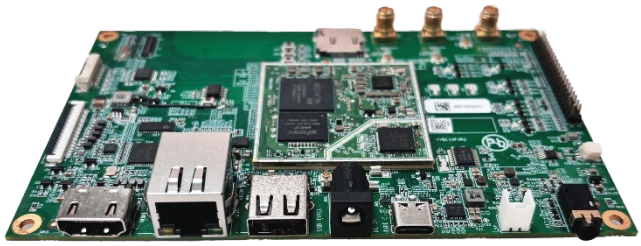
VT-SBC-VOSM700-EVB
Evaluation Board



User Manual

Version: 1.3

Revision History

No.	Version	Description	Date
1	V1.0	First release	May 28, 2025
2	V1.1	Updated pinout description of the GPIO header.	Oct. 13, 2025
3	V1.2	Added test procedures for the audio connectors.	Feb. 10, 2026
4	V1.3	Aligned the content with the Wiki documentation format: 1. Added sections 3.1 and 3.2 on quick startup of the board. 2. Removed sections related to custom development. 3. Updated sections 3.4 “Image Flashing” and 3.5 “Device Access”. 4. Added section 3.5.3 “SSH Login”	March 30, 2026

Table of Contents

Foreword

Thank you for purchasing VT-SBC-VOSM700-EVB evaluation board ("the Board" or "the Product"). This manual intends to provide guidance and assistance necessary on setting up, operating or maintaining the Product. Please read this manual and make sure you understand the functionality of the Product before putting it into use.

Intended Users

This manual is intended for:

- Embedded software developer
- Custom development software engineer
- Other technically qualified personnel

Copyright

Disclaimer

While all information contained herein has been carefully checked to assure its accuracy in technical details and typography, Vantron does not assume any responsibility resulting from any error or features of this manual, nor from improper uses of this manual or the software.

It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without notice.

Technical Support and Assistance

Should you have any question about the Product that is not covered in this manual, contact your sales representative for solution. Please include the following information in your question:

- Product name and PO number;
- Complete description of the problem;
- Error message you received, if any.

Vantron Technology, Inc.

Address: 48434 Milmont Drive, Fremont, CA 94538

Tel: (650) 422-3128

Email: sales@vantrontech.com

Symbology

This manual uses the following signs to prompt users to pay special attention to relevant information.

	Caution for latent damage to system or harm to personnel
	Attention to important information or regulations

General Safety Instructions

The Product is supposed be installed by knowledgeable, skilled persons familiar with local and/or international electrical codes and regulations. For your safety and prevention of damage to the Product, please read and observe carefully the following safety instructions prior to installation and operation. Keep this manual well for future reference.

- Do not disassemble or otherwise modify the Product. Such action may cause heat generation, ignition, electronic shock, or other damages including human injury, and may void your warranty.
- Keep the Product away from heat source, such as heater, heat dissipater, or engine casing.
- Do not insert foreign materials into any opening of the Product as it may cause the Product to malfunction or burn out.
- To ensure proper functioning and prevent overheating of the Product, do not cover or block the ventilation holes of the Product.
- Follow the installation instructions with the installation tools provided or recommended.
- The use or placement of the operation tools shall comply with the code of practice of such tools to avoid short circuit of the Product.
- Cut off the power before inspection of the Product to avoid human injury or product damage.

Precautions for Power Cables and Accessories

⚠ Use proper power source only. Make sure the supply voltage falls within the specified range.

⚠ Place the cables properly at places without extrusion hazards.

⚠ There is a coin cell battery for powering the RTC. Therefore, please avoid short circuit of the battery during transportation or operation at high temperatures.

⚠ Cleaning instructions:

- Power off before cleaning the Product
- Do not use spray detergent
- Clean with a damp cloth
- Do not try to clean exposed electronic components unless with a dust collector

⚠ Power off and contact Vantron technical support engineer in case of the following faults:

- The Product is damaged
- The temperature is excessively high
- Fault is still not solved after troubleshooting according to this manual

⚠ Do not use in combustible and explosive environment:

- Keep away from combustible and explosive environment
- Keep away from all energized circuits
- Unauthorized removal of the enclosure from the device is not allowed
- Do not change components unless the power cable is unplugged
- In some cases, the device may still have residual voltage even if the power cable is unplugged. Therefore, it is a must to remove and fully discharge the device before replacement of the components.

CHAPTER 1 INTRODUCTION

1.1 Product Overview

The VT-SBC-VOSM700-EVB evaluation board provides a complete, production-ready development ecosystem based on the VOSM700 system-on-module. Designed to accelerate time-to-market, it offers developers a fully integrated platform with standard connectors, robust power management, and signal protection—eliminating common hardware integration hurdles and ensuring reliable operation.

At its core is the powerful MediaTek MT8390 chipset, paired with 4GB LPDDR4x and 64GB eMMC. The board features an octa-core CPU and an Arm Mali-G57 MC2 GPU, enabling 4K HDMI/MIPI DSI display and 4K video CODEC for optimal multimedia performance. The 5th-Gen NPU delivers up to 4 TOPS of computing power for efficient AI inference and on-device machine learning tasks. Connectivity options include Wi-Fi, Bluetooth, and Gigabit Ethernet, making it highly versatile for diverse IoT applications. A wide array of expansion interfaces further enables seamless peripheral integration to fully utilize the board's capabilities.

By offering a fully validated, application-ready hardware foundation, this evaluation board dramatically shortens development cycles and mitigates project risks, allowing engineering teams to focus on vertical-specific customization rather than hardware challenges. It supports Android and Linux Yocto operating systems, with option available for other Linux distributions, making it ideal for a range of advanced applications, including smart retail, self-service terminals, industrial automation, and digital media.

1.2 Product Feature

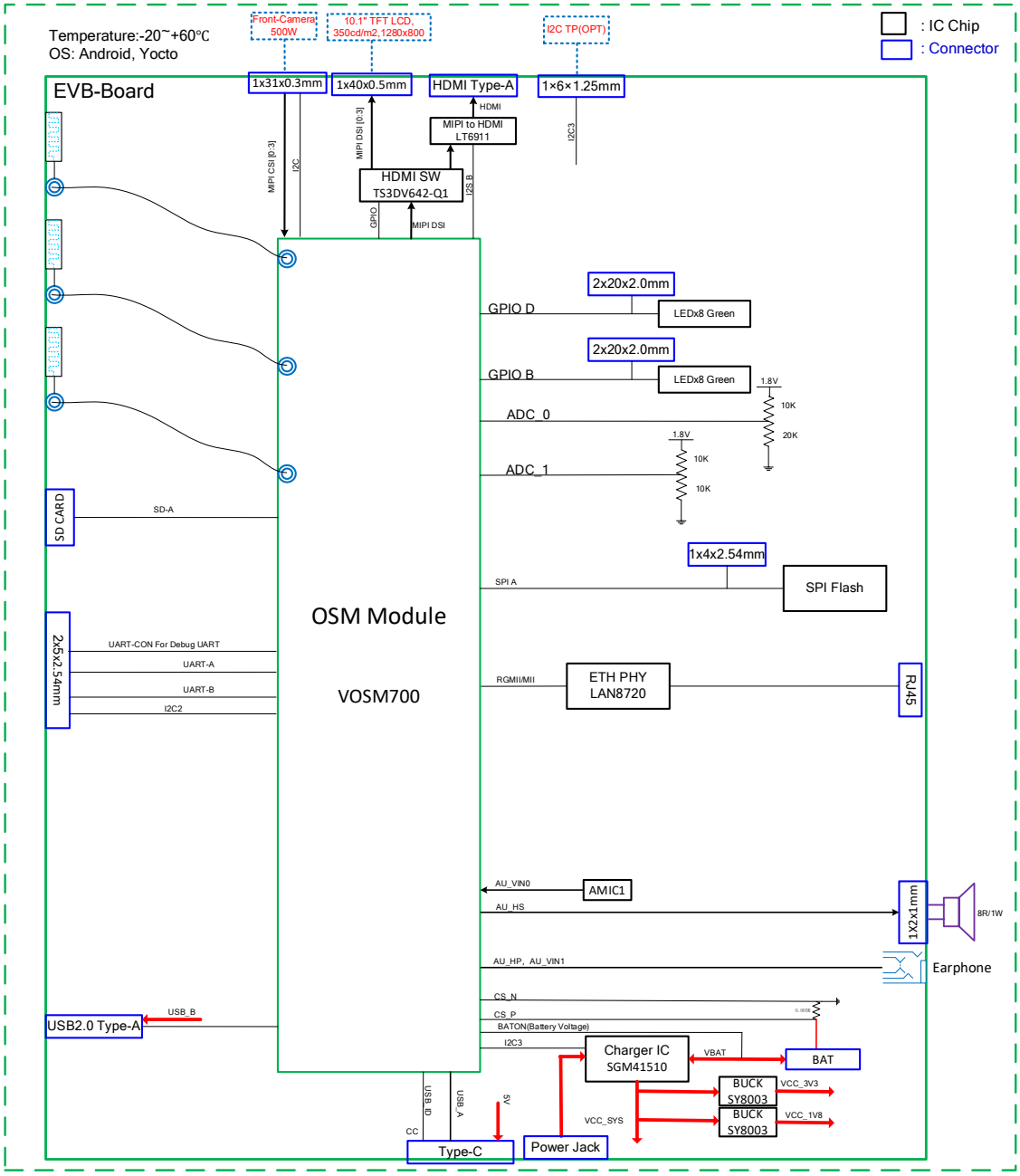
- MTK MT8390 (G700), Dual-core ARM Cortex-A78 (2.2GHz) + Hexa-core ARM Cortex-A55 (2.0GHz) processor
- Arm Mali-G57 MC3 GPU, 950MHz, Supports OpenGL ES 3.2, Vulkan 1.1, OpenCL 2.2
- MediaTek 5th-Gen NPU (MDLA3.0 + Tensilica VP6), 4.0 TOPS
- Tensilica HiFi 5 audio DSP
- Onboard 4GB memory + 64GB storage for responsive multitasking
- Display support with MIPI DSI at 1080p@30Hz / HDMI at 1080p@120Hz or 4096 x 2304@60Hz
- Ethernet, Wi-Fi 5, BT 5.2 connectivity
- 4-Lane MIPI CSI interface, supporting 32MP@30fps for camera ISP
- Rich interfaces for flexible expansion (GPIO, UART, USB, I^2C, TP)

1.3 Terminology/Acronym

Please refer to the table below for acronyms or terminologies used in this document, especially for those included in the pinout description of the device.

Terminology/Acronym	Description
NC	No connection
VCC	Voltage common collector
GND	Ground
P (+)	Positive difference signal
N (-)	Negative difference signal
SCL	Serial clock
SDA	Serial data
I	Input
O	Output
I/O	Input/output
P	Power/ground
RX	Receive data
TX	Transmit data
PCIe	Peripheral component interconnect express
MDI	Media dependent interface
INT	Interrupt
RST	Reset
HDMI	High-Definition Multimedia Interface
I^2C	Inter-Integrated Circuit

1.4 Block Diagram



1.5 Specifications

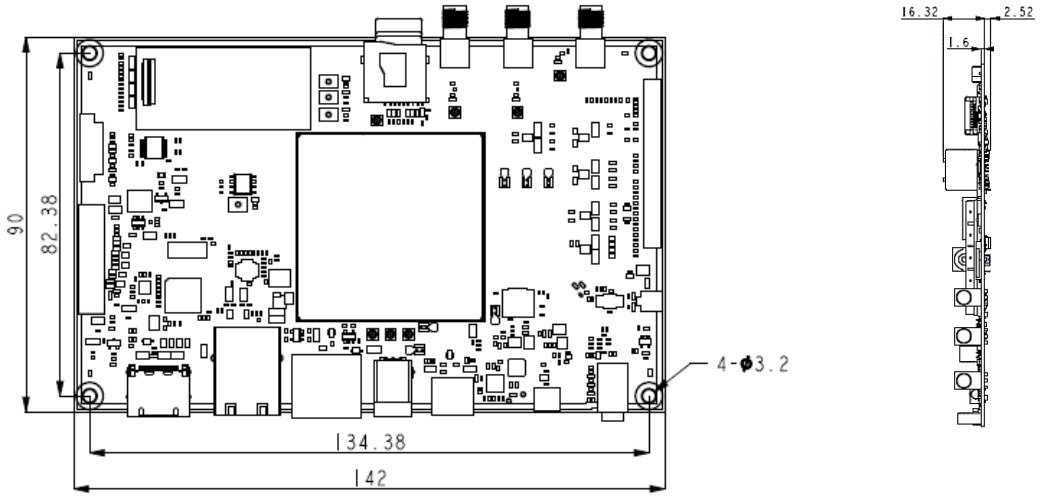
VT-SBC-VOSM700-EVB Evaluation Board			
System	CPU	MediaTek MT8390, Dual-core ARM Cortex-A78 (2.2GHz) + Hexa-core ARM Cortex-A55 (2.0GHz)	
	GPU	Arm Mali-G57 MC3 GPU, 950MHz, Supports OpenGL ES 3.2, Vulkan 1.1, OpenCL 2.2	
	NPU	MediaTek 5th-Gen NPU (MDLA3.0 + Tensilica VP6), 4.0 TOPS	
	Memory	4GB 64-bit LPDDR4x	
	Storage	64GB eMMC 5.1	
Communication	Ethernet	1 x RJ45, 10M/100Mbps	
	Wi-Fi & Bluetooth	Wi-Fi 802.11 a/b/g/n/ac + Bluetooth 5.2	
Media	Video processing	4K@30Hz, H.265/H.264 video encoding 4K@75Hz, H.265/H.264/VP9/AV1 video decoding	
	Audio DSP	Tensilica HiFi 5	
I/O	Display	1 x 4-Lane MIPI DSI, 1080p@30Hz / 1 x HDMI, 1080p@120Hz or 4K@60Hz	
	MIPI CSI	1 x 4-lane MIPI CSI, 32MP@30fps	
	Audio	1 x 3.5mm Combo audio jack (CTIA)	1 x A-Mic
	USB	1 x USB 2.0 Type-A	1 x USB Type-C (USB 2.0 OTG)
	Battery	1 x Battery connector	
	GPIO header	9 x GPIO, 1 x Debug UART (1.8V), 2 x Communication UART (1.8V), 1 x PWM, 1 x I^2C, 1 x JTAG	
	SD slot	1 x Micro SD slot	
	TP	1 x I^2C TP	
	Key	1 x Power key	1 x Reset key
		1 x Volume -	1 x Recovery Key
Power	Input	5V/3A DC input	1 x Power jack
Software	Operating system	Android, Linux Yocto	
	Device management	BlueSphere MDM (Optional for Android version)	
Mechanical	Dimensions	142mm x 90mm x 18.84mm (EVB)	45mm x 45mm (SOM)
Environmental Condition	Temperature	Operating: -20℃ ~ +60℃ (Optional: -40℃ ~ +80℃)	Storage: -40℃ ~ +80℃
	Humidity	5%~95% RH (Non-condensing)	

1.6 Operating system

The VT-SBC-VOSM700-EVB supports Android 13 and later operating systems, with option for Linux Yocto and other distributions. This manual elaborates on the use of the Yocto system.

1.7 Mechanical Dimensions

- 142mm x 90mm x 18.84mm



1.8 Memory and Storage

1.8.1 LPDDR4 RAM

The VT-SBC-VOSM700-EVB comes with 4GB 64-bit LPDDR4x for responsive multitasking.

1.8.2 eMMC Flash

The VT-SBC-VOSM700-EVB provides an eMMC 5.1 flash, offering a default capacity of 64GB, used as the default boot and storage device. The board also offers a Micro SD for storage expansion.

1.9 Power Supply and Consumption

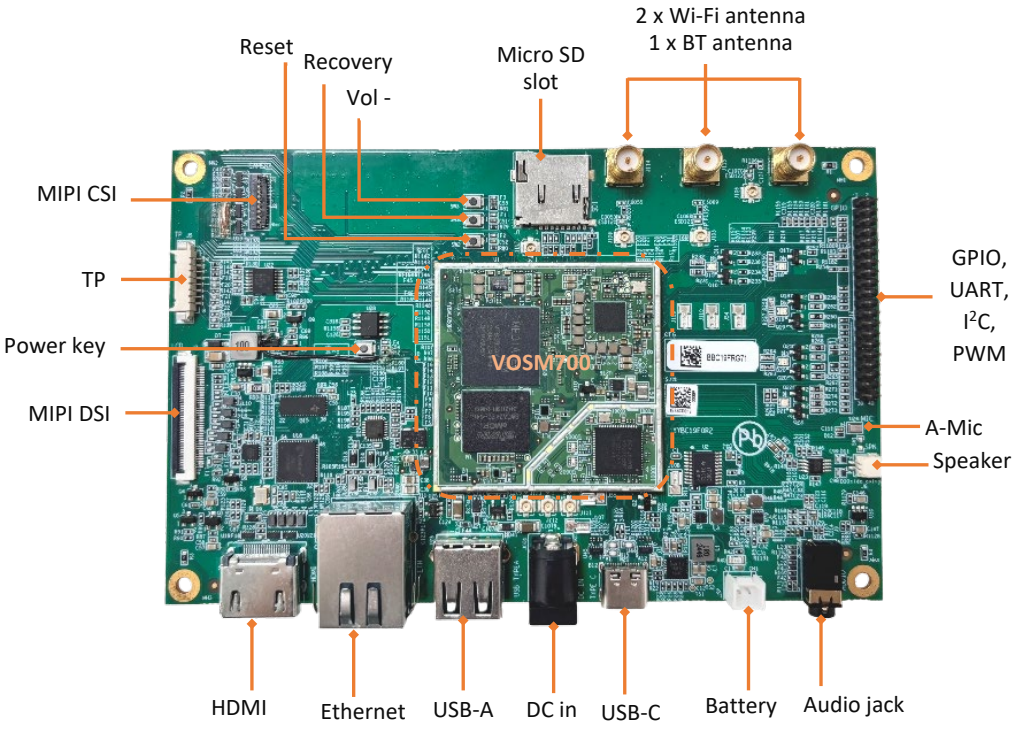
The VT-SBC-VOSM700-EVB supports 5V~12V DC power input via a DC jack. To ensure safe operation, the input power is recommended not to exceed 12V⎓3A. Excessive input power may cause overheating or damage to the circuit board.

1.10 Environmental Conditions

The VT-SBC-VOSM700-EVB operates at temperatures ranging from -20°C to +60°C, with an optional extended temperature range of -40°C to +80°C. It is designed for storage within a temperature range of -30°C to +70°C. The board is intended for use and storage in environments with a relative humidity of 0% to 95%, non-condensing.

CHAPTER 2 HARDWARE DESCRIPTION

2.1 Product Layout

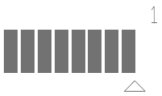


2.2 Connectors and Jumpers

The board offers a variety of interfaces to connect different peripherals, enabling extended functionality. This section is going to brief the connectors/jumpers on the board with corresponsive pinout description.

2.2.1 Identification of Pin 1

Unless otherwise stated, pin 1 of a connector is seated on a square pad that is different from the round pads used for other pins. Sometimes, pin 1 is next to a trigonal mark on the board. When there are two rows of pins on a connector, the row with pin 1 is composed of odd numbers and the other is composed of even numbers.



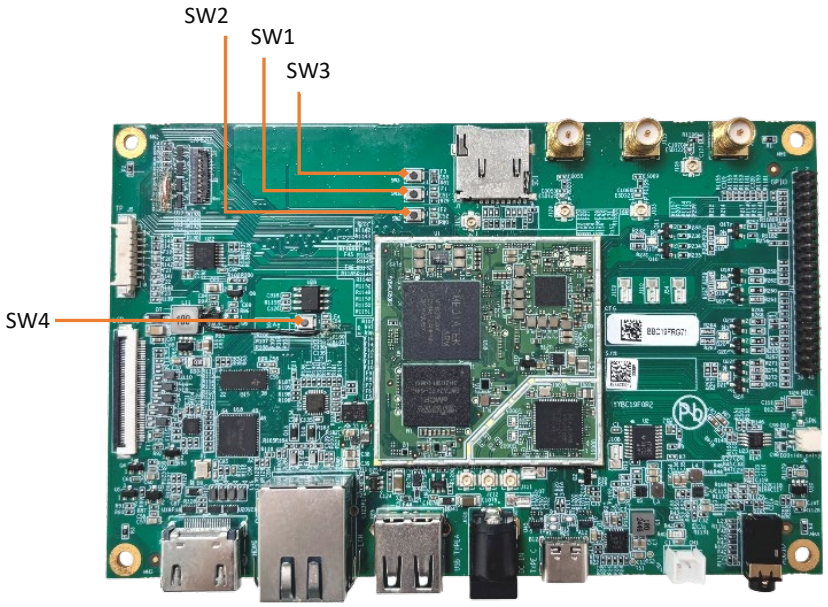
Usually, there will be numbers or marks next to the pins of a connector on the board to indicate the pinouts.

2.2.2 J001 Power supply

The VT-SBC-VOSM700-EVB provides a 2.5mm/5.5mm DC jack for 5V power input. The power adapter included in the packing list provides 5V⎓3A input.

2.2.3 Button

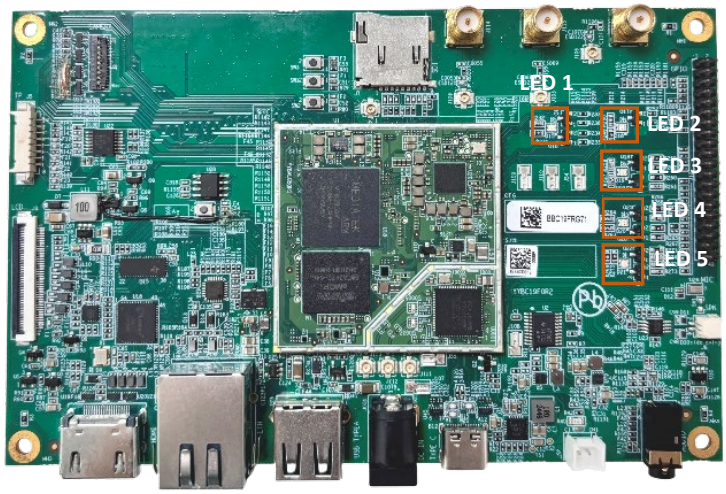
The evaluation board supports four normally open push-buttons for power on/off, reset, recovery (download), and user-defined functions.



Button	Description
SW3 **Home**	Reserved. Can be verified at the software level (Section 3.6.7)
SW1 **Recovery**	To enter the flashing mode: 1. Press and hold SW1 before powering on the board → power on the board → release the button; **OR** 2. Press and hold SW1 → press SW2 → release both buttons.
SW2 **Reset**	Short press (< 500ms) to reset and restart the board.
SW4 **Power**	Short press (< 500 ms) to turn on/off the system.

2.2.4 LED Indicator

The evaluation board features five dual-color LED indicators as shown below.



The LED indicators display red and green colors. They are inactive by default and are configurable via software.

2.2.5 Debug Interface

The VT-SBC-VOSM700-EVB is equipped with two debug interfaces: debug UART and USB Type-C.

2.2.5.1 J12 Debug UART

The debug UART is included in the GPIO header.

Specifications:

- Location: J12 (2 x 20, 2.0mm pitch)
- Voltage: 1.8V level
- Baud rate: 921600
- Pinout: See the figure above

2.2.5.2 J1 USB Type-C

The evaluation board provides a USB 2.0 Type-C port that supports OTG for firmware flashing via ADB. By default, the evaluation board operates as a Device.

Specifications:

- Location: J1
- Mode: Device / Host
- Max. output: 5 V⎓0.5 A
- Overcurrent protection supported
- Charging after power-off is NOT supported

2.2.6 U26 USB Type-A

The evaluation board offers a USB 2.0 Type-A, allowing users to connect external peripherals and extend the board's functionality.

Specifications:

- Location: U26
- Mode: Host
- Max. output: 5 V⎓0.5 A
- Overcurrent protection supported
- Charging after power-off is NOT supported

2.2.7 J11 Ethernet port

The VT-SBC-VOSM700-EVB offers an RJ45 Ethernet jack that supports a transmission rate of 10/100Mbps.

The Ethernet jack is designated as a WAN port, intended for connecting to a router or switch to establish Internet access.

Specifications:

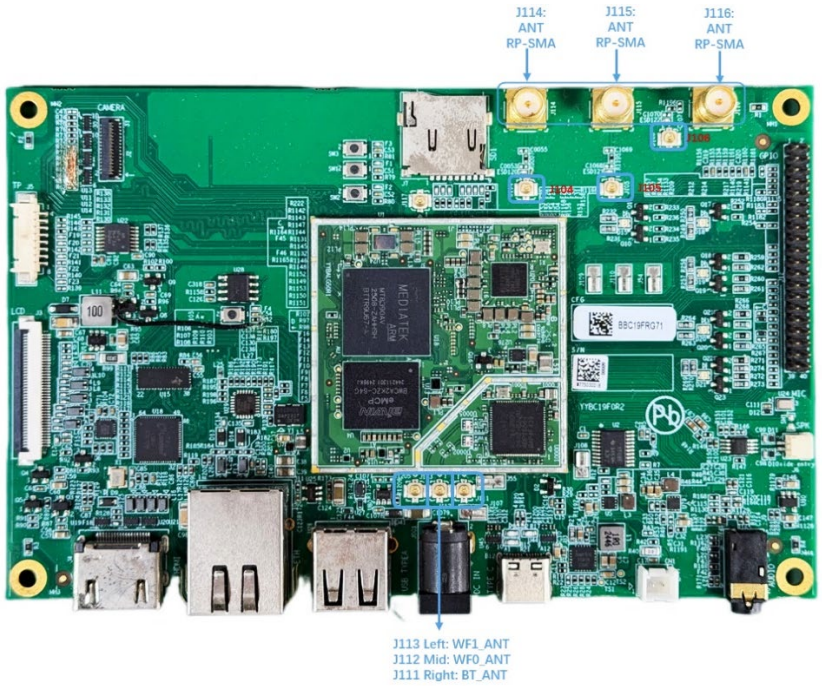
- Location: J11
- Link aggregation not supported
- ESD Immunity: 8kV for contact discharge and 15kV for air discharge
- Surge immunity: 2kV

2.2.8 Wi-Fi and Bluetooth

The VT-SBC-VOSM700-EVB offers a combo Wi-Fi and Bluetooth module, combining Wi-Fi 802.11 a/b/g/n/ac and Bluetooth 5.2.

The WLAN/BT antenna signals from the core board are routed to three U.FL-R-SMT connectors (J111 for BT, J112 & J113 for Wi-Fi) on the carrier board.

They are then connected to J104, J105, J106 via coaxial cables, and further connected to rod antennas through connectors J114, J115, J116.



2.2.9 J4 HDMI

The VT-SBC-VOSM700-EVB supports single display output via either an HDMI or MIPI DSI. Both interfaces are sourced from the MIPI DSI signals on the VOSM700 system-on-module.

Therefore, the HDMI and MIPI DSI interfaces **cannot** be used simultaneously.

Specifications:

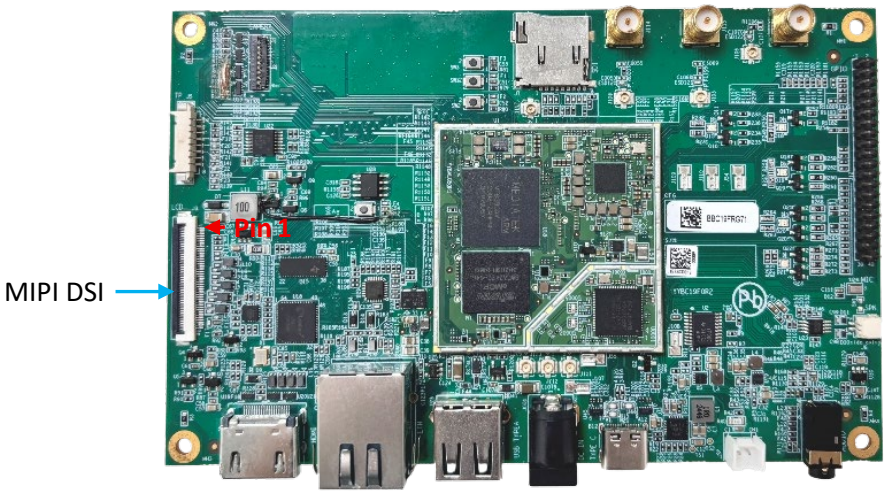
- Location: J4 (HDMI 2.0 Type-A connector)
- Up to 1080p@120Hz or 4096 x 2304@60Hz
- Support HDCP1.4/2.2, CEC, 3D
- Support audio output (non-ARC)
- Support RGB/YUV color formats (Up to 10 bits)

2.2.10 J3 MIPI DSI

The VT-SBC-VOSM700-EVB offers a MIPI DSI interface for up to 1920 x 1080 @60Hz image output. It is designed **NOT** to be used simultaneously with the HDMI interface as both interfaces are sourced from the same MIPI DSI signals on the VOSM700 SoM.

Specifications:

- Location: J3 (1 × 40, 0.5mm pitch)
- Up to 1920 × 1080@60Hz
- Support RGB color format (Up to 8 bits)
- Maximum speed: 2.5Gbps/lane



Pinout description:

Pin	Name	Type	Level	Description
1	LED+	P		LED anode
2	LED+	P		LED anode
3	NC			No connection
4	NC			No connection
5	NC			No connection
6	NC			No connection
7	NC			No connection
8	NC			No connection
9	LED-	P		LED cathode
10	LED-	P		LED cathode
11	GND	P	0V	Ground
12	NC			No connection
13	NC			No connection
14	NC	P		No connection
15	NC			No connection
16	GND	P	0V	Ground
17	NC			No connection
18	NC			No connection
19	GND	P	0V	Ground
20	MIPI_D3P	O		MIPI DSI differential lane 3 +
21	MIPI_D3N	O		MIPI DSI differential lane 3 -
22	GND	P	0V	Ground
23	MIPI_D2P	O		MIPI DSI differential lane 2 +
24	MIPI_D2N	O		MIPI DSI differential lane 2 -
25	GND	P	0V	Ground
26	MIPI_CLK+	O		MIPI DSI differential clock lane +
27	MIPI_CLK-	O		MIPI DSI differential clock lane -
28	GND	P	0V	Ground
29	MIPI_D1P	O		MIPI DSI differential lane 1 +
30	MIPI_D1N	O		MIPI DSI differential lane 1 -

Pin	Name	Type	Level	Description
31	GND	P	0V	Ground
32	MIPI_D0P	O		MIPI DSI differential lane 0 +
33	MIPI_D0N	O		MIPI DSI differential lane 0 -
34	GND	P	0V	Ground
35	NC			No connection
36	RST_LCD	O	3.3V	MIPI LCD reset control, active low
37	GND	P	0V	Ground
38	VCC_LCD	P	3.3V	3.3V power supply
39	VCC_LCD	P	3.3V	3.3V power supply
40	NC			No connection

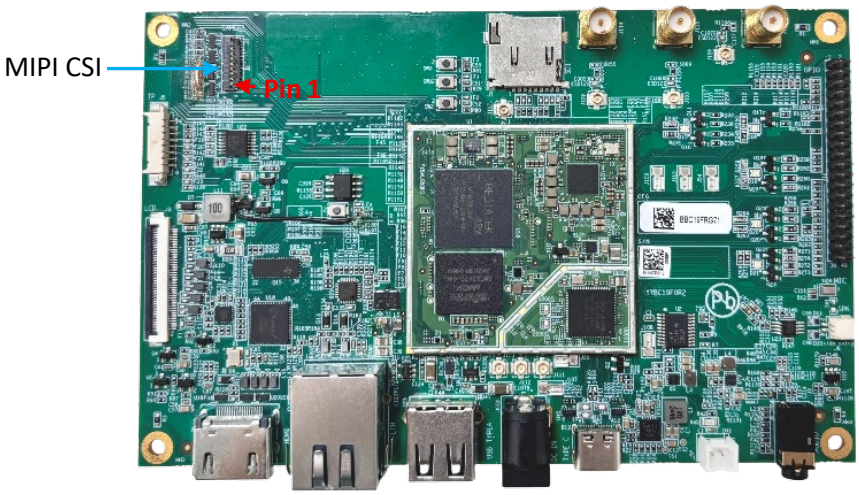
2.2.11 J2 MIPI CSI

VT-SBC-VOSM700-EVB offers a 4-lane MIPI CSI interface for connecting a camera.

Specifications:

- Location: J2 (1 × 31, 0.3mm pitch)
- Support 4 lanes
- Maximum speed: 2.5Gbps/lane
- Support RAW/YCBCR/RGB color formats

The board now supports the SONY IMX214 13MP CMOS image sensor with resolutions up to 4224 × 3136@30fps.



Pinout description:

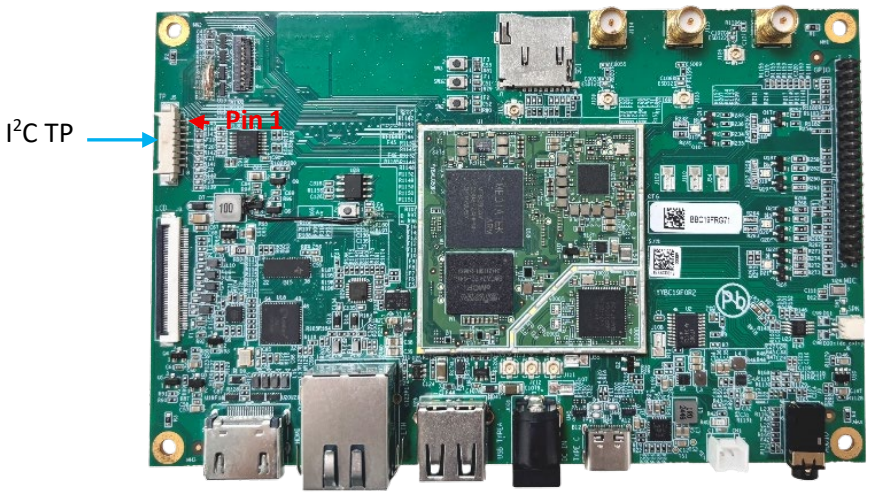
Pin	Name	Type	Level	Description
1	GND	P	0V	Ground
2	MIPI_CSI_D3N	I		MIPI CSI differential lane 3 -
3	MIPI_CSI_D3P	I		MIPI CSI differential lane 3 +
4	GND	P	0V	Ground
5	MIPI_CSI_D2N	I		MIPI CSI differential lane 2 -
6	MIPI_CSI_D2P	I		MIPI CSI differential lane 2 +
7	GND	P	0V	Ground
8	MIPI_CSI_D1N	I		MIPI CSI differential lane 1 -
9	MIPI_CSI_D1P	I		MIPI CSI differential lane 1 +
10	GND	P	0V	Ground
11	MIPI_CSI_D0N	I		MIPI CSI differential lane 0 -
12	MIPI_CSI_D0P	I		MIPI CSI differential lane 0 +
13	GND	P	0V	Ground
14	MIPI_CSI_CLKN	I		MIPI CSI clock -
15	MIPI_CSI_CLKP	I		MIPI CSI clock +
16	GND	P	0V	Ground
17	I2C_SCL_CAM	O	1.8V	Camera serial clock
18	I2C_SDA_CAM	I/O	1.8V	Camera serial data
19	RESET#_CAM	O	1.8V	Camera reset
20	PWN#_CAM	O	1.8V	Camera power down
21	GND	P	0V	Ground
22	MCLK_CAM	O	1.8V	Camera main clock
23	GND	P	0V	Ground
24	NC			No connection
25	VCC_1.8V_CAM	P	1.8V	1.8V Power supply
26	VCC_1.8V_CAM	P	1.8V	1.8V Power supply
27	VCC_1.2V_CAM	P	1.2V	1.5V Power supply
28	VCC_2.8V_CAM	P	2.8V	2.8V Power supply
29	VCC_2.8V_CAM	P	2.8V	2.8V Power supply
30	NC			No connection
31	GND	P	0V	Ground

2.2.12 J5 I²C TP

The I²C touch panel header is for connecting a touch panel.

Specifications:

- Location: J5 (1 × 8, 1.25mm pitch)
- Voltage: 3.3V level
- 400KHz clock
- Support 3.3V⎓0.5A output



Pinout description:

Pin	Name	Type	Level	Description
1	GND	P	0V	Ground
2	GND	P	0V	Ground
3	NC			No connection
4	VCC_3V3	P	3.3V	3.3V power supply
5	RSTB_TP	O	3.3V	TP reset
6	EINT_TP	I	3.3V	TP interrupt
7	SDA0_TP	I/O	3.3V	I²C serial data
8	SCL0_TP	O	3.3V	I²C serial clock

2.2.13 J9 Audio Jack

The VOS700 module offers line out, SPDIF, and I²S audio output signals.

The evaluation board is equipped with a 3.5mm combo audio jack, compatible with four-section headphone output and microphone input. The audio jack supports the CTIA standard.

Specifications:

- Location: J9
- Support microphone input and headphone output
- Support inline control, dual-channel audio

2.2.14 J6 Speaker

The evaluation board offers a 2-pin speaker header for mono audio output.

Specifications:

- Location: J6 (1 × 2, 1.0mm pitch)
- Impedance: Default 4Ω, optional 8Ω
- Maximum power: 1W

Pinout description:

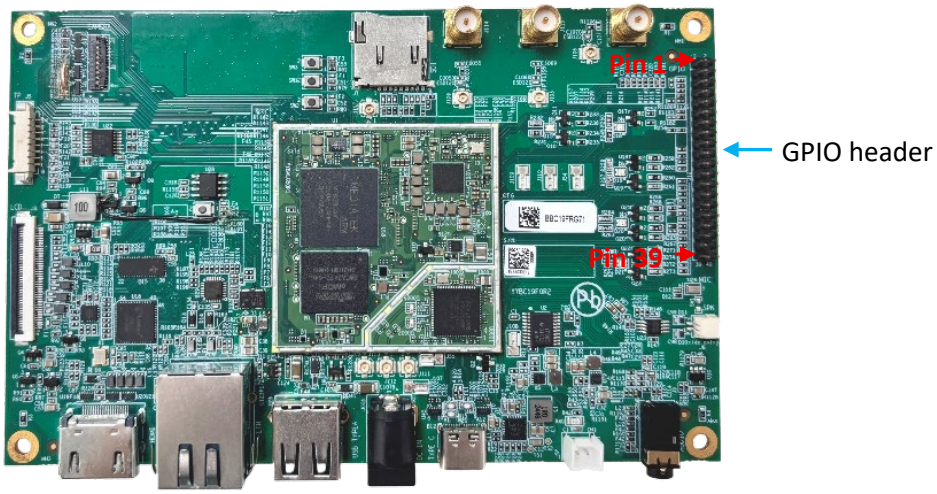
Pin	Name	Type	Level	Description
1	LN	O		BTL negative of left channel
2	LP	O		BTL positive of left channel

2.2.15 U24 Mic

The evaluation board offers an analog microphone for mono-channel sound capturing.

2.2.16 J12 GPIO Header

The evaluation board is equipped with a 2 × 20 GPIO header at a 2.0 mm pitch. It provides 17 native GPIO pins, several of which support multiplexing for alternative functions. The header also includes one debug UART (1.8 V) and two communication UARTs (1.8 V).



Pinout description:

Pin	Name	Type	Level	Description
1	VCC_5V0	P	5V	5V⎓0.5A power output
2	GND	P	0V	Ground
3	VCC_3V3	P	3.3V	3.3V⎓0.5A power output
4	GND	P	0V	Ground
5	VCC_1V8	P	1.8V	1.8V⎓0.5A power output
6	GND	P	0V	Ground
7	UTXD1	O	1.8V	UART 1 transmit data Baud rate: 921600
8	GPIO/PWM_C	I/O	1.8V	Default: GPIO Muxed: PWM output for LED 1 control (max. frequency: 39MHz)

Pin	Name	Type	Level	Description
9	URXD1	I/O	1.8V	UART 1 receive data Baud rate: 921600
10	GPIO_B_2	I/O	1.8V	Default: GPIO Muxed: GPIO output for LED 1 control
11	GND	P	0V	Ground
12	GPIO_B_3	I/O	1.8V	GPIO
13	UTXD2	O	1.8V	UART 2 transmit data Baud rate: 921600
14	NC			No connection
15	URXD2	I	1.8V	UART 2 receive data Baud rate: 921600
16	NC	-	-	No connection
17	GND	P	0V	Ground
18	NC	-	-	No connection
19	SCL2	O	1.8V	I^2C serial clock (frequency: 400KHz)
20	NC	-	-	No connection
21	SDA2	I/O	1.8V	I^2C serial data (frequency: 400KHz)
22	GPIO_D_0	I/O	1.8V	Default: GPIO Muxed: GPIO output for LED 2 control
23	NC	-	-	No connection
24	GPIO_D_1	I/O	1.8V	Default: GPIO Muxed: GPIO output for LED 3 control
25	GPIO/SPDIF_IN	I	1.8V	Default: GPIO Muxed: SPDIF input
26	GPIO_D_2	I/O	1.8V	Default: GPIO Muxed: GPIO output for LED 2 control
27	GND	P	0V	Ground

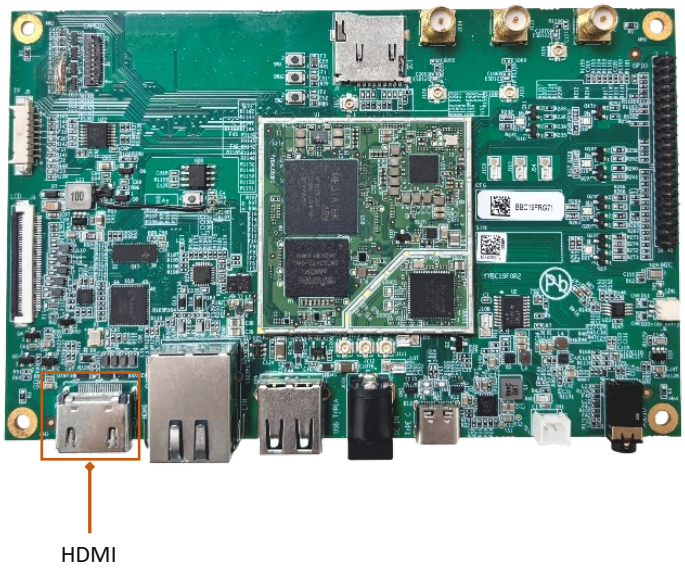
Pin	Name	Type	Level	Description
28	GPIO_D_3	I/O	1.8V	Default: GPIO Muxed: GPIO output for LED 4 control
29	GPIO/SPDIF_OUT	O	1.8V	Default: GPIO Muxed: SPDIF output
30	GPIO_D_4	I/O	1.8V	Default: GPIO Muxed: GPIO output for LED 4 control
31	UART0_RX_Debug	I	1.8V	Debug UART receive data Baud rate: 921600
32	GPIO_D_5	I/O	1.8V	Default: GPIO Muxed: GPIO output for LED 5 control
33	UART0_TX_Debug	O	1.8V	Debug UART transmit data Baud rate: 921600
34	GPIO_D_6	I/O	1.8V	Default: GPIO Muxed: GPIO output for LED 5 control
35	GPIO/JTRST	O	1.8V	Default: GPIO Muxed: JTAG reset
36	GPIO/JTCK	O	1.8V	Default: GPIO Muxed: JTAG sync clock
37	GPIO/JTDO	O	1.8V	Default: GPIO Muxed: JTAG data output
38	GPIO/JTMS	I	1.8V	Default: GPIO Muxed: JTAG test mode select
39	GND	P	0V	Ground
40	GPIO/JTDI	I	1.8V	Default: GPIO Muxed: JTAG data input

CHAPTER 3 YOCTO SYSTEM MANUAL

3.1 Getting Started

To access the system, follow the steps below.

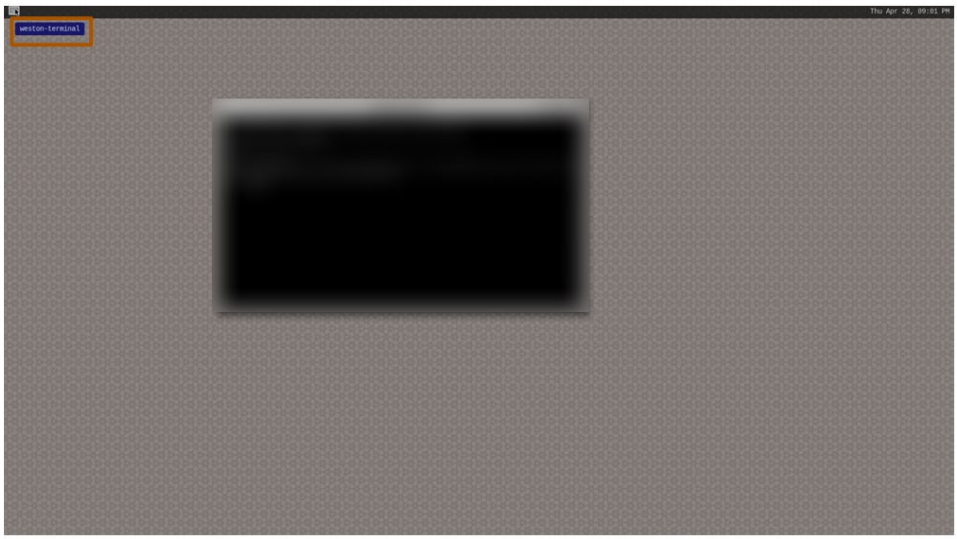
1. Connect the VT-SBC-VOSM700-EVB to a display via the HDMI interface.



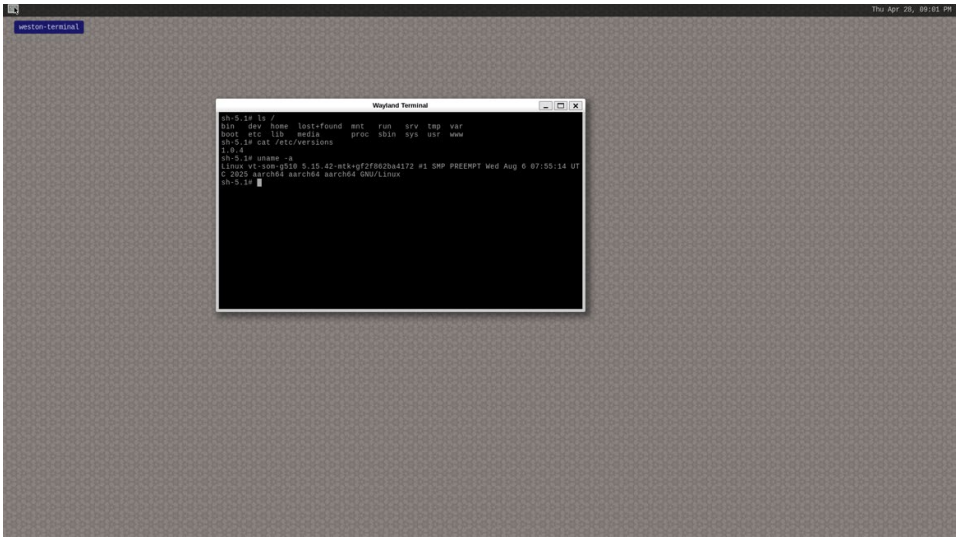
The evaluation board provides one HDMI interface and one MIPI DSI interface. Both derive from the MIPI DSI signals on the VOSM700 system-on-module. The active display output interface is configured before shipment. Unless otherwise required, the HDMI interface is set to default.

2. Power on the board using a 5V⎓3A power adapter and wait for the system to boot into the Weston desktop.
3. Connect a USB mouse and keyboard to the board. (A USB hub may be needed, as only one USB Type-A port is available on the board.)

4. Click the terminal icon in the upper-left corner.



5. A command-line terminal will open, providing access to the system.



*The user logs in as **root** with no password required.*

3.2 User Information

The system logs in automatically as root without a password after power-up.

User Type	User Name	Password
Super User	root	None

3.3 Hardware Functionality Test

Once the VT-SBC-VOSM700-EVB is powered on and the system has booted, follow the test procedures below to verify hardware functionality.

3.3.1 Device Information

- CPU

Run the following command to view CPU architecture, number of cores, and cache information:

```
$ lscpu
```

- Memory

Run the following command to view memory capacity and current usage:

```
$ cat /proc/meminfo
```

- Storage Space

Run the following command to view disk utilization and available space for all mounted file systems:

```
$ df -h
```

- Kernel Version

Run the following command to view the kernel version, build timestamp, CPU architecture, and other system details:

```
$ uname -a
```

- System and Release Versions

Run the following commands to view the system version and release version:

```
$ cat /etc/os-release
$ cat /etc/versions
```

- Root File System Type

Run the following command to identify the root filesystem type:

```
$ cat /etc/issue
```

- EEPROM

1. Run the following command to read the evaluation board's serial number:

```
$ vtvdm -r sn
```

2. Run the following command to the evaluation board MAC address:

```
$ vtvdm -r mac0
```

3.3.2 Display

The evaluation board provides one HDMI interface and one MIPI DSI interface. Both interfaces are sourced from the MIPI DSI signals on the VOSM700 system-on-module. Therefore, simultaneous dual-display operation is **NOT** supported.

The active display output interface is configured when flashing the system image to the board. By default, the VT-SBC-VOSM700-EVB uses HDMI for display output.

- Video Playback

1. Connect a display to either the HDMI or MIPI DSI interface. The system will automatically detect the connection and attempt to output a signal.
2. Prepare an MP4 format video.
3. Modify the location and file name parameters to specify the playback file using the following command:

```
$ gst-launch-1.0 filesrc location=/home/root/test.mp4 ! qtdemux ! avdec_h264 ! videoconvert ! waylandsink
```

- Backlight Control for the MIPI Display

Use the following command to adjust the backlight brightness of the MIPI display.

```
$ echo 50 > /sys/class/backlight/backlight-lcd0/brightness
```

Execute the following command in the current directory to view the maximum supported brightness value.

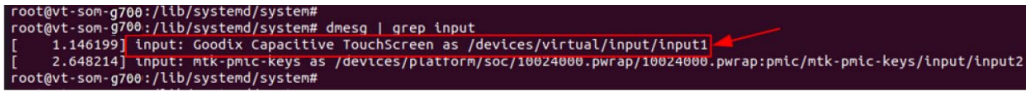
```
$ cat /sys/class/backlight/backlight-lcd0/max_brightness
```

- Touch Panel for the MIPI Display

When using the MIPI DSI interface for display, a touchscreen can be attached.

1. Display the kernel log to verify the input device driver status. Confirm the touch panel is detected and listed with a name like “xxx TouchScreen”.

```
# dmesg | grep input
```



2. Configure the tslib environment variables to set up the touch panel runtime environment.

```
$ export TSLIB_CONSOLEDEVICE=none
$ export TSLIB_FBDEVICE=/dev/fb0
$ export TSLIB_TSDEVICE=/dev/input/event1        // matches the detected node
$ export TSLIB_CONFFILE=/etc/ts.conf
$ export TSLIB_PLUGINDIR=/usr/lib/ts
$ export LD_PRELOAD=/usr/lib/libts.so.0
$ export TSLIB_CALIBFILE=/etc/pointercal
```

3. Stop the graphics service to release the framebuffer and input devices for testing.

```
$ systemctl stop weston.socket
$ systemctl stop weston.service
```

4. Start the touch panel test program.

```
$ ts_test
```

5. Perform manual touch operations to verify that the touch function works correctly.

3.3.3 Audio

The VT-SBC-VOSM700-EVB provides audio interfaces including a 3.5mm combo audio jack, speaker output, and analog microphone (A-mic) input.

- Headphone Playback

1. Set the DAC input multiplexer to the normal audio path.

```
$ amixer -c mt8390evk cset name='DAC In Mux' Normal Path
```

2. Route audio playback to the headphone (HP) output.

```
$ amixer -c mt8390evk cset name='HP Mux' 'Audio Playback'
```

3. Disable the line-out left (LOL) path to isolate headphone audio.

```
$ amixer -c mt8390evk cset name='LOL Mux' 'Open'
```

4. Set the headphone volume (range: 0-30, loud to mute).

```
$ amixer -c mt8390evk cset name='Headset Volume' 4
```

Level mapping: 0 -> +8 dB, 1 -> +7 dB, 18 -> −10 dB, 31 -> −40 dB (mute).

5. Verify headphone audio output by playing a test audio file (e.g., test.wav).

```
$ aplay test.wav
```

- Microphone Recording

1. Route analog input 1 (AIN1) to left PGA (Programmable Gain Amplifier).

```
$ amixer -c mt8390evk cset name='PGA_L_Mux' 'AIN1'
```

2. Adjust the PGA1 gain/volume for optimal jack mic sensitivity (range: 0-4, quite to loud).

```
$ amixer -c mt8390evk cset name='PGA1 Volume' 3
```

Level mapping: 0->0dB， 1->6dB， 2->12dB， 3->18dB， 4->24dB.

3. Record 10 seconds of audio from the jack mic (48kHz, 32-bit little-endian).

```
$ arecord -D jack_mic -r 48000 -f S32_LE -d 10 sample.wav
```

4. Play back the recorded file to confirm audio clarity.

```
$ aplay sample.wav
```

- Speaker Playback

1. Disable the headphone (HP) path to isolate audio output to the speaker.

```
$ amixer -c mt8390evk cset name='HP Mux' 'Open'
```

2. Route the left DAC playback to line-out left (LOL) (speaker output path).

```
$ amixer -c mt8390evk cset name='LOL Mux' 'Playback_L_DAC'
```

3. Set the speaker/line-out volume (range: 0-30, loud to mute).

```
$ amixer -c mt8390evk cset name='Lineout Volume' 4
```

Level mapping: 0 -> +8 dB, 1 -> +7 dB, 18 -> −10 dB, 31 -> −40 dB (mute).

4. Play a test audio file (e.g., Chains.wav) through the speaker.

```
$ aplay -D jack_speaker Chains.wav
```

- A-Mic Recording

1. Route the analog input 0 (AIN0) to the left PGA (A-mic input).

```
$ amixer -c mt8390evk cset name='PGA_L_Mux' 'AIN0'
```

2. Set the PGA1 gain/volume for optimal A-mic sensitivity (range: 0-4, quite to loud).

```
$ amixer -c mt8390evk cset name='PGA1 Volume' 3
```

Level mapping: 0->0dB， 1->6dB， 2->12dB， 3->18dB， 4->24dB.

3. Record 10 seconds of audio from the A-mic (48kHz, 32-bit little-endian)

```
$ arecord -D amic -r 48000 -f S32_LE -d 10 record.wav
```

4. Play back the recorded file to confirm audio clarity.

```
$ aplay record.wav
```

3.3.4 Camera

1. Automatically identify the MediaTek camera device node and store in the 'VIDEO_DEV' array.

```
$ declare -a VIDEO_DEV=(`v4l2-ctl --list-devices | grep mtk-v4l2-camera -A 3 | grep video | tr -d "\n"`)
```

2. Stream the camera feed to the Weston Wayland display for real-time preview (720p resolution).

```
$ gst-launch-1.0 v4l2src device=${VIDEO_DEV[1]} ! v4l2convert output-io-mode=dmabuf-import \
! video/x-raw,width=1280,height=720 ! waylandsink sync=false
```

3. Capture a 4K (3840×2160) JPEG still image and save it to a file.

```
$ v4l2-ctl -d ${VIDEO_DEV[2]} --set-fmt-video=width=3840,height=2160,pixelformat=JPEG,sizeimage=4194304 --stream-mmap=1 --stream-count=1 --stream-to=capture.jpg --verbose
```

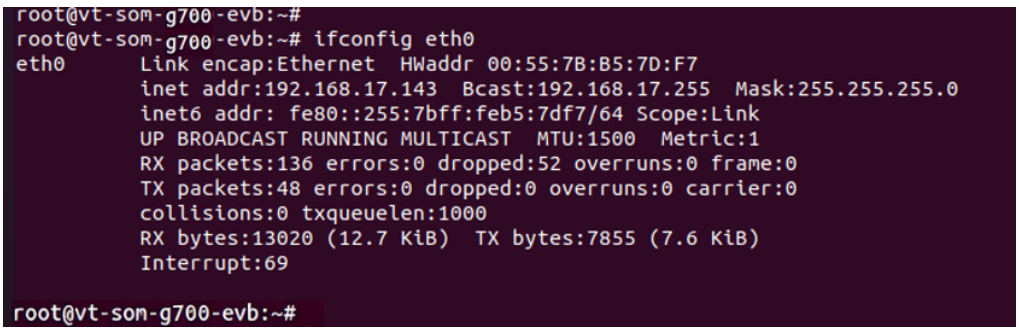
3.3.5 Ethernet

The board provides a Gigabit Ethernet port operating in WAN mode by default. Its IP configuration is set to use DHCP.

The Ethernet interface is mapped as **eth0** in the filesystem by default.

1. Use the ifconfig command to check the network interface information.

```
$ ifconfig eth0
```



2. Use the ping command to verify network connectivity:

```
$ ping 192.168.17.218
```

If replies are received and no packet loss occurs, the connection is working properly.

3.3.6 Wi-Fi Client

The Wi-Fi interface is mapped as **wlan0** in the filesystem by default.

1. Scan for available Wi-Fi APs.

```
# iw wlan0 scan | grep -Ei ssid
SSID: G405-B710
SSID: Vantron-6F
SSID: vantron_test_321
……
```

2. Create a configuration entry for connecting to the target Wi-Fi network. For example, for SSID "M19" with password "12345678".

```
$ wpa_passphrase M19 12345678 >> /etc/wpa_supplicant/wpa_supplicant-wlan0.conf
```

3. Reload the Wi-Fi daemon so it connects to M19 with the new saved credentials.

```
$ systemctl restart wpa_supplicant@wlan0.service
```

4. Request an IP address lease for the wlan0 interface using DHCP.

```
$ udhcpc -i wlan0
```

5. Verify network connectivity.

```
$ ping -I wlan0 www.google.com
```

If the domain name cannot be resolved, configure DNS resolution first.

```
$ rm /etc/resolv.conf
$ echo "nameserver 8.8.8.8" >> /etc/resolv.conf
$ sync
```

After completing the configuration, execute the ping command again.

3.3.7 Bluetooth

The board supports pairing with Bluetooth devices across various profiles, including HID (keyboards, mice), HOGP (BLE HID devices), A2DP (Bluetooth speakers), and GATT-based devices (heart rate monitors, sensors, smartphones).

- HID and HOGP

1. Launch **bluetoothctl** to enter the Bluetooth interactive shell, then power on the Bluetooth controller and start scanning available Bluetooth device:

```
# hciconfig -a
hci0:   Type: Primary  Bus: SDIO
        BD Address: E2:0F:54:1D:FA:F0  ACL MTU: 1021:6  SCO MTU: 244:4
        DOWN
        RX bytes:14452 acl:0 sco:0 events:1167 errors:0
        TX bytes:293782 acl:0 sco:0 commands:1167 errors:0
        Features: 0xbf 0x3e 0x8d 0xfe 0xdb 0xff 0x7b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH SNIFF
        Link mode: PERIPHERAL ACCEPT
# bluetoothctl
Agent registered
[CHG] Controller E2:0F:54:1D:FA:F0 Pairable: yes
[bluetooth]# power on
[CHG] Controller E2:0F:54:1D:FA:F0 Class: 0x00000000
Changing power on succeeded
[CHG] Controller E2:0F:54:1D:FA:F0 Powered: yes
[bluetooth]# scan on
Discovery started
[CHG] Controller E2:0F:54:1D:FA:F0 Discovering: yes
[NEW] Device 33:03:F7:99:2E:B4 ble_master
[NEW] Device 4B:7B:0E:D2:9E:CB NZpN45pzUQ8TKcvRFKkxTgWY0
[NEW] Device 5C:C5:63:C7:3D:05 5C-C5-63-C7-3D-05
[NEW] Device 40:DC:69:79:01:11 40-DC-69-79-01-11
[NEW] Device 5C:C5:63:C9:35:D7 5C-C5-63-C9-35-D7
```

2. Set the target HID/HOGP device to the pairing mode so that it can be discovered by the board.
3. Run the following commands to pair the board with the HID/HOGP device.

```
[bluetooth]# pair xx:xx:xx:xx:xx                // address of the HID/HOGP device
[bluetooth]# connect xx:xx:xx:xx:xx             // address of the HID/HOGP device
[bluetooth]# exit
```

4. After pairing with the HID/HOGP device, exit bluetoothctl.
5. A corresponding eventX device will be created under /dev/input. Use evtest to quickly verify HID functionality:

```
$ cd /dev/input
$ evtest eventX
```

6. Disconnect the HID/HOGP device and remove it from the pairing list.

```
[bluetooth]# disconnect xx:xx:xx:xx:xx          // address of the HID/HOGP device
[bluetooth]# remove xx:xx:xx:xx:xx              // address of the HID/HOGP device
```

- GATT

1. Ensure that the **nRF Connect** application is installed on the target GATT device.
2. On the target GATT device:
   - Launch the **nRFConnect** application
   - Select "SCANNER"
   - Tap on "scan"
3. Launch **bluetoothctl** on the evaluation board to enter the Bluetooth interactive shell, then power on the Bluetooth controller and start scanning available Bluetooth device. (The commands are the same as those used in the **HID and HOGP** section).
4. Enable advertising on the board so that the device's Bluetooth can be discovered by the **nRF Connect** application during scanning.

```
[bluetooth]# menu advertise                 // Enter the advertise menu
[bluetooth]# name bluetoothAdv              // Set the advertisement name for the board
[bluetooth]# back                           // Return to the main menu
[bluetooth]# advertise peripheral           // Enable advertising
```

5. The evaluation board with the name “bluetoothAdv” will be discovered by the GATT device.

- Bluetooth Speaker

Audio playback via Bluetooth speakers relies on PulseAudio, pactl, and paplay. These components require a **non-root user** to run properly. Since the system logs in as the root user by default without a password, you must create a dedicated non-root user to use this functionality.

1. Create a new user and switch to it:

```
$ useradd test                    // Name the user as “test”
$ chown -R test /run/user/0/      // Change the ownership of the runtime directory
$ su test                         // Switch the current session from root to “test”
```

2. Start the **PulseAudio** daemon for audio processing:

```
$ pulseaudio -D --exit-idle-time -1
```

3. Launch **bluetoothctl** to enter the Bluetooth interactive shell, then power on the Bluetooth controller and start scanning available Bluetooth device. (The commands are the same as those used in the **HID and HOGP** section).

4. Set the target Bluetooth speaker to the pairing mode so that it can be discovered by the board.

5. Run the following commands to pair the board with the Bluetooth speaker.

```
[bluetooth]# pair xx:xx:xx:xx:xx          // address of the Bluetooth speaker
[bluetooth]# connect xx:xx:xx:xx:xx       // address of the Bluetooth speaker
[bluetooth]# exit
```

6. List all available audio sinks with pactl.

```
$ pactl list sinks
```

7. Locate the A2DP sink for the Bluetooth device:

```
Name: bluez_sink.XXX.a2dp_sink
```

8. Play audio to the Bluetooth speaker by specifying the A2DP sink with paplay.

```
$ paplay /usr/share/sounds/alsa/Front_Right.wav -d <sink Name>
```

3.3.8 USB

- USB Type-A

The USB 2.0 Type-A interface supports USB storage devices, USB mice, USB keyboards, USB hubs, and other standard USB peripheral devices.

After inserting a USB flash drive into the USB 2.0 Type-A interface, use the following commands to view its information or manage the device.

1. Display the details of all storage devices and their partitions:

```
$ fdisk -l
$ blkid
```

2. The system automatically mounts USB drive partitions after insertion. Use the command to check the space usage of the file system disk.

```
$ df -h
```

3. Use the following dd command to test actual write speed of the USB drive.

```
$ sudo dd if=/dev/zero of=/dev/sda bs=1M count=100 oflag=sync
```

This measures the time required to write 100MB of data.

4. Use the following dd command to test actual read speed of the USB drive.

```
$ sudo dd if=/dev/sda of=/dev/null bs=1M count=100 iflag=direct
```

This measures the time required to read 100MB of data.

The oflag/iflag parameters specify I/O operation methods and may be removed during execution.

- USB Type-C

  The USB Type-C interface supports OTG for ADB functionality. It also supports the USB Ethernet adapter mode, with a fixed IP address: 192.168.96.1.

  Refer to Section **3.5.2** for the specific steps to log in to the board via ADB.

3.3.9 UART

The evaluation board offers two UART communication ports on the GPIO header, identified as /dev/ttyS1 (pin 7, pin 9) and /dev/ttyS2 (pin 13, pin 15) in the filesystem. Both ports use a 1.8 V logic level.

Refer to **Section** 2.2.16 for the GPIO header pinout and interface voltage level specifications.

For quick functional verification, you can perform a loopback test by shorting the transmit (TX) and receive (RX) pins of the UART port, then open a serial communication tool (e.g., microcom) to verify and view data transmission between the pins.

Test Procedures:

1. Use jumper wires to short the data transmission and reception pins of an UART.

   Wiring configuration: TX ↔ RX

2. Open a terminal and use **microcom** to configure and test data transmission over the interface (UART1 for instance):

   ```
   $ busybox microcom -t 5000 -s 115200 /dev/ttyS1          // Baud rate: 115200 baud
   ```

3. If keyboard input appears directly below the running command, the loopback test is successful.

   The -t parameter sets a 5-second timeout; the command exits automatically if no data transmission or reception occurs.

3.3.10 GPIO

The board provides 17 native GPIO pins, several of which support multiplexing for alternative functions.

Use the echo command to set the level of the GPIO pins.

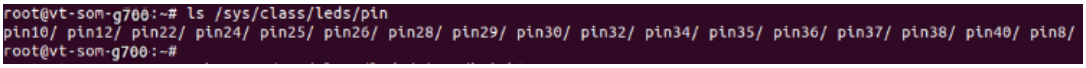
1. Navigate to the sysfs directory for GPIO control and list the configurable files.

```
# ls /sys/class/leds/pins
```

2. Set the output level of a specific pin to high or low (default low):

```
# echo 0 > /sys/class/leds/pin10/brightness          // set pin 10 to high
# echo 1 > /sys/class/leds/pin10/brightness          // set pin 10 to low
```

Controllable pins are listed in the output of the ls command, with the pin names matching the physical pins on the header. Refer to Section **2.2.16** for the pinout of the header.



3.3.11 Button

The board supports four normally open push-buttons for power on/off, reset, recovery (download), and user-defined functions as described in Section **2.2.3**.

The SW3 (Home) button is reserved and can be tested/monitored using the **evtest** tool. Its status is exposed via /dev/input/event1 corresponding to the mtk-pmic-keys input device.

1. Launch the **evtest** tool.

```
# evtest
```

2. After executing the command , the system automatically scans all input devices under /dev/input/event* and lists available options.
3. Enter the event number corresponding to the button device—enter **1** for /dev/input/event1 (mtk-pmic-keys, home button controller) — then press **Enter**.
4. The system displays detailed information about the selected button device, confirming supported key events.

5. Press and release the SW3 button; event logs will print in the terminal to confirm detection.

   If no events are printed, the button/hardware/driver is faulty.

6. Press Ctrl + C to stop the evtest tool and exit the test mode.

```
root@vt-som-g700:~# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      mt8370-evk HDMI Jack
/dev/input/event1:      mtk-pmic-keys
Select the device event number [0-1]: 1
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x1
Input device name: "mtk-pmic-keys"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 102 (KEY_HOME)
    Event code 116 (KEY_POWER)
Properties:
Testing ... (interrupt to exit)
Event: time 1774444115.331041, type 1 (EV_KEY), code 102 (KEY_HOME), value 1
Event: time 1774444115.331041, -------------- SYN_REPORT ------------
Event: time 1774444115.526326, type 1 (EV_KEY), code 102 (KEY_HOME), value 0
Event: time 1774444115.526326, -------------- SYN_REPORT ------------
Event: time 1774444116.486593, type 1 (EV_KEY), code 102 (KEY_HOME), value 1
Event: time 1774444116.486593, -------------- SYN_REPORT ------------
Event: time 1774444116.684991, type 1 (EV_KEY), code 102 (KEY_HOME), value 0
Event: time 1774444116.684991, -------------- SYN_REPORT ------------
```

3.4 Image Flashing

The flashing is performed on a host PC connected with the evaluation board via the USB Type-C port.

Make sure you have installed **Genio tools** on your Linux or Windows host PC before proceeding with the process.

Follow the steps below to set up required software, USB device rules and install Genio tools on the Ubuntu host PC.

3.4.1 Host System Requirements

- Linux: Ubuntu 20.04 or later LTS version / Windows system
- Internet connection
- Administrator privilege
- Python 3.9 or later (Python 3.12 recommended)
- Pip3 20.3 or later
- Fastboot 34.0.4 or later

3.4.2 Software Setup and Genio Tools Installation

- Linux Host

1. Python3 and pip

If you don't have Python and pip installed, run the following commands to install them.

```
$ sudo apt update
$ sudo apt-get install python3
$ sudo apt-get install python3-pip
```

Check Python and pip versions:

```
$ python3 --version
Python 3.12.2
$ pip3 --version
pip 23.3.1 from /usr/bin/pip3 (python 3.12)
```

If the pip3 version is under 20.3, please upgrade it by running the following command:

```
$ pip3 install --upgrade pip
```

2. Fastboot

Genio tools use fastboot to flash images, and you can run the following command to install it:

```
$ sudo apt update
$ sudo apt-get install adb fastboot
```

3. USB Device Rules

To enable communication between the host PC and the board via USB without root privileges, create a udev rule to grant user-level access to the device:

```
$ echo -n 'SUBSYSTEM=="usb", ATTR{idVendor}=="0e8d", ATTR{idProduct}=="201c",
 MODE="0660", TAG+="uaccess"
SUBSYSTEM=="usb", ATTR{idVendor}=="0e8d", ATTR{idProduct}=="0003", MODE="0660",
 TAG+="uaccess"
SUBSYSTEM=="usb", ATTR{idVendor}=="0403", MODE="0660", TAG+="uaccess"
SUBSYSTEM=="gpio", MODE="0660", TAG+="uaccess"
' | sudo tee /etc/udev/rules.d/72-aiot.rules
```

```
$ sudo udevadm control --reload-rules
$ sudo udevadm trigger
```

udev is a device manager for the Linux kernel, enabling the allocation of permissions to users or groups to access devices (e.g., via ADB). To interact with the board, please add the new udev rule and your user account to the plugdev group.

```
$ echo 'SUBSYSTEM=="usb", ATTR{idVendor}=="0e8d", ATTR{idProduct}=="201c",
 MODE="0660", GROUP="plugdev"' | sudo tee -a /etc/udev/rules.d/96-rity.rules
$ sudo udevadm control --reload-rules
$ sudo udevadm trigger
$ sudo usermod -a -G plugdev $USER
```

After adding your user account to the group, you must **logout and login again** for the changes to take effect.

4. Genio Tools

After finishing the installation of the required packages and necessary configurations as instructed above, you are now prepared to install Genio Tools:

```
$ pip3 install -U genio-tools
```

After installing Genio Tools, **logout and login again** to ensure that the **PATH** environment variable is properly updated.

Following re-login, you can verify the installation by executing genio-config.

```
$ genio-config
fastboot: OK
udev rules: OK (md5: a3b2767b42ee01d7c62bf394400528ae)
Serial device write access: OK
```

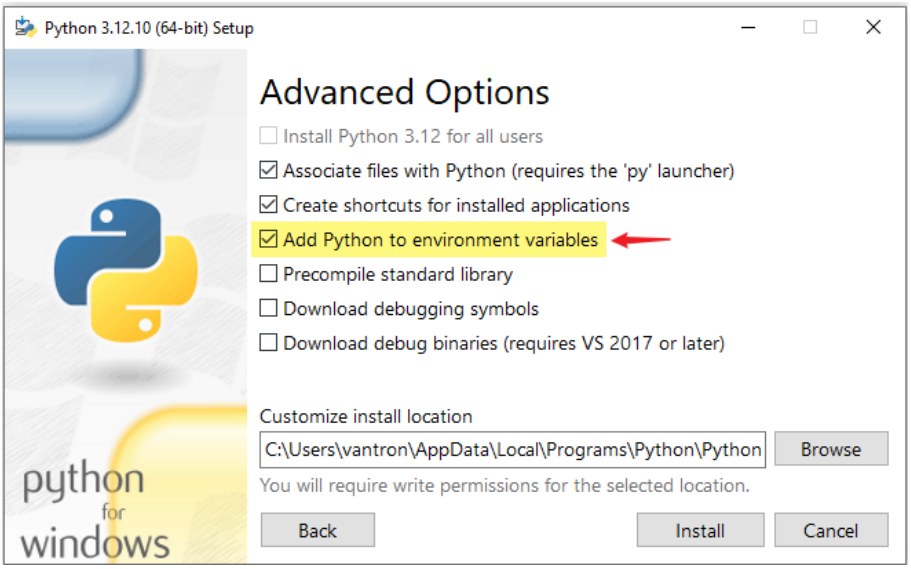
In case your environment is not set up correctly, the tools provide instructions on how to correctly configure it.

- Windows Host

1. Python 3

There are known issues reported for Python 3.10 & 3.14 on Windows platform when installing Genio tools due to missing dependencies. You are recommended to download and install Python 3.12 from https://www.python.org/downloads/windows/.

Make sure you've ticked the **Add Python to environment variables** or **Add Python to PATH** checkbox during installation:



After Python installation, configure the following environment variable in Windows:

1. Press Windows + R to open the Run dialog.
2. Type cmd into the box and then press Ctrl + Shift + Enter to run the command prompt as an administrator.

```
// Set the environment variable temporarily (valid only for the current CMD session)
set REQUESTS_CA_BUNDLE=%LOCALAPPDATA%\.certifi\cacert.pem
// Set the environment variable permanently (takes effect after restarting CMD/PC)
setx REQUESTS_CA_BUNDLE %LOCALAPPDATA%\.certifi\cacert.pem
```

2. Pip3

During Python installation on Windows, verify that the pip3 version is above 20.3:

- Run the following command:

```
> pip3 --version
pip 25.3 from C:\Users\user\AppData\Local\Programs\Python\Python312\Lib\site-packages\pip (python 3.12)
```

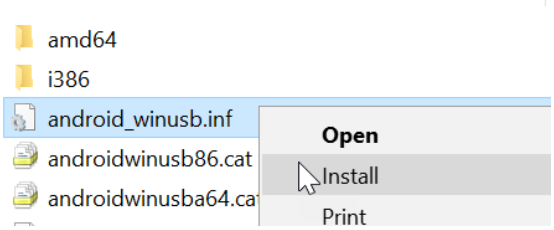
- If the version is not greater than 20.3, upgrade pip3 immediately after installation.

3. Fastboot and ADB

Genio tools require the fastboot device driver to be properly installed on Windows for image flashing, while ADB can be used to connect the board to the host PC. The Google USB Driver contains drivers for both fastboot and ADB on Windows.

To install the drivers:

1. Download Google USB Driver from https://developer.android.com/studio/run/win-usb.
2. Extract the downloaded zip file. The filename is typically in the format of "usb_driver_r13-windows.zip".
3. Locate the **android_winusb.inf** file in the extracted folder.
4. Right click on the **android_winusb.inf** file and select **Install** from the context menu.



5. Confirm the pop-up dialog.

For details, please refer to https://developer.android.com/studio/run/oem-usb#InstallingDriver.

The fastboot device driver is categorized as Android Device > Android Bootloader Interface in Windows.

To install fastboot tool on Windows:

1. Download **SDK Platform-Tools for Windows** from https://developer.android.com/studio/releases/platform-tools.
2. Extract the downloaded zip file.
3. Add the unzipped directory to the **PATH** environment variable.
4. Check if fastboot tool is properly installed by executing the following command.

```
> fastboot --version

fastboot version 35.0.2-12147458
```

If the fastboot version is under 34.0.4, it may cause corrupted storage when flashing images larger than 4GB. Please update to version 34.0.4 or later.

3.4.3 Flashing Procedure

Commands for flashing the image are the same on both Windows host and Linux host.

- Prerequisites

1. Firmware package: 700RBALG5GT71_VT-SOM-G700-EVB_Yocto_Vxx_Image_xxx.zip
2. A host PC with Genio Tools and necessary software installed
3. The VT-SBC-VOSM700-EVB evaluation board and 5V⎓3A power adapter
4. USB Type-A to Type-C cable

- Enter Download Mode and Start Flashing

1. Verify the firmware integrity by calculating the MD5 checksum and comparing it with the provided value.

```
$ md5sum 700RBALG5GT71_VT-SOM-G700-EVB_Yocto_Vxx_Image_xxx.zip

fccb81c5a7e813d38e36f2d3e6894e5d  700RBALG5GT71_VT-SOM-G700-EVB_Yocto_Vxx_Image_xxx.zip

$ cat 700RBALG5GT71_VT-SOM-G700-EVB_Yocto_Vxx_xxx_Image_xxx.zip.md5

fccb81c5a7e813d38e36f2d3e6894e5d  700RBALG5GT71_VT-SOM-G700-EVB_Yocto_Vxx_Image_xxx.zip
```

2. Unzip the provided firmware package.

```
$ unzip 700RBALG5GT71_VT-SOM-G700-EVB_Yocto_Vxx_Image_xxx.zip
```

3. Navigate to the image directory.

```
$ cd 700RBALG5GT71_VT-SOM-G700-EVB_Yocto_Vxx_Image_xxx/SW/Image
```

4. Select and execute the corresponding command based on the display in use.

   For HDMI display:

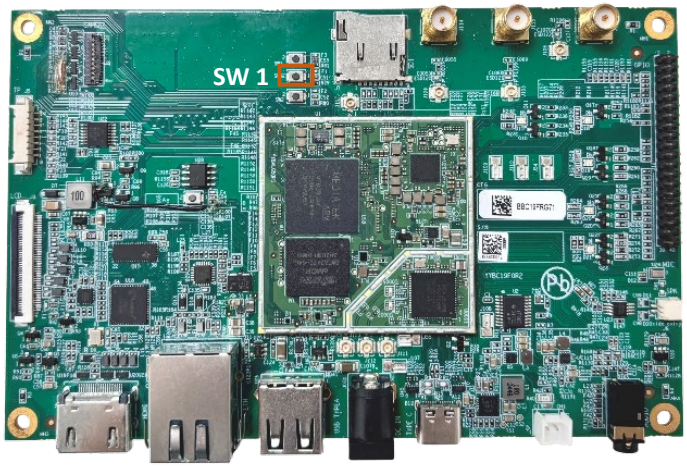
```
$ genio-flash -i vtlinux-image-weston-vtlinux-wayland --load-dtbo gpu-mali.dtbo --load-dtbo apusys.dtbo \
        --load-dtbo video.dtbo --load-dtbo camera-imx214-csi0.dtbo --load-dtbo vt-display-hdmi.dtbo
```

   For MIPI display:

```
$ genio-flash -i vtlinux-image-weston-vtlinux-wayland --load-dtbo gpu-mali.dtbo --load-dtbo apusys.dtbo \
        --load-dtbo video.dtbo --load-dtbo camera-imx214-csi0.dtbo --load-dtbo vt-display-dsi.dtbo
```

By default, genio-flash will flash the full image contents from the designated image directory.

5. Press and hold the **SW1** button.



6. Power on the evaluation board.
7. Wait about 3 seconds, then release the SW1 button.
8. The flashing process will start automatically after SW1 is released.

3.5 Device Access

The VT-SBC-VOSM700-EVB offers two options to connect to a host PC for debugging and development purposes, namely via the debug UART or the USB Type-C.

3.5.1 Via Debug UART

Refer to the following parameters when using the port.

Baud Rate	Data Bit	Polarity	Stop Bit	Device Name
921600	8	None	1	/dev/ttyUSB0

Refer to Section **2.2.5.1 or 2.2.16** for the debug UART pin definitions on the GPIO header.

Wiring:

1. Use a Serial-to-USB adapter and DuPont wires to connect the board's debug UART to a host PC.

   Wiring: TX<-->RX, RX<-->TX, GND<-->GND

2. Power on the development board using a 5V⎓3A power adapter.

- Linux Host

1. Install a serial communication program (such as minicom).

```
$ sudo apt-get update
$ sudo apt-get install minicom
```

2. Enumerate the device node of the debug UART.

```
$ ls /dev/ttyUSB*
/dev/ttyUSB0
```

3. Open the host terminal and enter the following command to open the debug UART using minicom.

```
$ sudo minicom -D /dev/ttyUSB0 -b 921600
```

   - -D: UART device name
   - -b: Baud rate

4. You should see boot log output in the terminal.

- Windows Host

Prerequisites:

- **Serial terminal emulator**

A serial terminal emulator is required to communicate with the UART serial console of the board. Recommended serial terminal emulators for Windows:

PuTTY

Tera Term

MobaXterm

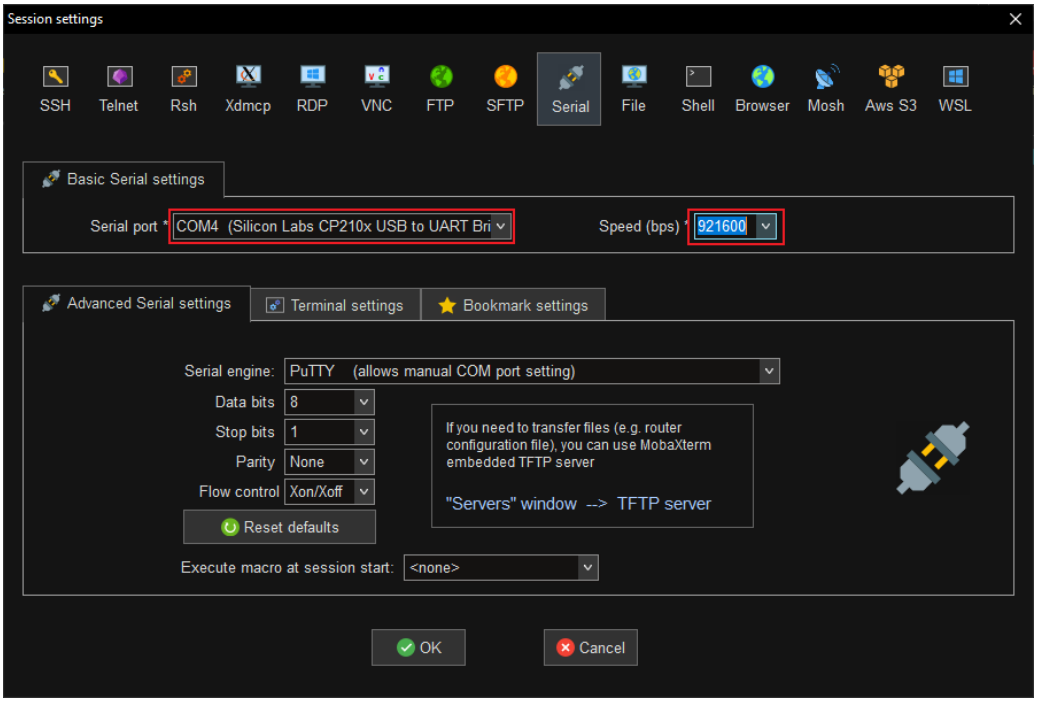
- **USB-to-UART Driver**

To properly identify the UART device on the host PC, install the corresponding driver for your Serial-to-USB adapter.

Debugging Steps:

1. Open **Device Manager** and check the COM port number identified by the host computer.

2. Open a serial communication program (such as MobaXterm), select the identified COM port, and set the baud rate to 921600.



3. Click **OK** to start the session. The boot logs will appear, and the system will automatically log in once the boot process is complete.

3.5.2 Via ADB

Before you start, follow the instructions in Section **3.4.2** to install the fastboot and ADB tools on your Linux/Windows host.

1. Connect the evaluation board to the host PC using a USB Type-A to Type-C cable.
2. Power on the evaluation board using a 5V⎓3A power adapter.
3. Open a PowerShell terminal (on a Windows host) or a Terminal (on a Linux host).
4. Verify if the board is successfully connected to the host.

```
$ adb devices
```

5. Enter `adb shell` followed by `bash` to log into the device.

```
$ adb shell
# bash
```

3.5.3 SSH Login

Prerequisites:

1. The evaluation board is connected to your PC via one of these network options:

   Ethernet, USB Type-C-to-Ethernet, or Wi-Fi.

2. You have obtained the board's IP through the debug UART, ADB, or the Yocto desktop terminal via the following command.

```
$ ifconfig
```

Login Procedure:

1. Install SSH client if it is not available.

   Windows host: Refer to the recommended serial terminal emulators for Windows system in Section **3.5.1**.

   Linux host:

   ```
   $ sudo apt-get update
   $ sudo apt-get install ssh
   ```

2. Enter the following command and select "Yes" when prompted to log into the evaluation board (connection via USB-C to Ethernet).

   ```
   $ ssh root@192.168. 96.1
   ```

CHAPTER 4 DISPOSAL AND PRODUCT WARRANTY

4.1 Disposal

When the device comes to end of life, you are suggested to properly dispose of the device for the sake of the environment and safety.

Before you dispose of the device, please back up your data and erase it from the device.

It is recommended that the device is disassembled prior to disposal in conformity with local regulations. Please ensure that the abandoned batteries are disposed of according to local regulations on waste disposal. Do not throw batteries into fire or put in common waste canister as they are explosive. Products or product packages labeled with the sign of “explosive” should not be disposed of like household waste but delivered to specialized electrical & electronic waste recycling/disposal center.

Proper disposal of this sort of waste helps avoid harm and adverse effect upon surroundings and people’s health. Please contact local organizations or recycling/disposal center for more recycling/disposal methods of related products.

4.2 Warranty

Product warranty

VANTRON warrants to its CUSTOMER that the Product manufactured by VANTRON, or its subcontractors will conform strictly to the mutually agreed specifications and be free from defects in workmanship and materials (except that which is furnished by the CUSTOMER) upon shipment from VANTRON. VANTRON's obligation under this warranty is limited to replacing or repairing at its option of the Product which shall, within **24 months** after shipment, effective from invoice date, be returned to VANTRON's factory with transportation fee paid by the CUSTOMER and which shall, after examination, be disclosed to VANTRON's reasonable satisfaction to be thus defective. VANTRON shall bear the transportation fee for the shipment of the Product to the CUSTOMER.

Out-of-Warranty Repair

VANTRON will furnish the repair services for the Product which are out-of-warranty at VANTRON's then-prevailing rates for such services. At customer's request, VANTRON will provide components to the CUSTOMER for non-warranty repair. VANTRON will provide this service as long as the components are available in the market; and the CUSTOMER is requested to place a purchase order up front. Parts repaired will have an extended warranty of 3 months.

Returned Products

Any Product found to be defective and covered under warranty pursuant to Clause above, shall be returned to VANTRON only upon the CUSTOMER's receipt of and with reference to a VANTRON supplied Returned Materials Authorization (RMA) number. VANTRON shall supply a RMA, when required within three (3) working days of request by the CUSTOMER. VANTRON shall submit a new invoice to the CUSTOMER upon shipping of the returned products to the CUSTOMER. Prior to the return of any products by the CUSTOMER due to rejection or warranty defect, the CUSTOMER shall afford VANTRON the opportunity to inspect such products at the CUSTOMER's location and no Product so inspected shall be returned to VANTRON unless the cause for the rejection or defect is determined to be the responsibility of VANTRON. VANTRON shall in turn provide the CUSTOMER turnaround shipment on defective Product within **fourteen (14) working days** upon its receipt at VANTRON. If such turnaround cannot be provided by VANTRON due to causes beyond the control of VANTRON, VANTRON shall document such instances and notify the CUSTOMER immediately.