AN5730

Application note

Guidelines for measuring the system power consumption on the STM32MP25x line MPUs

Introduction

This application note applies to all devices of the STM32MP25x product lines. It provides power consumption values on various use cases measured on an STM32MP257F device mounted on an STM32MP257F-EV1 evaluation board and running on the STM32 MPU OpenSTLinux distribution.

This application note provides indicative typical values measured on one sample at room temperature. The measured values can vary depending on the device process lot characteristics (slow, typical, or fast) and ambient temperature.

The STM32MP25x product line devices are built on an Arm® Cortex®-A35 with dual core MPU subsystem combined with an Arm® Cortex®-M33 CPU.

The STM32MP25x product line datasheets [DS14284 and DS14285] present the power consumption values of the devices calculated using bare metal software (not using the Linux® operating system). The values are available for basic run modes and for various low-power modes such as Run2, Stop1, Stop2, LP-Stop1, LPLV-Stop1, LP-Stop2, LPLV-Stop2, Standby1, and Standby2. Refer to the [RM0457] reference manual for a full description of these modes.

Table 1. Applicable products

Reference	Product lines	Products
STM32MP25x	STM32MP251	STM32MP251A, STM32MP251C, STM32MP251D, STM32MP251F
	STM32MP253	STM32MP253A, STM32MP253C, STM32MP253D, STM32MP253F
	STM32MP255	STM32MP255A, STM32MP255C, STM32MP255D, STM32MP255F
	STM32MP257	STM32MP257A, STM32MP257C, STM32MP257D, STM32MP257F

1 General information

This document applies to the STM32MP25x Arm®-based microprocessors.

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

Reference documents

For further information on the STM32MP25x product line devices, refer to the following documents and deliverables available from the www.st.com website.

[RM0457]	Reference manual of the STM32MP25x product lines
[DS14284 and DS14285]	STM32MP25x product line datasheets
[DS14278]	STPMIC25 power management IC datasheet
[AN5726]	STM32MP2 MPUs using low-power modes, application note
[AN5489]	Getting started with the STM32MP25x MPU hardware development, application note
[DB5295]	STM32CubeMP2 embedded software for the STM32MP2 series, data brief
[wiki1]	STM32MP2 wiki article: STM32MP2 boards
[wiki2]	STM32MP2 wiki article: Unpack the STM32MP257x-EV1 board
[wiki3]	STM32MP2 wiki article: How to configure the Ethernet interface
[wiki4]	STM32MP2 wiki article: How to stream the RAW camera over a network
[wiki5]	STM32MP2 wiki article: X-LINUX-AI Starter package
[wiki6]	STM32MP2 wiki article: Power overview

1.1 Overview

As mentioned in the introduction, this application note applies to all devices of the STM32MP25x product lines even though it considers only full-featured products such as the STM32MP257F. The table below describes the main characteristics of these products.

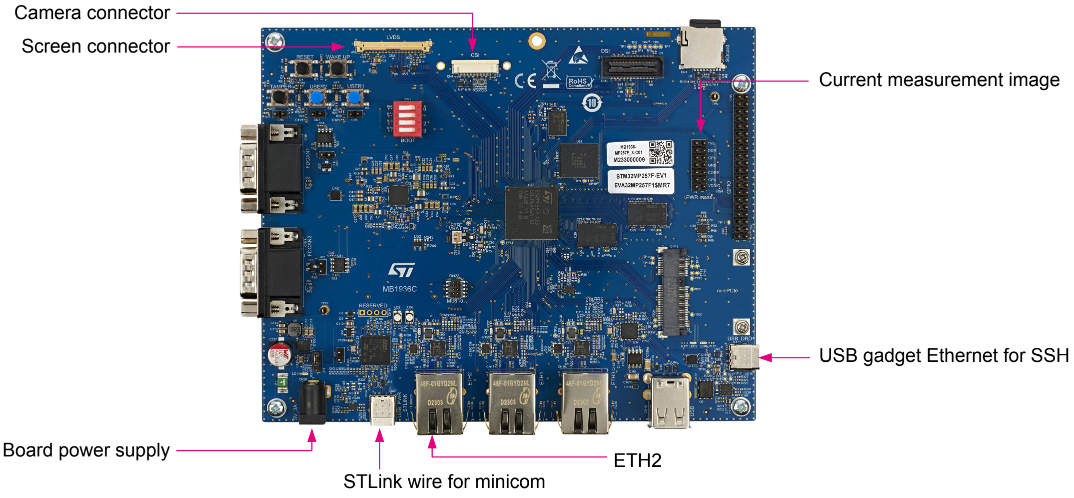
Table 2. Configurations of the STM32MP25x product line devices

Product line	Reference manual	DSI/LVDS	DCMIPP	FDCAN	GPU	Core
STM32MP251	RM0457	No	Yes	No	No	Single
STM32MP253		No	Yes	Yes	No	Dual
STM32MP255		Yes	Yes	Yes	Yes	Dual
STM32MP257		Yes	Yes	Yes	Yes	Dual

2 STM32MP257F-EV1 board

This application note identifies the STM32MP257F-EV1 board as the evaluation board (MB1936D).
For a detailed description of the evaluation board, refer to the article on the [wiki1] web page.

Figure 1. Main board MB1936D



2.1 Power supply overview

To optimize the power consumption, the power domains are split in several parts and which are regrouped in different power supplies provided by the PMIC.
Here is a table which described which measurement is connected to which supply domain and if it is linked to the SOC or to the board.

Table 3. Board power supply overview

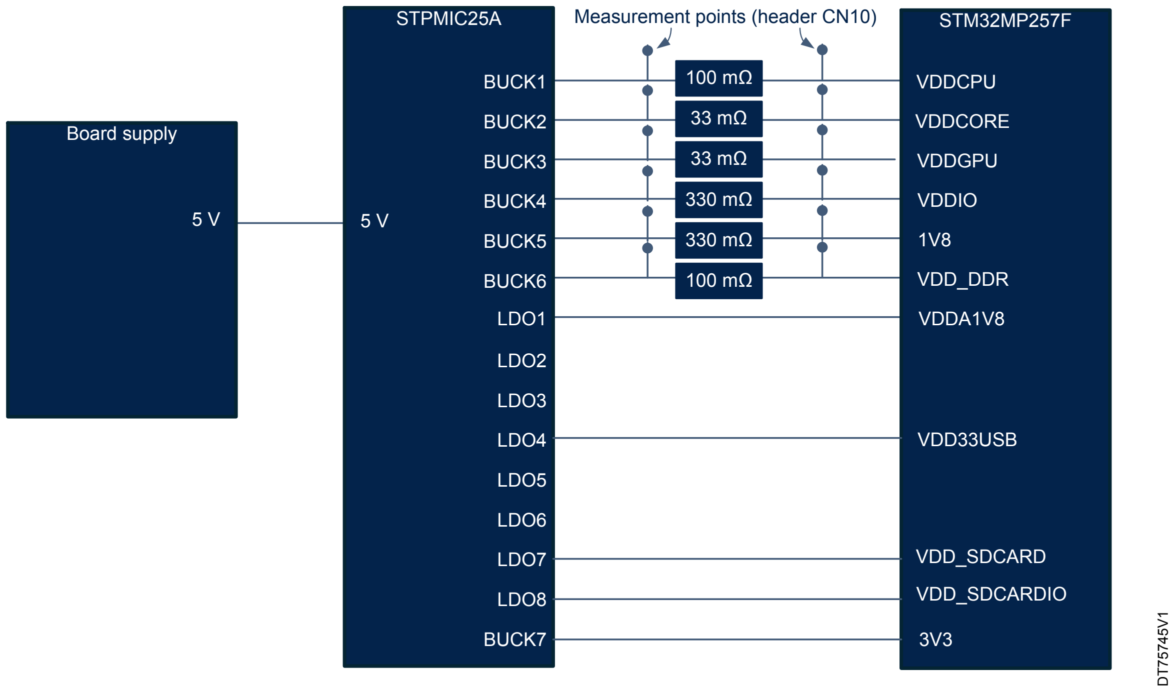
Measured supply	Supply domain	Group
VDDCPU	VDDCPU	SOC
VDDGPU	VDDGPU	SOC
1V8	VDDA18PLL1	SOC
	VDDA18PLL2	SOC
	VDDA18PLL3	SOC
	VDDA18USB	SOC
	VDDIO2	SOC
	VDDA18COMBOPHY	SOC
	VDD18LVDS	SOC
	VDDA18CSI	SOC
	VDDA18DSI	SOC
	VDDA18ADC	SOC
	VDDA18DDR	SOC
	DSI connector	Board
VDDCORE	VDDCORE	SOC
	VDDCSI	SOC
	VDDDSI	SOC

Measured supply	Supply domain	Group
VDDCORE	VDDLVDS	SOC
	VDDCOMPBOPHY	SOC
	VDDCOMBOPHYTX	SOC
	VDDPCIECLK	SOC
VDDIO	VDD	SOC
	VDDIO3	SOC
	VDDIO4	SOC
VDD_DDR	VDDQDDR	SOC
	DDR	Board
	VTT_DDR	Board

2.2 Measurement points

On the board, the STPMIC is connected to each power supplies with serial resistors with small values which make possible the measurement of the driven current. And then by knowing the voltages, one can deduce the power. However, some modifications of the board have to be done to make this possible.

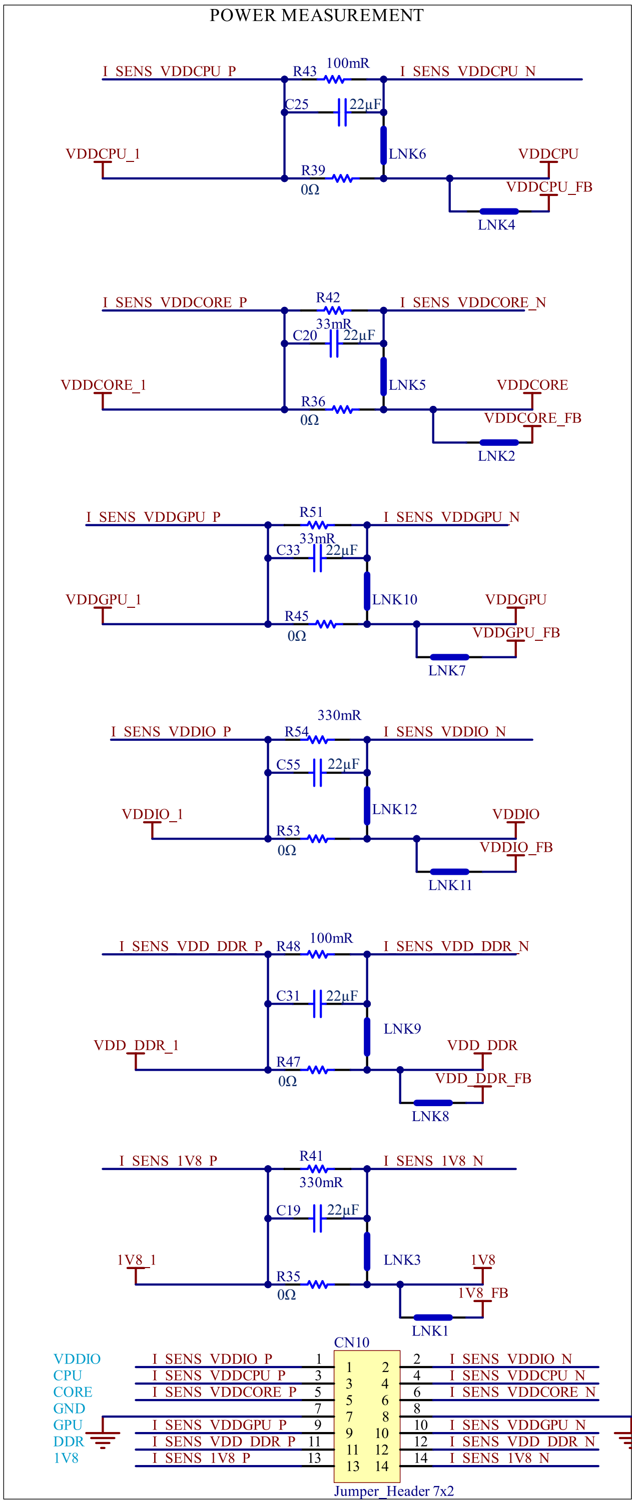
Figure 2. Block diagram of power control



Indeed, on the STM32MP257F-EV1 board, one cannot directly measure the consumption of the various voltage domains: V_{DD}, V_{DDCORE}, V_{DD_DDR}, and V_{DDCPU}.

Once the shunt resistors are removed, the V_{DDCORE}, V_{DD}, V_{DD_DDR}, V_{1V8}, V_{DDGPU}, and V_{DDCPU} supply paths include a serial 0.033 Ω, 0.33 Ω, 0.1 Ω, 0.33 Ω, 0.033 Ω, or 0.1 Ω resistor as illustrated below.

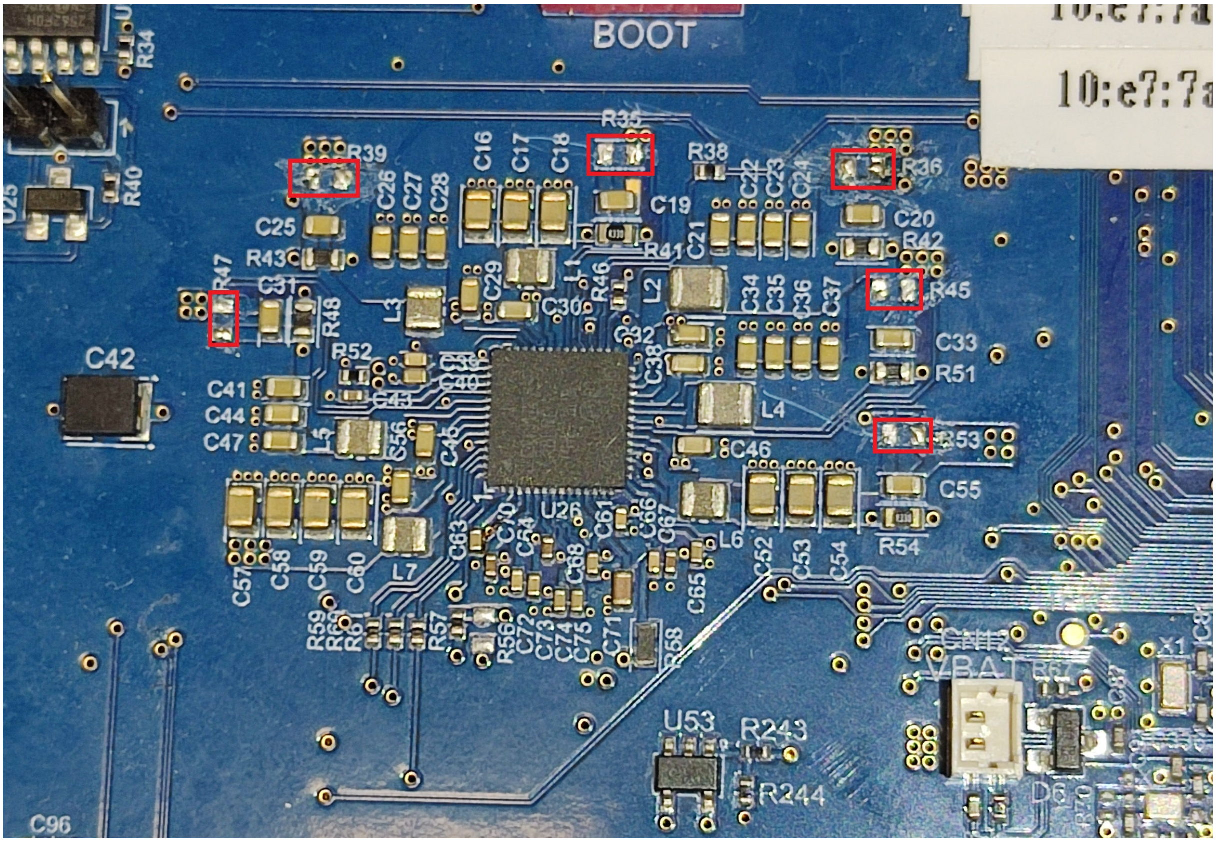
Figure 3. Power supply lines



DT75747.V1

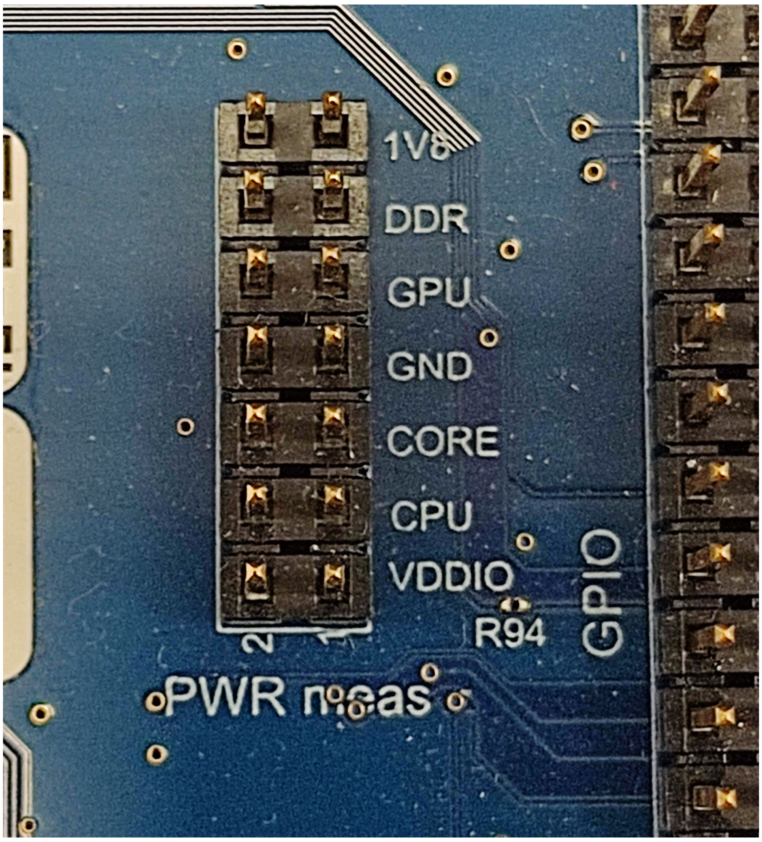
To make the measurements provided in this application note, one must modify the MB1936D board as follows: remove the following 0 Ω shunt resistors present on the front side of the board (near U28): R35, R36, R39, R45, R47, and R53. See the figure below to locate their position.

Figure 4. Resistors to remove for current measurement



One can measure the voltage drop across the serial resistors on the power supply lines using the test points illustrated in the figure below. From them, one can deduce the current consumption.

Figure 5. **Measurement points on the board**



2.3 STM32MP2 OpenSTLinux distribution

Unless otherwise specified, all measurements were made using the STM32MP2 OpenSTLinux distribution starter package:

ST OpenSTLinux—Weston (a Yocto project-based distro): stm32mp2-openstlinux-6.6-yocto-scarthgap-mpu-v24.12.05 version

3 Power consumption on the STM32MP25x product lines

This section provides some use cases for power consumption measurements using the STM32MP257F device mounted on an STM32MP257F-EV1 board.

3.1 CoreMark® on a Linux® MPU

First step: prepare the CoreMark® executable

Prerequisite: required material:

- One evaluation board: STM32MP257F-EV1
- The STM32MP2 developer package SDK
- A USB-C® cable

In the following, the USB-C® cable is connected to the USB gadget on the connector CN15 to copy files (`scp`) or launch commands (`ssh`).

The sequence below indicates the commands to execute on the Linux® station.

1. Download CoreMark® from the EEMBC (Embedded microprocessor Benchmark consortium) website:

```
PC $> mkdir ~/Coremark
PC $> cd ~/Coremark
PC $> git clone https://github.com/eembc/coremark.git
```

2. Create a specific STM32MP25 folder:

```
PC $> mkdir -p coremark/STM32MP25_linux
PC $> cp coremark/linux/* coremark/STM32MP25_linux
PC $> mkdir -p coremark/STM32MP25_linux/posix
PC $> cp coremark/posix/* coremark/STM32MP25_linux/posix
```

3. Update `coremark/STM32MP25_linux/posix/core_portme.mak`.
   To do so, execute the following steps:
   a. Comment the lines 24 and 36 as shown below:

```
#CC? = cc
#LFLAGS_END += -lrt
```

   b. Update the lines 60 and 61 with `IP@` for the Ethernet connection (uncomment the line 61 for the run cases):

```
LOAD = scp $(OUTFILE) root@<IP@>:/home/root/
#RUN = ssh root@<IP@> -c /home/root/
```

   c. Comment the lines 65 and 66 as shown below:

```
#LOAD =   echo Loading done
#RUN =
```

Note: To get `<IP@>`, run the `ifconfig` command on the target.

4. Update the makefile:

```
PC $> cd ~/Coremark/coremark
PC $> gedit Makefile &
```

   Remove the space between LOAD and OUTFILE in the makefile at the line 122:

```
$(LOAD)$(OUTFILE)
```

5. Use the source of the developer package to compile for the STM32MP25. (The path depends on where the sources are installed.)

```
PC $> source ~/STM32MP25/Developer-Package/SDK/environment-setup-cortexa35-ostl-linux
```

6. Compile CoreMark® and run `coremark.exe`:

```
PC $> cd ~/Coremark/coremark
PC $> make PORT_DIR=STM32MP25_linux XCFLAGS="-DMULTITHREAD=2 -DUSE_PTHREAD -pthread"
```

7. If the run failed and generate some errors, make the *.exe executable if not already done:

```
PC $> chmod +x coremark.exe
```

8. Copy the CoreMark® executable onto the MB1936D SD™ card if not done by the `.mak` script:
   Copy the file to the target by using `scp`:

```
PC $> scp coremark.exe root@<IP@>:/usr/local/
```

Note: Administrator rights on the Linux® station are required to run `sudo`.

Second step: power measurements

To run CoreMark® on the board, run the commands below (called a 2K performance in this case):

```
Board $> cd /usr/local
Board $> ./coremark.exe > run.log
Board $> cat run.log
```

The table below shows the consumption measurements made on CoreMark®.
One can check the V_{DDCPU} voltage by measuring the voltage across "GND" and "CPU" on the "PWR meas" connector (Figure 5. Measurement points on the board.) In this way, one can force the MPU clock and the V_{DDCPU} power supply to one of the two schemes below:

- 0.8 V @ 1.2 GHz (default)
- 0.91 V @ 1.5 GHz

Run the following commands: (1.5 GHz scheme illustrated. Replace 1.500.000 by 1.200.000 to scale the frequency.)

```
Board $> cd /sys/devices/system/cpu/cpu0/cpufreq/
Board $> echo 1500000 > scaling_min_freq
Board $> echo 1500000 > scaling_max_freq
```

Run the following command to print the temperature in millidegrees celsius (about 37.5°C in the example below).

```
Board $> cat /sys/devices/virtual/thermal/thermal_zone0/temp
37489
```

Table 4. CoreMark®[1] power consumption of the STM32MP257F + MB1936D

Power supply scheme	Power supply domain	Current (mA)	Voltage (V)	Power (mW)	Sum of Powers per group	Iterations per second	Die temperature
Group_SOC	V_{DDCPU}	154	0.8	124	486	6031	38
	V_{DDGPU}	0	0	0			
	V_{DDCORE}	349	0.81	283			
	V_{DDIO}	9	3.3	29			
	1V8[2]	29	1.79	51			
Group_Board	V_{DD_DDR}	178	1.19	212	212		
Total	-	-	-	-	699		

1. *CoreMark® on the MPU core (1200 MHz) VDD_CPU = 0.8 V*
2. *1V8 can supply board features such as the DSI connector. This is an example of measurement, and the board is not optimized.*

Table 5. **CoreMark®[1] power consumption of the STM32MP257F + MB1936D**

Power supply scheme	Power supply domain	Current (mA)	Voltage (V)	Power (mW)	Sum of Powers per group (mW)	Iterations per second	Die temperature (°C)
Group_SOC	V_{DDCPU}	234	0.91	213	579	7543	40
	V_{DDGPU}	0	0	0			
	V_{DDCORE}	352	0.81	285			
	V_{DDIO}	9	3.3	29			
Group_Board	1V8[2]	29	1.79	52			
	V_{DD_DDR}	178	1.19	212	212		
Total	-	-	-	-	791		

1. *CoreMark® on the MPU core (1500 MHz) VDD_CPU = 0.91 V*
2. *1.8V can supply board features such as the DSI connector. This is an example of measurement, and the board is not optimized.*

3.2 Graphical use case: streaming a camera over the network

The goal of this use case is to stream camera data over a network to another board, which then displays the stream. Then, the consumption is measured on both boards.

The figure below shows how the boards are connected to the Linux® station.

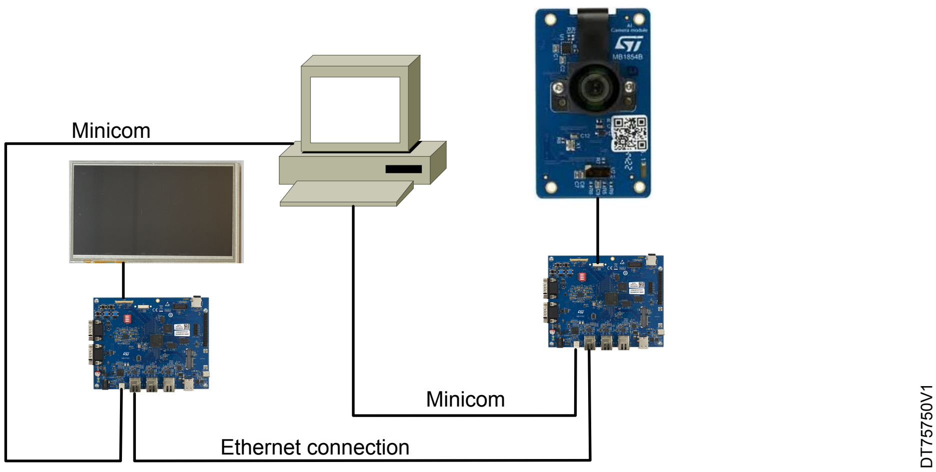


Figure 6. **Interconnection diagram**

Note: Useful information is available at the [wiki2] web page.

The graphical use case is built with the following settings:

- Camera sensor: B-CAM-IMX (Sony IMX335, not provided with the evaluation board)
- Two evaluation boards: STM32MP257F-EV1 (shunt resistors have to be removed on both boards. See Section 2: STM32MP257F-EV1 board.)
- An Ethernet cable
- Two USB Type-C® cables for the STLink virtual communications port (VCP)
  Unplug the cable of the USB gadget for Ethernet if plugged, and plug a USB Type-C® cable into STLink wire for minicom instead on both boards. Refer to Section 2.2: Measurement points.

3.2.1 Setting up the Ethernet

For useful information on how to proceed, go to the [wiki3] web page.

To set up the Ethernet, follow the sequence below:

1. Plug the Ethernet cable into the ETH2 connector of each board to connect the two boards together. This connector corresponds to the first network interface (end0).
2. Plug the two USB Type-C® cables into the STLink virtual communications port (VCP) of each board to connect the two boards to the Linux® station.
3. Execute the commands below to connect the Linux® station to the boards on two different terminals:

```
PC $> sudo minicom -D /dev/ttyACM0
PC $> sudo minicom -D /dev/ttyACM2
```

4. To configure the Ethernet, execute the following on both boards:

```
Board $> cd /lib/systemd/network/
Board $> vi 52-static.network.static
```

5. Replace or use the address 192.168.72.2/24 on one board and 192.168.72.3/24 on the other board.
6. Finally, on both boards, enable static IP on eth0/end0:

```
Board $> cp 80-wired.network 80-wired.network.notused
Board $> cp 52-static.network.static 52-static.network
Board $> systemctl restart systemd-networkd.service
```

The network is now configured and you can use one board to ping the other.

3.2.2 Setting up the camera

To set up the camera, plug the camera cable into the CSI connector (CN4), called camera connector in Section 2.2: Measurement points.

Before launching the pipeline to stream the camera to the screen, it can be useful to do the following actions:

- Check if the camera is present.
- Set up the camera and the digital camera interface pipe processing (DCMIPP).

These two actions are carried out through scripts. Refer to Appendix A: cam_check.sh, and Appendix B: cam_control.sh at the end of this document for the scripts to use.

The sequence below indicates the commands to execute on the board to get the scripts ready.

1. Copy the scripts from the demo folder:

```
Board_Cam $> mkdir /usr/local/an5730
Board_Cam $> cp /usr/local/demo/application/camera/bin/check_camera_preview_mp25.sh
/usr/local/an5730/cam_check.sh
Board_Cam $> cp /usr/local/demo/application/camera/bin/launch_camera_control_mp25.sh
/usr/local/an5730/cam_control.sh
```

2. Update /usr/local/an5730/cam_control.sh.
   To do so, execute the following steps:
   a. Comment the line 125 as shown below:

   ```
   # config_dcmipp_media_ctl $WIDTH $HEIGHT $FPS
   ```

   b. Add the line 130 below:

   ```
   config_dcmipp_media_ctl $WIDTH $HEIGHT $FPS
   ```

   c. Add the lines 140 to 142 below:

   ```
   echo "device="$(media-ctl -d $DCMIPP_MEDIA -e "dcmipp_main_capture")" ! "$GST_CAPS
   #Kill exposure correction background task
   killall cam_control.sh
   ```

Checking that a camera is present

To check if a camera is present, execute `cam_check.sh`:

```
Board_Cam $> /usr/local/an5730/cam_check.sh
/usr/local/an5730
mediadev=/dev/media0
sensorsubdev="imx335 0-001a"
sensordev= /dev/v4l-subdev7
interfacesubdev=
```

Setting the camera parameters

To set the camera parameters, execute `cam_control.sh`:

```
Board_Cam $> /usr/local/an5730/cam_control.sh
With GPU
sensor_subdev=imx335 0-001a
sensor_dev=/dev/v4l-subdev7
bridge_subdev=48020000.csi
bridge_dev=/dev/v4l-subdev6
sensorbuscode=SRGGB12_1X12
SRGGB10_1X10
SENSORWIDTH=2592
SENSORHEIGHT=1940
Mediacontroller graph:
  media-ctl -d platform:48030000.dcmipp -l "'dcmipp_input':1->'dcmipp_dump_postproc':0[0]"
  media-ctl -d platform:48030000.dcmipp -l "'dcmipp_input':2->'dcmipp_main_isp':0[1]"
  media-ctl -d platform:48030000.dcmipp --set-v4l2 "'imx335 0-001a':0[fmt:SRGGB10_1X10/2592x1
940]"
  media-ctl -d platform:48030000.dcmipp --set-v4l2 "'48020000.csi':1[fmt:SRGGB10_1X10/2592x19
40]"
  media-ctl -d platform:48030000.dcmipp --set-v4l2 "'dcmipp_input':2[fmt:SRGGB10_1X10/2592x19
40]"
  media-ctl -d platform:48030000.dcmipp --set-v4l2 "'dcmipp_main_isp':1[fmt:RGB888_1X24/2592x
1940 field:none]"
  media-ctl -d platform:48030000.dcmipp --set-v4l2 "'dcmipp_main_postproc':0[compose:(0,0)/64
0x480]"

device=/dev/video1
video/x-raw, format=RGB16, width=640, height=480
```

3.2.3 Setting up the pipeline between the boards

For useful information on how to proceed, go to the [wiki4] web page.

To set up the pipeline between boards, follow the sequence below:

1. Open the pipeline on the board with the screen with the following command:

```
Board_Screen $> gst-launch-1.0 udpsrc port=5000 ! application/x-rtp, encoding-name=JPEG !
rtpjpegdepay ! jpegparse ! decodebin ! videoconvert ! autovideosink sync=false
```

2. Then, launch the stream as follows:

```
Board_Cam $> gst-launch-1.0 v4l2src device=/dev/video2 ! video/x-raw, format=RGB16, width=
640,height=480, framerate=30/1 ! encodebin profile="image/jpeg" ! rtpjpegpay ! udpsink
host=192.168.72.3 port=5000
```

This opens a window with the camera stream on the screen.

Note: `device=/dev/video2` and `format=RGB16` are deduced from the execution of the script `cam_control.sh`.

```
GStreamer graph:
v4l2src device=/dev/video2 ! video/x-raw, format=RGB16, width=640, height=480
```

3.2.4 Power consumption tables

The table below summarizes the consumption of the different use cases with the MPU clock running at 1200 MHz. The total power is obtained by adding all the power measurements from the following V_{DDs}: V_{DD_CORE}, V_{DD}, V_{DD_DDR}, V_{1V8}, V_{DD_DDR}, and V_{DD_CPU}.

Table 6. Power consumptions of the board with camera with the MPU clock at 1200 MHz

Power supply scheme	Power supply domain	Current (mA)	Voltage (V)	Power (mW)	Sum of Powers per group (mW)	Die temperature (°C)
Group_SOC	V_{DDCPU}	34	0.8	27	433	38
	V_{DDGPU}	0	0	0		
	V_{DDCORE}	361	0.82	296		
	V_{DDIO}	17	3.3	55		
	1V8	31	1.8	55		
Group_Board	V_{DD_DDR}	182	1.2	219	219	
Total	-	-	-	-	652	

Table 7. Power consumptions of the board with screen with the MPU clock at 1200 MHz

Power supply scheme	Power supply domain	Current (mA)	Voltage (V)	Power (mW)	Sum of Powers per group (mW)	Die temperature (°C)
Group_SOC	V_{DDCPU}	32	0.8	25	524	38
	V_{DDGPU}	33	0.9	29		
	V_{DDCORE}	365	0.82	299		
	V_{DDIO}	17	3.3	56		
	1V8	64	1.8	114		
Group_Board	V_{DD_DDR}	194	1.2	233	233	
Total	-	-	-	-	757	

The CPU consumption is greater when the camera module is on, but the GPU is on when the screen shows the camera module view. That is why the overall consumption is quite similar.

The table below summarizes the consumption of the different use cases with the MPU clock running 1500 MHz.

Table 8. Power consumptions of the board with camera with the MPU clock at 1500 MHz

Power Supply Scheme	Power Supply Domain	Current (mA)	Voltage (V)	Power (mW)	Sum of Powers per group (mW)	Die temperature (°C)
Group_SOC	V_{DDCPU}	45	0.91	41	448	41
	V_{DDGPU}	0	0	0		
	V_{DDCORE}	361	0.82	296		
	V_{DDIO}	17	3.3	55		
	1V8	31	1.8	56		
Group_Board	V_{DD_DDR}	182	1.2	219	219	
Total	-	-	-	-	666	

Table 9. Power consumptions of the board with screen with the MPU clock at 1500 MHz

Power Supply Scheme	Power Supply Domain	Current (mA)	Voltage (V)	Power (mW)	Sum of Powers per group (mW)	Die temperature (°C)
Group_SOC	V_{DDCPU}	42	0.91	38	536	39
	V_{DDGPU}	32	0.9	29		
	V_{DDCORE}	365	0.82	299		
	V_{DDIO}	17	3.3	56		
	1V8	63	1.8	113		
Group_Board	V_{DD_DDR}	195	1.2	234	234	
Total	-	-	-	-	769	

3.3 Artificial intelligence use case: object detection

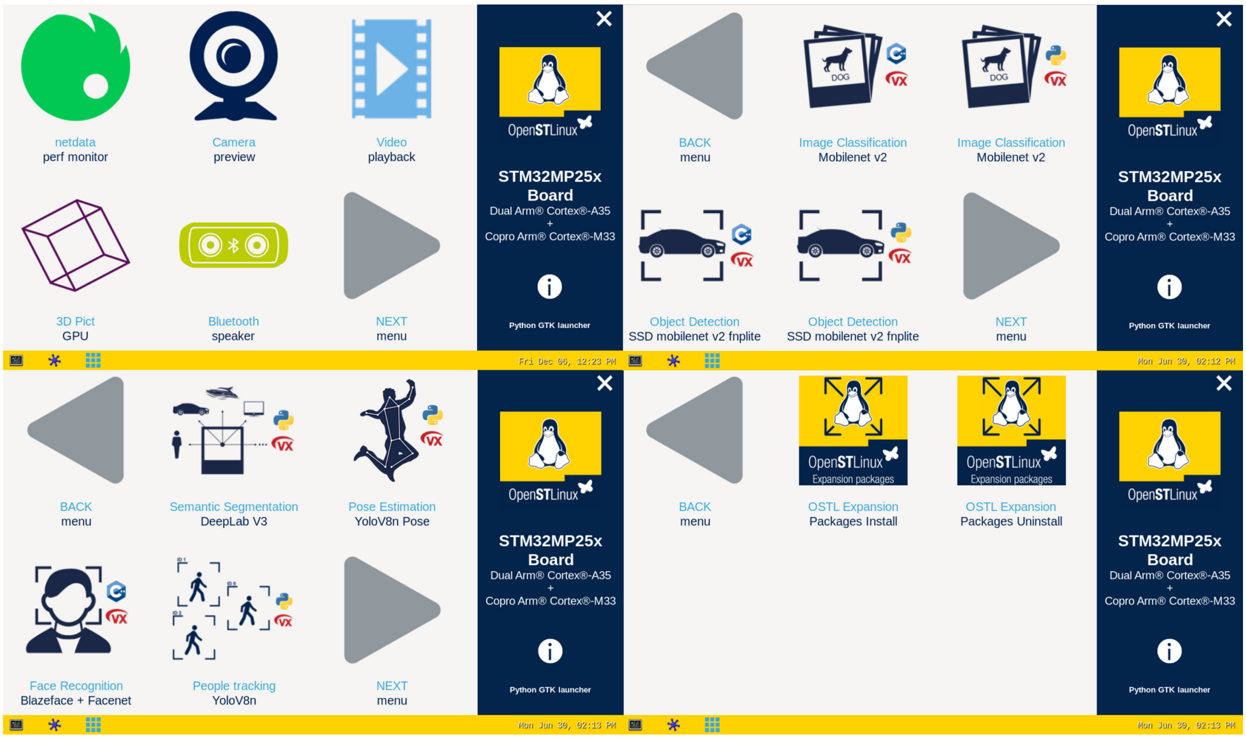
The graphical use case is built with the following settings:

- One evaluation board: STM32MP257F-EV1
- Camera sensor: B-CAM-IMX (Sony IMX335, not provided with the evaluation board)
- Display screen: B-LVDS7-WSVGA (7" TFT LCD module with capacitive touch panel, not provided with the evaluation board)
- An Ethernet cable
- One USB-C® cable for the STLink wire for minicom

This use case uses an object detection model of the demonstration package from the X-LINUX-AI tool. For the installation of the X-LINUX-AI tool and the demonstration package used in this use case, follow the [wiki5] web page.

Once the instructions on the [wiki5] web page have been followed, the demo launcher should look like the following:

Figure 7. **X-LINUX-AI demo launcher**



DT77183V1

Click on "Object Detection SSD mobilenet v2 fnplite" to launch the object detection model.

Table 10. **Power consumptions of the board with object detection model with the MPU clock at 1500 MHz**

Power supply scheme	Power supply domain	Current (mA)	Voltage (V)	Power (mW)	Sum of powers per group (mW)	Die temperature (°C)
Group_SOC	V_{DDCPU}	104	0.91	95	726	39
	V_{DDGPU}	144	0.9	130		
	V_{DDCORE}	427	0.82	350		
	V_{DDIO}	9	3.3	29		
	1V8	68	1.8	122		
Group_Board	V_{DD_DDR}	311	1.2	374	374	
Total	-	-	-	-	1099	

3.4 Operating modes available in OpenSTLinux

The STM32MP25x product line devices operate in various power modes. Refer to [DS14284 and DS14285] for further information. You can also find useful information at the [wiki6] web page.

OpenSTLinux provides two variations of the CRun operating mode and several low-power modes. Table 11. Power consumption in various power modes with STM32MP257F + MB1936D reports these modes and provides the expected device consumption for each mode.

The measurements are provided on the board MB1936D using an "ev1" Weston distribution, which means that some current on the V_{DD} supply is dissipated in low-power mode. This dissipation is due to some board components such as the screen or Ethernet.

Low-power use cases

1. Run1:
   This is the default mode in OpenSTLinux.
2. Run2:
   Execute the following sequence to bring the board to Run2 mode during 60 seconds before returning to the previous mode.

```
Board $> cd /usr/local/Cube-M33-examples/STM32MP257F-EV1/Demonstrations/LowPower_SRAM_Demo
/
Board $> ./fw_cortex_m33.sh start
Board $> systemctl stop netdata
Board $> echo deep > /sys/power/mem_sleep
Board $> echo enabled > $(find /sys/bus/platform/devices/soc*/ -name usb1)/power/wakeup
Board $> echo +60 > /sys/class/rtc/rtc0/wakealarm; systemctl suspend
```

3. The Stop1, LP-Stop1, and LPLV-Stop1 low-power modes are **not** enabled in the provided distribution. The low-power modes listed below are used instead.
   a. Stop2:
      It requires that Stop2 be implemented in the device tree.
   b. LP-Stop2:
      The activation of a wake-up source that prevents the device from entering a deeper low-power mode is needed. For that purpose, one can use the UART wake-up source.

```
Board $> echo enabled > /sys/class/tty/ttySTM0/power/wakeup
Board $> echo mem > /sys/power/state
```

   c. LPLV-Stop2:
      This mode requires the activation of GPIO wake-up in the device tree.
   d. Standby1 DDR in self-refresh (SR)
      The command below resets the board (to clean up any unwanted wake-up source settings).

```
Board $> echo mem > /sys/power/state **A35 in CStop and system in Standby1, DDR in SR**
```

   e. Standby2 DDR OFF:
      Reset the board or press the wake-up button. The command below resets the board.

```
Board $> shutdown -P now **A35 in CStop with PDDS=1 and system in Standby2, DDR Off**
```

   f. VBAT2:
      Connect a power source to the VBAT connector (CN12), then power down V_{DD} (one way to do so can be by removing the power source selection connector JP4).

Table 11. Power consumption in various power modes with STM32MP257F + MB1936D

Low-power mode	Power supply scheme	Power supply domain	Current (mA)	Voltage (V)	Power (mW)	Sum of Powers per group (mW)
Run1	Group_SOC	V_{DDCPU}	17	0.8	13	368
		V_{DDGPU}	0	0	0	
		V_{DDCORE}	333	0.82	273	
		V_{DDIO}	9	3.3	30	
		1V8	28	1.8	51	

Low-power mode	Power supply scheme	Power supply domain	Current (mA)	Voltage (V)	Power (mW)	Sum of Powers per group (mW)
Run1	Group_Board	V_{DD_DDR}	179	1.2	215	215
	Total	-	-	-	-	582
Run2	Group_SOC	V_{DDCPU}	0	0	0	182
		V_{DDGPU}	0	0	0	
		V_{DDCORE}	149	0.82	123	
		V_{DDIO}	8	3.3	25	
		1V8	19	1.8	34	
	Group_Board	V_{DD_DDR}	32	1.2	39	39
	Total	-	-	-	-	221
LP-Stop2	Group_SOC	V_{DDCPU}	0	0	0	38
		V_{DDGPU}	0	0	0	
		V_{DDCORE}	13	0.82	11	
		V_{DDIO}	8	3.3	25	
		1V8	2	1.8	3	
	Group_Board	V_{DD_DDR}	11	1.2	13	13
	Total	-	-	-	-	51
Standby1	Group_SOC	V_{DDCPU}	0	0	0	8
		V_{DDGPU}	0	0	0	
		V_{DDCORE}	0	0	0	
		V_{DDIO}	2	3.3	8	
		1V8	0	0	0	
	Group_Board	V_{DD_DDR}	10	1.2	12	12
	Total	-	-	-	-	20
Standby2	Group_SOC	V_{DDCPU}	0	0	0	8
		V_{DDGPU}	0	0	0	
		V_{DDCORE}	0	0	0	
		V_{DDIO}	2	3.3	8	
		1V8	0	0	0	
	Group_Board	V_{DD_DDR}	0	0	0	0
	Total	-	-	-	-	8

Table 12. Power consumption in VBAT2 mode with STM32MP257F + MB1936D

Low-power mode	Power supply scheme	Power supply domain	Current (µA)	Voltage (V)	Power (µW)	Sum of powers per group (µW)
VBAT2	Group_SOC	V_{DDCPU}	0	0	0	105
		V_{DDGPU}	0	0	0	
		V_{DDCORE}	0	0	0	
		V_{DDIO}	32	3.3	105	
		1V8	0	0	0	
	Group_Board	V_{DD_DDR}	0	0	0	0
	Total	-	-	-	-	105

For more details on the power modes of the STM32MP25x product lines, refer to [AN5726].

Appendix A cam_check.sh

```
#!/bin/sh

is_dcmipp_present() {
  DCMIPP_SENSOR="NOTFOUND"
  # on disco board ov5640 camera can be present on csi connector
  for video in $(find /sys/class/video4linux -name "video*" -type l);
  do
    if [ "$(cat $video/name)" = "dcmipp_main_capture" ]; then
      cd $video/device/
      mediadev=/dev/$(ls -d media*)
      cd -
      for sub in $(find /sys/class/video4linux -name "v4l-subdev*" -type l);
      do
        subdev_name=$(tr -d '\0' < $sub/name | awk '{print $1}')
        #HFR FIXME get rid of driver name
        if [ "$subdev_name" = "gc2145" ] || [ "$subdev_name" = "ov5640" ] || [ "$subdev_name"
 = "imx335" ]; then
          DCMIPP_SENSOR=$subdev_name
          V4L_DEVICE="device=/dev/$(basename $video)"
          sensorsubdev="$(tr -d '\0' < $sub/name)"
          sensordev=$(media-ctl -d $mediadev -p -e "$sensorsubdev" | grep "node name"
| awk -F\name '{print $2}')
          #interface is connected to input of isp (":1 [ENABLED" with media-ctl -p)
          interfacesubdev=$(media-ctl -d $mediadev -p -e "dcmipp_main_isp" | grep ":1
\[ENABLED" | awk -F\" '{print $2}')
          echo "mediadev="$mediadev
          echo "sensorsubdev=\""$sensorsubdev\"
          echo "sensordev="$sensordev
          echo "interfacesubdev="$interfacesubdev

          return
        fi
      done
    fi
  done
}

get_webcam_device() {
  WEBCAM_found="NOTFOUND"
  for video in $(find /sys/class/video4linux -name "video*" -type l | sort);
  do
    if [  $(cat $video/name | grep -q "dcmi";echo $?) -ne 0 ] &&  [ $(cat $video/name
| grep -q "stm";echo $?) -ne 0 ]; then
      WEBCAM_found="FOUND"
      break;
    fi
  done
}

# ------------------------------
#         main
# ------------------------------

# camera detection
# detect if we have a gc2145 or ov5640 plugged and associated to dcmipp
is_dcmipp_present
if [ "$DCMIPP_SENSOR" != "NOTFOUND" ]; then
  exit 0
fi
get_webcam_device
if [ "$WEBCAM_found" = "FOUND" ]; then
  exit 0
fi
exit 1
```

Appendix B cam_control.sh

```
#!/bin/sh

cmd() {
  cmd=$1
  echo "$cmd"
  eval "$cmd" > /dev/null 2>&1
}

# DCMIPP bus info
DCMIPP_MEDIA="platform:48030000.dcmipp"

get_webcam_device() {
  for video in $(find /sys/class/video4linux -name "video*" -type l | sort);
  do
    if [ "$(cat "$video/name")" != "dcmipp_main_capture" ]; then
      V4L_DEVICE="device=/dev/$(basename "$video")"
      break;
    fi
  done
}

config_dcmipp_media_ctl() {
  width=$1
  height=$2
  fps=$3

  # Starting from the dcmipp_input go down until the first source subdev
  current_subdev="dcmipp_input"
  source_subdev="dcmipp_input"
  while [ -n "$source_subdev" ]; do
    source_subdev=$(media-ctl -d $DCMIPP_MEDIA -p -e "$current_subdev" | grep '<-' | grep -v
'dcmipp_tpg' | awk -F\" '{print $2}')
    if [ -n "$source_subdev" ]; then
      current_subdev=$source_subdev
    fi
  done
  sensor_subdev=$current_subdev
  sensor_dev=$(media-ctl -d $DCMIPP_MEDIA -e "$sensor_subdev")
  bridge_subdev=$(media-ctl -d $DCMIPP_MEDIA -p -e "$sensor_subdev" | grep '\->' | awk -F\"
'{print $2}')
  if [ "$bridge_subdev" = "dcmipp_input" ]; then
    bridge_subdev=""
  else
    bridge_dev=$(media-ctl -d $DCMIPP_MEDIA -e "$bridge_subdev")
  fi
  echo "sensor_subdev=$sensor_subdev"
  echo "sensor_dev=$sensor_dev"
  if [ "$bridge_subdev" != "" ]; then
    echo "bridge_subdev=$bridge_subdev"
    echo "bridge_dev=$bridge_dev"
  fi

  #Use sensor in raw-bayer format
  sensorbuscode=$(v4l2-ctl --list-subdev-mbus-codes -d "$sensor_dev" | grep SRGGB
| awk -FMEDIA_BUS_FMT_ '{print $2}')
  echo "sensorbuscode=$sensorbuscode"

  SENSORWIDTH=0
  SENSORHEIGHT=0
  case "$sensor_subdev" in
    *"ov5640"*)
      #OV5640 only support 720p with raw-bayer format
      SENSORWIDTH=1280
      SENSORHEIGHT=720
      #OV5640 claims to support all raw bayer combinations but always output SBGGR8_1X8...
      sensorbuscode=SBGGR8_1X8
      ;;
```

```
    *"imx335"*)
      #IMX335 expose both RGGB10 and RGGB12 however only RGGB10 can work on CSI 2 lanes
      sensorbuscode=SRGGB10_1X10
      main_postproc=$(media-ctl -d $DCMIPP_MEDIA -e dcmipp_main_postproc)
      #Enable gamma correction
      v4l2-ctl -d "$main_postproc" -c gamma_correction=1
      #Do exposure correction continuously in background
      sleep 3  && while : ; do /usr/local/demo/bin/dcmipp-isp-ctrl -i0 -g > /dev/null ; done
&
      ;;
  esac

  #Let sensor return its prefered resolution & format
  media-ctl -d $DCMIPP_MEDIA --set-v4l2
"'$sensor_subdev':0[fmt:$sensorbuscode/${SENSORWIDTH}x${SENSORHEIGHT}@1/${fps} field:none]" >
 /dev/null 2>&1
  sensorfmt=$(media-ctl -d $DCMIPP_MEDIA --get-v4l2 "'$sensor_subdev':0" | awk -F"fmt:"
'{print $2}' | awk -F" " '{print $1}')
  SENSORWIDTH=$(echo "$sensorfmt" | awk -F"/" '{print $2}' | awk -F"x" '{print $1}')
  SENSORHEIGHT=$(echo "$sensorfmt" | awk -F"/" '{print $2}' | awk -F"x" '{print $2}'
| awk -F" " '{print $1}' | awk -F"@" '{print $1}')
  echo "SENSORWIDTH=$SENSORWIDTH"
  echo "SENSORHEIGHT=$SENSORHEIGHT"

  #Use main pipe for debayering, scaling and color conversion
  echo "Mediacontroller graph:"
  cmd "  media-ctl -d $DCMIPP_MEDIA -l \"'dcmipp_input':1->'dcmipp_dump_postproc':0[0]\""
  cmd "  media-ctl -d $DCMIPP_MEDIA -l \"'dcmipp_input':2->'dcmipp_main_isp':0[1]\""
  cmd "  media-ctl -d $DCMIPP_MEDIA --set-v4l2
\"'$sensor_subdev':0[fmt:$sensorbuscode/${SENSORWIDTH}x${SENSORHEIGHT}]\""
  if [ "$bridge_subdev" != "" ]; then
    cmd "  media-ctl -d $DCMIPP_MEDIA --set-v4l2
\"'$bridge_subdev':1[fmt:$sensorbuscode/${SENSORWIDTH}x${SENSORHEIGHT}]\""
  fi
  cmd "  media-ctl -d $DCMIPP_MEDIA --set-v4l2
\"'dcmipp_input':2[fmt:$sensorbuscode/${SENSORWIDTH}x${SENSORHEIGHT}]\""
  cmd "  media-ctl -d $DCMIPP_MEDIA --set-v4l2
\"'dcmipp_main_isp':1[fmt:RGB888_1X24/${SENSORWIDTH}x${SENSORHEIGHT} field:none]\""
  cmd "  media-ctl -d $DCMIPP_MEDIA --set-v4l2
\"'dcmipp_main_postproc':0[compose:(0,0)/${width}x${height}]\""
  echo ""
}

# ------------------------------
#         main
# ------------------------------

# graphic backend detection
if [ -f /etc/default/weston ] && $(grep "^OPTARGS" /etc/default/weston | grep -q "use-pixman"
 ) ;
then
  echo "Without GPU"
  ADDONS="videoconvert ! queue !"
else
  echo "With GPU"
  ADDONS=""
fi

WIDTH=640
HEIGHT=480
FPS=30
FMT=RGB16

# Check if dcmipp is available
if media-ctl -d $DCMIPP_MEDIA > /dev/null 2>&1; then
  comp_board=$(tr -d '\0' < /proc/device-tree/compatible | sed "s|^st,|;|" | cut -d';' -f2
| head -n 1 |tr '\n' ' ' | sed "s/ //g")
  # for the time being, libcamera is only enabled on MP25-EVAL & MP25-DK
  if $(echo $comp_board | grep -qG "stm32mp2[0-9]*[abcdef]-ev1") || $(echo $comp_board
| grep -qG "stm32mp2[0-9]*[abcdef]-dk") ; then
```

```
    GST_SOURCE="libcamerasrc name=cs src::stream-role=view-finder cs.src"
  else
    # config_dcmipp_media_ctl $WIDTH $HEIGHT $FPS
    V4L_DEVICE="device=$(media-ctl -d $DCMIPP_MEDIA -e "dcmipp_main_capture")"
    V4L_OPT=""
    GST_SOURCE="v4l2src $V4L_DEVICE $V4L_OPT"
  fi
  config_dcmipp_media_ctl $WIDTH $HEIGHT $FPS
  GST_CAPS="video/x-raw, format=$FMT, width=$WIDTH, height=$HEIGHT"

else
  get_webcam_device
  # suppose we have a webcam
  V4L_OPT="io-mode=4"
  GST_SOURCE="v4l2src $V4L_DEVICE $V4L_OPT"
  GST_CAPS="video/x-raw, width=$WIDTH, height=$HEIGHT"
  v4l2-ctl --set-parm=20
fi

echo "device="$(media-ctl -d $DCMIPP_MEDIA -e "dcmipp_main_capture")" ! "$GST_CAPS

#Kill exposure correction background task
killall cam_control.sh
```

Revision history

Table 13. **Document revision history**

Date	Revision	Changes
05-Feb-2025	1	Initial release based on OpenSTLinux 5.1.
12-Aug-2025	2	Updated: • Section 2.3: STM32MP2 OpenSTLinux distribution. • Section 3.1: CoreMark® on a Linux® MPU. • Section 3.2.2: Setting up the camera. • Section 3.2.3: Setting up the pipeline between the boards. • All the tables in the Section 3.2.4: Power consumption tables. • Section 3.4: Operating modes available in OpenSTLinux and Table 11. Power consumption in various power modes with STM32MP257F + MB1936D. • Appendix A: cam_check.sh. • Appendix B: cam_control.sh. Added: • [wiki6] to the Section 1: General information. • Section 3.3: Artificial intelligence use case: object detection. Deleted: • Appendix C cam_launch.sh

Contents

List of tables

List of figures