

```
root@sm2s-imx8mp:~# v4l2-ctl -d $(media-ctl -e mxc-mipi-csi2.1) --log
```

If nothing is reported, there are no ECC errors:

```
Status Log:
   [   341.657425] mxc-mipi-csi2.1: ================   START
STATUS   ================
   [   341.657436] mxc-mipi-csi2.1: ==================   END
STATUS   ==================
```

3.4 V4L Commands

Video4Linux (V4L for short) is a collection of device drivers and an API for supporting realtime video capture on Linux systems.

To find the V4L subdevice for the AP1302 use this command:

```
root@sm2s-imx8mp:~# media-ctl -e ap1302.4-003c
/dev/v4l-subdev3
```

Note that the subdevice will change between single and dual configurations.

To view the parameters to adjust video settings, use the following command with the V4L subdevice (/dev/v4l-subdev3) that you just found:

```
root@sm2s-imx8mp:~# v4l2-ctl -d /dev/v4l-subdev3 -l
```

This will list user controls for brightness, saturation, and auto-focus, among others.

As an example, to change the orientation and gamma:

```
v4l2-ctl -d $(media-ctl -e ap1302.4-003c) -c vflip=0 && \
v4l2-ctl -d $(media-ctl -e ap1302.4-003c) -c hflip=1 && \
v4l2-ctl -d $(media-ctl -e ap1302.4-003c) -c gamma=0
```

To change the white balance on the AR0144:

```
v4l2-ctl -d $(media-ctl -e ap1302.4-003c) -c white_balance_auto_preset=0
```

Software

4.1 eIQ

NXP's eIQ software enables several machine learning frameworks across their i.MX processors. To view NXP's eIQ reference material, visit [nxp.com/eiq](nxp.com/eiq).

4.1.1 eIQ inference runtime overview for i.MX8M+

eIQ optimizes machine learning on NXP's processor by using delegates to accelerate supported operations in hardware. Read more about how delegates work in eIQ here:

http://avnet.me/maaxboard-ml-delegates

The following five inference engines are currently supported in the NXP eIQ software stack: TensorFlow Lite, ONNX Runtime, PyTorch, DeepView RT, and OpenCV. Additionally, ONNX

Runtime, Tensor Flow Lite, and DeepViewRT also support acceleration on the GPU or NPU through Neural Network Runtime (NNRT).

The only inference engine currently supplied by the out of box Yocto image is Tensorflow Lite.

4.1.2 Tensorflow Lite

Tensor flow Lite Tensor Flow Lite supports hardware acceleration on i.MX8 processors sing the VX Delegate. It also supports the XNNPACK library to delegate computation on the CPU.

Note: Since Tensor Flow Lite 2.6.0, the floating point models are executed via the XNNPACK Delegate by default.

The out of box Yocto image includes Tensorflow Lite v2.6.0 and examples. Navigate to the examples folder:

```
cd /usr/bin/tensorflow-lite-2.6.0/examples
```

Here you can run the tensorflow lite benchmark and the evaluations tools imagenet_image_classification_run_eval and inference_diff_run_eval. There is also an example c++ app included to test inference. Here I will demonstrate running the C++ app on CPU, NPU, and GPU.

4.1.2.1 Run on CPU

Run the example mobilenet model on the CPU with the following commands:

```
root@sm2s-imx8mp:/usr/bin/tensorflow-lite-2.6.0/examples#
./label_image -m mobilenet_v1_1.0_224_quant.tflite    -i grace_hopper.bmp -l labels.txt
```

This should return:

```
INFO: Loaded model mobilenet_v1_1.0_224_quant.tflite
```

```
INFO: resolved reporter
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
XNNPACK delegate created.
INFO: Applied XNNPACK delegate.
INFO: invoked
INFO: average time: 49.501 ms
INFO: 0.764706: 653 military uniform
INFO: 0.121569: 907 Windsor tie
INFO: 0.0156863: 458 bow tie
INFO: 0.0117647: 466 bulletproof vest
INFO: 0.00784314: 835 suit
```

To use the XNNPACK delegate to optimize performance on the CPU, use the -use_xnnpack=true delegate.

```
root@sm2s-imx8mp:/usr/bin/tensorflow-lite-2.6.0/examples# ./label_image -m
mobilenet_v1_1.0_224_quant.tflite -i grace_hopper.bmp -l labels.txt --use_xnnpack=true
```

4.1.2.2 Run on the NPU

To run the model on the NPU hardware accelerator, add the external_delegate_path flag to apply the VX delegate:

```
./label_image -m mobilenet_v1_1.0_224_quant.tflite -i grace_hopper.bmp -l labels.txt --
external_delegate_path=/usr/lib/libvx_delegate.so
```

Optionally, you can use the NNAPI delegate to accelerate your model on the NPU by setting the –use_nnapi=true flag. However, the VX delegate is preferred and tends to be slightly faster.

4.1.2.3 Run on the GPU

To differentiate between the 3D GPU and the NPU, use the USE_GPU_INFERENCE environmental variable. For example, to run the model accelerated on the GPU, use this command:

```
USE_GPU_INFERENCE=0 ./label_image -m mobilenet_v1_1.0_224_quant.tflite -i
grace_hopper.bmp -l labels.txt
```

If USE_GPU_INFERENCE=1, the graph is executed on the GPU. If USE_GPU_INFERENCE=0, it specifies not to use the GPU. On the Edge AI Kit, GPU acceleration is the least preferred method for most models.

4.1.2.4 Verify whether acceleration is running on NPU or GPU

To verify whether hardware acceleration is running on the Verisilicon NPU, you can look at the interrupt for the galcore-3D driver before and after running your model with the following command:

```
root@sm2s-imx8mp:/usr/bin/tensorflow-lite-2.6.0/examples# cat /proc/interrupts | grep
galcore:3d
```

If the number of interrupts per clock increases (2nd column from the left; in this case 178), the model has been delegated to the NPU for this many cycles.
Similarly, you can look at the galcore 2D driver interrupt before and after running inference to see if your model is accelerated on the Vivante GPU. Check the GPU driver interrupt with the following command:

```
root@sm2s-imx8mp:/usr/bin/tensorflow-lite-2.6.0/examples# cat /proc/interrupts     | grep
galcore:2d
```

4.1.2.5 Python API

The image also includes the Tensorflow Lite interpreter-only Pythons API and example file label_image.py. The Python example supports external delegates also. The switch — ext_delegate and –ext_delegate_options can be used to specify the external delegate library and optionally its arguments.

4.1.2.6 Building your own Tensorflow Lite applications

See NXP's i.MX Machine Learning User's Guide for instructions for building your own Tensor flow Lite applications with delegation enabled.

4.1.3 Streamers Neural Network Inference example applications

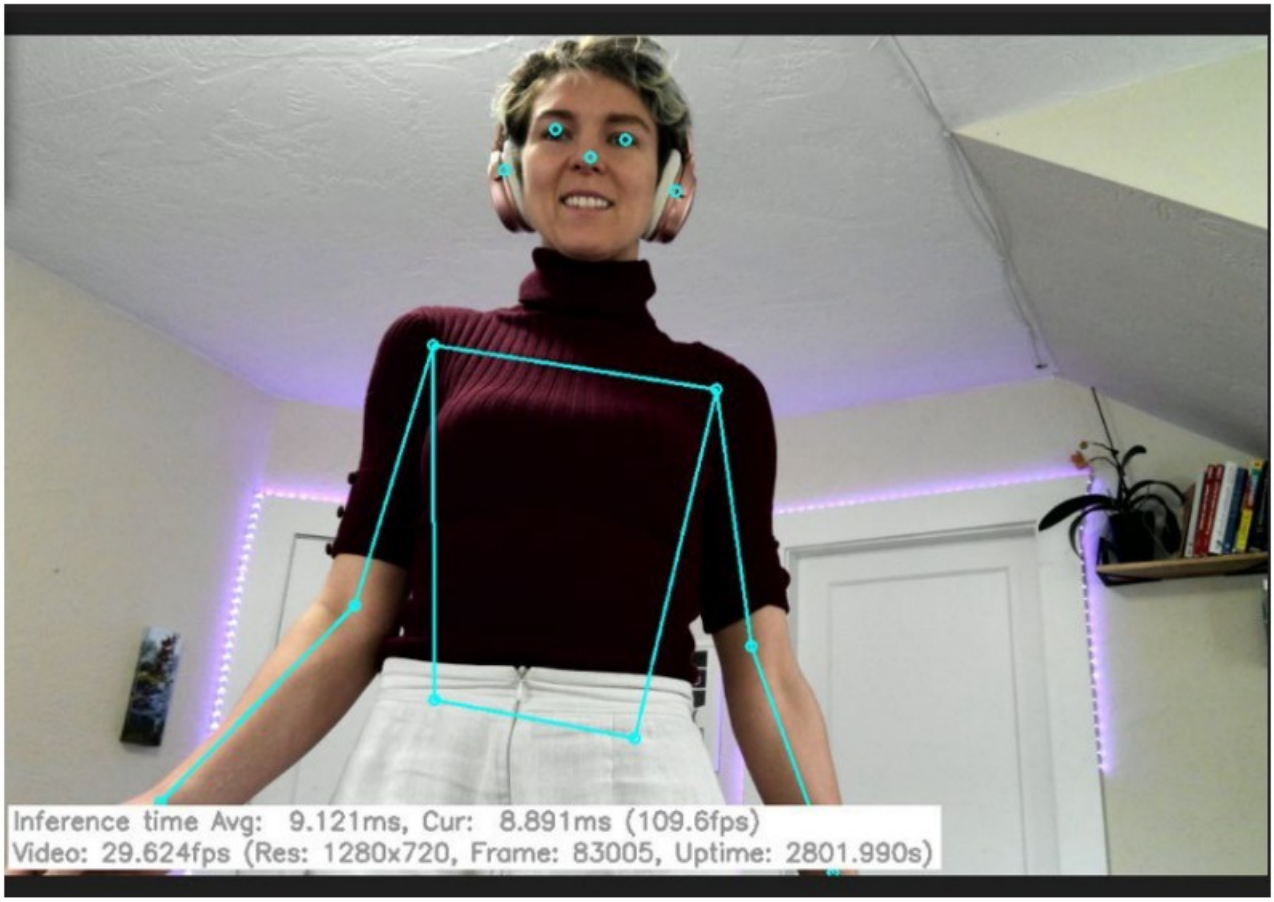
The out of box image includes four sample applications demonstrating Streamer Neural Network inference for NXP i.MX processors. It uses an object detection model (mobilome SSD) and a pose estimation model (Posenet).You can find more information about these apps on the eiq-apps-imx page on CodeAurora here: **source.codeaurora.org/external/imx/eiq-apps-imx**

You can run these examples on the board using the scripts found in /usr/bin:

- /usr/bin/gstnninferencedemo-mobilenet-ssd-camera
- /usr/bin/gstnninferencedemo-mobilenet-ssd-video
- /usr/bin/gstnninferencedemo-posenet-camera
- /usr/bin/gstnninferencedemo-posenet-video

Note that the video examples take a path to a video file as input (720p30 video is recommended), while the camera examples target a single MIPI camera. For example, for the Mobilenet SSD Camera example, run the following script:

```
gstnninferencedemo-mobilenet-ssd-camera
```



Appendix

5.1 Hardware Documents

For hardware details please refer to:

- Edge AI Kit Hardware User Guide
- Edge AI Kit Block Diagram

5.2 Software Documents

Edge AI Kit supports Yocto Linux, for additional information, please refer to the following documents accessible from the Edge AI Kit product page at **http://avnet.me/imx8mplus-edgeai**

- Edge AI Kit Linux Yocto User Manual
  – This document (describes how to reflash RZBoard and aspects of the BSP functionality)
- Edge AI Kit Linux Yocto Development Guide
  – Detailed guidance on how to rebuild the Linux system image
- The files for the QT application showing face detection and distance measurement are on Avnet's github here:
  https://github.com/Avnet/stereovision-app

5.3 Linux System Image and Application Development

5.3.1 Out of box System Image

To download operating system images, visit **http://avnet.me/imx8mplus-edgeai** and select "Reference Designs" to download the out of box Yocto image.

5.3.2 Yocto BSP

The BSP is currently available by request only. To access the board support package (BSP) to build your own Yocto image for the Edge AI Kit, contact **support.boards@avnet.com**. In your email, please clearly state that you

are working with the Edge AI Kit and would like that BSP.

5.3.3 eIQ

NXP provides Machine Learning example code as part of its eIQ software. To download eIQ, and view NXP's reference material, visit [nxp.com/eiq](nxp.com/eiq).

5.4 Contact Information

Product Page: http://avnet.me/imx8mplus-edgeai

AVNET®
Reach Further™

http://avnet.me/imx8mplus-edgeai

Documents / Resources

	AVNET i.MX 8M Plus Linux-Yocto Edge AI Kit [pdf] User Manual i.MX 8M Plus, i.MX 8M Plus Linux-Yocto Edge AI Kit, Linux-Yocto Edge AI Kit, Edge AI Kit

References

- Avnet i.MX 8M Plus Edge AI Kit | Avnet Boards
- Optimizing Machine Learning on MaaXBoard Part 1: Delegates - Blog - Single-Board Computers - element14 Community
- eIQ® ML Software Development Environment | NXP Semiconductors
- GitHub - Avnet/stereovision-app: Stereo Vision QT app created for Edge AI Kit demo
- balenaEtcher - Flash OS images to SD cards & USB drives