

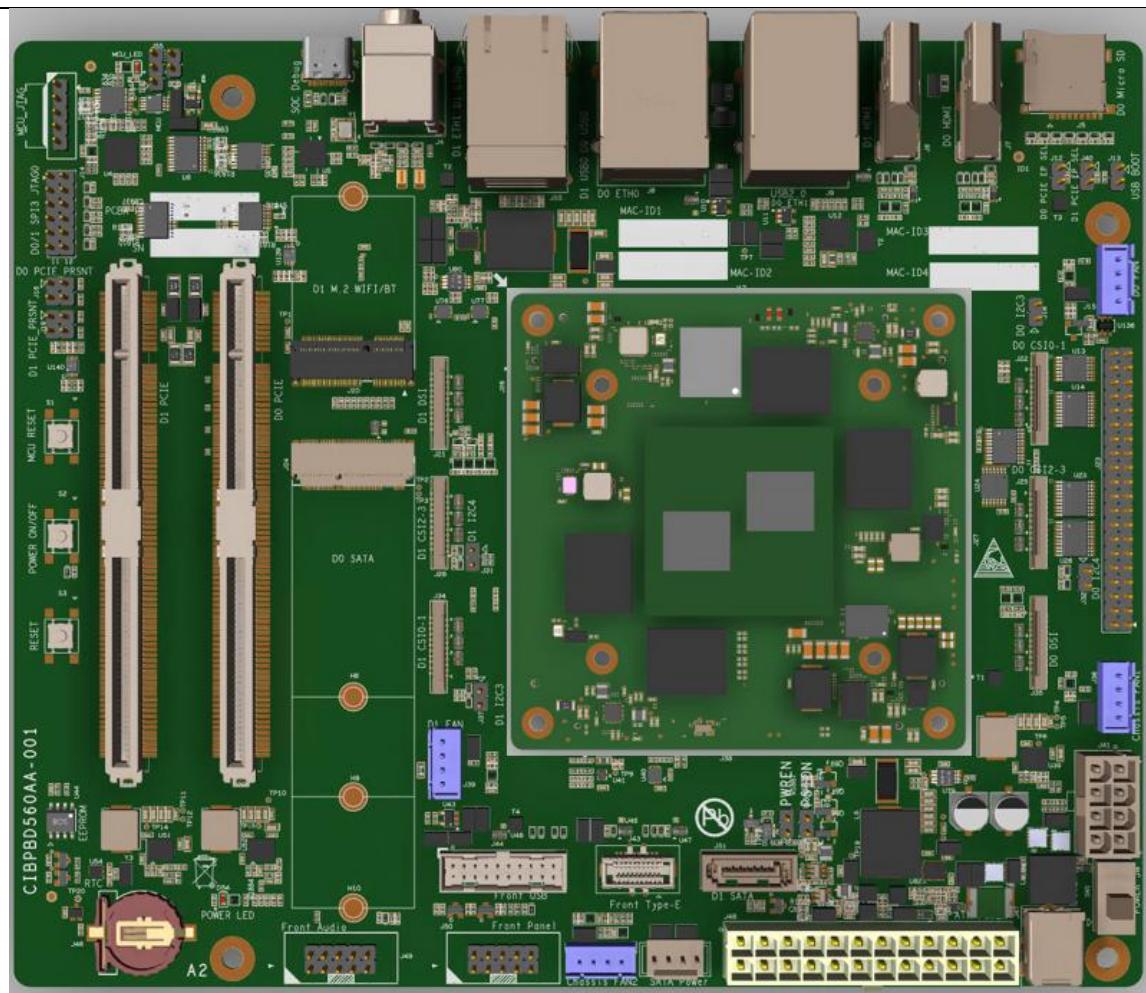
ESWIN

EBC77 Series Development Board User Manual

EIC7702X SoC

Version 1.5

2025/10/11



Notice

Copyright © 2025 by Beijing ESWIN Computing Technology Co., Ltd., and its affiliates ("ESWIN"). All rights reserved.

ESWIN retains all intellectual property and proprietary rights in and to this product. Except as expressly authorized by ESWIN, no part of the software may be released, copied, distributed, reproduced, modified, adapted, translated, or created derivative work of, in whole or in part.

INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND DOES NOT REPRESENT A COMMITMENT ON THE PART OF ESWIN. UNLESS OTHERWISE SPECIFIED, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT ARE PROVIDED "AS IS" WITHOUT WARRANTIES, GUARANTEES, OR REPRESENTATIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED.

"ESWIN" logo, and other ESWIN icons, are trademarks of Beijing ESWIN Computing Technology Co., Ltd. And(or) its parent company.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Revision Record

Document version	Date	Notice
Rev 1.5	2025/10/11	Document format update
Rev 1.4	2025/09/30	Card name update
Rev 1.3	2025/09/28	Document release update
Rev 1.2	2025/04/14	EVB2 changed to EBC7702. The alignment issue between the figure and the picture has been resolved.
Rev 1.0	2025/03/04	Original Version

Contents

EBC77 SERIES DEVELOPMENT BOARD USER MANUAL	0
EIC7702X SoC	0
CONTENTS	III
CONTENT OF FIGURES	IV
CONTENT OF TABLES	VI
1 SAFETY REMINDER	1
1.1 COPYRIGHT STATEMENT	1
1.2 IMPORTANT SAFETY INFORMATION	1
2 INTRODUCTION	1
2.1 EBC7702 DEVELOPMENT BOARD COMPNENTS	2
3 DESIGN BLOCK DIAGRAM	5
3.1 SYSTEM DIAGRAM	5
3.2 EBC7702 DEVELOPMENT BOARD POWER DIAGRAM	6
3.3 EBC7702 DEVELOPMENT BOARD EFFECT PICTURE	6
3.4 EBC7702 DEVELOPMENT BOARD PARTS LIST	8
3.5 EBC7702 DEVELOPMENT BOARD KEY FEATURES	8
4 HARDWARE INTRODUCTION	10
4.1 POWER ON/OFF	10
4.2 BUTTON	11
4.3 BOOT PIN SELECTION	12
4.4 REAR PANEL INTERFACE(INCLUDE DEBUG INTERFACE)	13
4.5 FRONT PANEL INTERFACE	14
4.6 SDIO WiFi/BT MODULE	16
4.7 PCIE	16
4.8 SATA	17
4.9 LED	18
4.10 40 PIN HEADER	18
4.11 TEST HEADERS	19
4.12 GND TEST PIN	23
5 SOFTWARE STACK USAGE AND BOARD-LEVEL IMAGE FLASHING	24
6 FLASHING PROGRAM AND BOARD INFORMATION	24
6.1 MCU	24
6.2 SOM	28

Content of Figures

Figure 2-1 EBC7702 Development Board Core-Board	2
Figure 2-2 EBC7702 Development Board Carrier-Board	3
Figure 3-1 EIC7702X Chip Diagram	5
Figure 3-2 EBC7702 Development Board System Diagram	5
Figure 3-3 EBC7702 Development Board Clock Diagram	6
Figure 3-4 EBC7702 Development Board Power Tree	6
Figure 3-5 EIC7702X Core Board Effect Picture	7
Figure 3-6 EBC7702 Development Board Carrier Board Effect Picture	7
Figure 4-1 ATX Power pin out	11
Figure 4-2 ATX Power connector	11
Figure 4-3 Button	12
Figure 4-4 BOOT Selection	12
Figure 4-5 Rear Panel Interface	13
Figure 4-6 DEBUG interface	13
Figure 4-7 Front panel interface diagram	14
Figure 4-8 Front Audio pin out	14
Figure 4-9 Front Panel pin out	15
Figure 4-10 Front USB Header(J44) pin out	15
Figure 4-11 Front Panel USB Plug	15
Figure 4-12 Type-E(J43) Connector pin out	16
Figure 4-13 Type-E plug	16
Figure 4-14 SDIO WiFi/BT	16
Figure 4-15 J12/J40 PCIE RC,EP mode select Header	17
Figure 4-16 Present detection Configuration Header	17
Figure 4-17 SATA power	18
Figure 4-18 EBC7702 Carrier board 40 PIN Header	19
Figure 4-19 40 Pin Header schematic	19
Figure 4-20 Test Pin	20
Figure 4-21 J11 MCU SWD Header	21
Figure 4-22 J14 JTAG0 SPI3 Test Header	21
Figure 4-23 USB Boot Configure Header	21
Figure 4-24 J15 D0 FAN	21
Figure 4-25 J39 D1 FAN	22
Figure 4-26 J36 Chassis FAN1	22
Figure 4-27 J33 Chassis FAN1	22
Figure 4-28 J1 JTAG0 SW default select Header	22

Figure 4-29 J45 Auto Power on Header	23
Figure 4-30 J42 Cancel Auto Power on Header	23
Figure 4-31 GND Test Pins	24
Figure 6-1 MCU_JTAG Wiring Reference	25
Figure 6-2 The detected MCU device information	25
Figure 6-3 MCU Successfully Flashed	25
Figure 6-4 J-Link Device Driver	26
Figure 6-5 J-Flash start	26
Figure 6-6 Configure the Chip Model	27
Figure 6-7 Verify Flash Address	27
Figure 6-8 Connect the Device	27
Figure 6-9 Open the File	28
Figure 6-10 MCU Successfully Flashed	28
Figure 6-11 EBC7702 Development Board Hardware Description	29
Figure 6-12 BOOT SEL Configuration	30
Figure 6-13 Obtain the device path of the USB drive	30
Figure 6-14 Creat the Partition Table	31
Figure 6-15 Edit Partition Information and Repartition in Uboot	32
Figure 6-16 Mode Confirmation	32
Figure 6-17 USB Rescan and Recognition	33
Figure 6-18 Die0 download and upgrade the bootloader image	34
Figure 6-19 Die1 download and upgrade the bootloader image	35
Figure 6-20 USB boot connection position	36
Figure 6-21 Board Corresponding Block Device Number	37
Figure 6-22 Distinction between die0 and die1 interfaces	37
Figure 6-22 Die0 is flashed using the recovery file corresponding to its display	38
Figure 6-24 Die1 is flashed using the recovery file corresponding to its display	38
Figure 6-25 Start flashing the Ubuntu version	39
Figure 6-26 Device boots up	39
Figure 6-27 Ubuntu version upgrade	39
Figure 6-28 Die0 bootloader upgrade	40
Figure 6-29 Die1 bootloader upgrade	41
Figure 6-30 If the phase adjustment temperature exceeds 40°C, cooling is required first	42
Figure 6-31 Phase adjustment in progress	42

Content of Tables

Table 2-1 EBC7702 Development Board Component Labels	3
Table 3-1 EBC7702 Development Board Parts List	8
Table 3-2 EBC7702 Development Board Key Features	8
Table 4-1 SW1 Net table	12
Table 4-2 BOOT_SEL Config table	12
Table 4-3 LED Define	18
Table 4-4 Test Headers	19
Table 6-1 Hardware Description	28
Table 6-2 Bootsel sw Signal List	29

1 Safety Reminder

1.1 Copyright Statement

The "EBC7702 Development Board" provided by ESWIN is solely for the use of end-users, and is only used for end-users to test and evaluate ESWIN's semiconductor components and software applications ("expected use"). No one is allowed to resell, redistribute, transfer this product or otherwise profit from it.

Please note that the user manual for the EBC7702 Development Board may be modified from time to time. You can contact ESWIN sales and after-sales for updates.

1.2 Important Safety Information

1.2.1 Warning

- Any power supply used in conjunction with the EBC7702 Development Board must comply with the local regulations and standards.
- The EBC7702 Development Board must be operated in a well-ventilated environment.
- This product can only be used on a flat, non-conductive surface, and must never come into contact with conductors at any time.
- Connecting incompatible devices to the EBC7702 may damage the development board.
- All accessories and peripheral devices used together with the EBC7702 development board must comply with the local regulations and standards, and must be labeled accordingly to ensure compliance with safety requirements.
- All cables and connection devices used together with the EBC7702 Development Board must be properly insulated in accordance with safety standards and requirements.

1.2.2 Safety Instructions

- The following requirements must be followed to prevent the EBC7702 Development Board from malfunctioning or getting damaged.
- Do not allow the development board to come into contact with water or be exposed to moisture.
- During operation, do not place or position the development board near conductive surfaces.
- Do not expose the development board to high temperatures. According to this document, the development board can only be used in a room temperature and air-flow environment.
- Avoid any mechanical or electrical damage to the printed circuit board, interfaces, and integrated circuits on the ESWIN EBC7702 Development Board.
- When the EBC7702 Development Board is powered on, avoid touching, picking up, or moving the board cards.

2 Introduction

The EBC7702 development platform is composed of two boards: SOM board and Carrier board. The EBC7702 Development Board includes the EIC7702X, LPDDR5 SDRAM, EMMC, EPHY, Power IC and Reset IC, which constitute the minimum system. The main components of the EBC7702 Development Board are the interface circuits and power supply system. This platform supports the Mini-DTX size specification (170X203mm), and it can also be placed in a 1U chassis.

2.1 EBC7702 Development Board Components

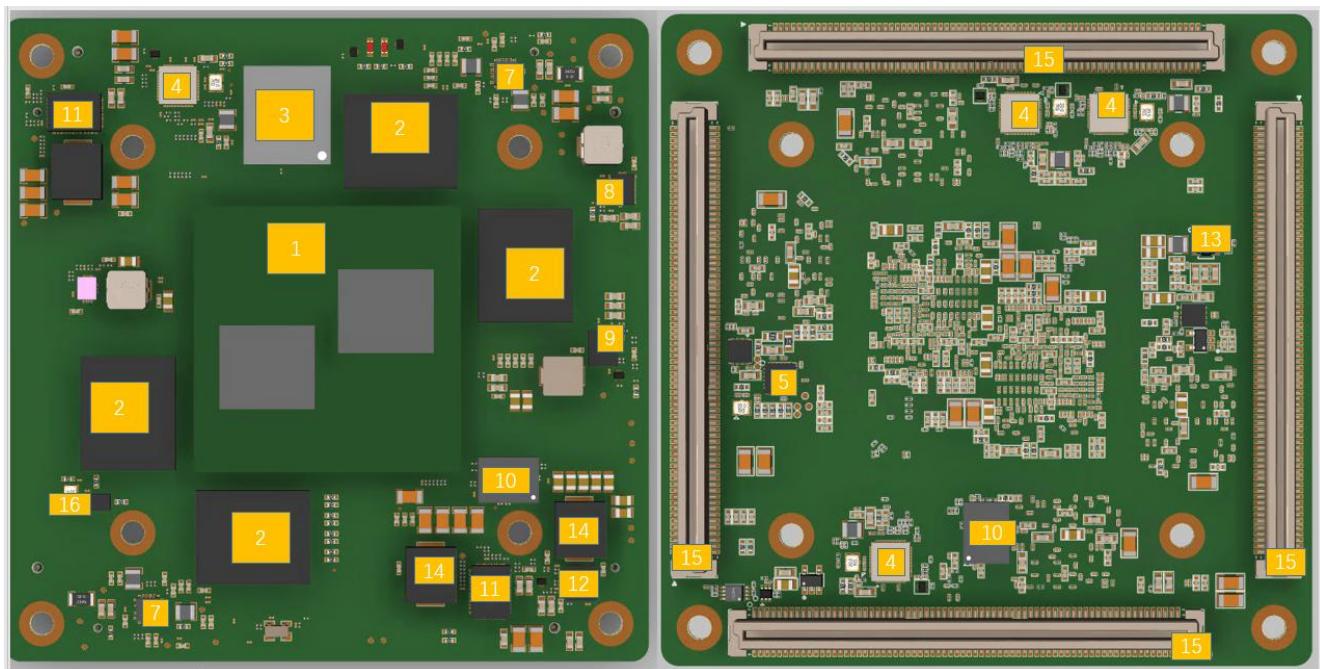


Figure 2-1 EBC7702 Development Board Core-Board

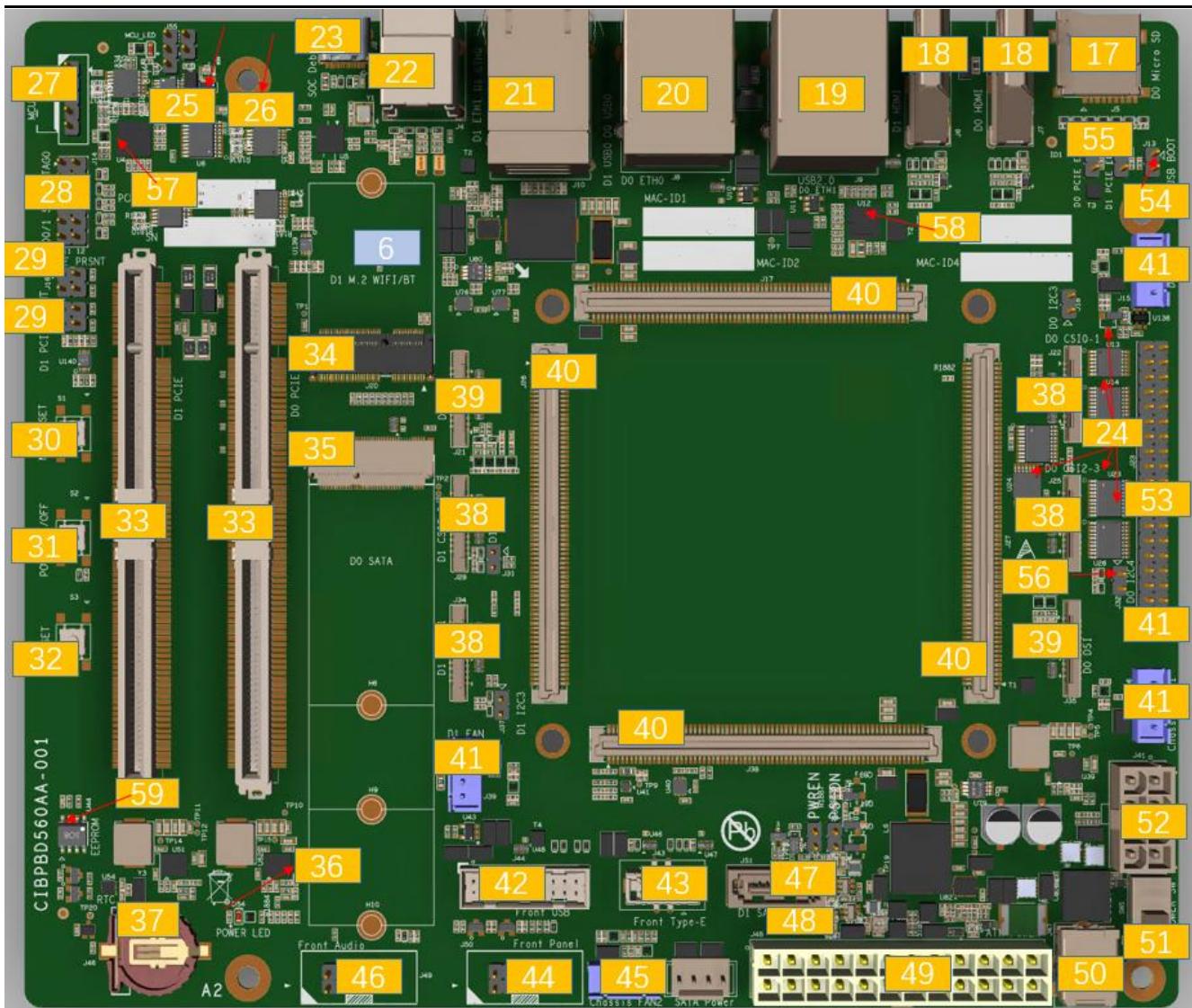


Figure 2-2 EBC7702 Development Board Carrier-Board

Table 2-1 EBC7702 Development Board Component Labels

Number	Components Name	Number	Components Name	Number	Components Name
1	EIC7702X Chip	28	JTAG0,SPI3 Test header	55	PCIERC,EP Selector Header
2	4x8 32GB LPDDR5 SDRAM	29	PCIE PRSNT	56	Die0 I2C4 Header
3	32GB eMMC flash	30	MCU RESET	57	MCU chip
4	Ethernet PHYS	31	POWER ON/OFF	58	USB2.0 HUB chip
5	SerDes ClockBuffer	32	EIC7702X RESET	59	EEPROM
6	PCIE ClockBuffer	33	4 lane X16 PCIe		
7	LPDDR5 PMICS	34	M.2 E-KEY SDIO WIFI/BT module		
8	VDDIO3.3 PMICS	35	M.2 M-KEY SATA		

EBC77 Series Development Board User Manual

9	VDD_CPU PMICS	36	POWER LED		
10	2x16GB SPI flash	37	RTC battery holder		
11	VDD_NPU PMICS	38	MIPI CSI connector		
12	VDD_SOC PMICS	39	MIPI DSI connector		
13	VDDIO18 PMICS	40	Female connector for board-to-board connection		
14	0624 type power inductor	41	Fan connector		
15	Male connector for board-to-board connection	42	FrontUSB Header		
16	EIC7702X ClockBuffer	43	Type-E USB3.0 connector		
17	Micro-SD card slot	44	Front Panel connector		
18	2xHDMI OUT	45	SATA3.0 connector		
19	2xUSB2.0, RJ45 (Can be remotely activated)	46	Front Audio Header		
20	2xUSB3.0, RJ45	47	5V,12V connector		
21	2xRJ45	48	AutoPower Header		
22	Audio JACK	49	ATX Power Header		
23	EIC7702X Debug	50	DC 12V Header		
24	electrical level shift	51	Power Main Switch		
25	EIC7702X&MCU JTAG MUX	52	ATX CPU 12V Power Header		
26	USB debug Reset LED	53	GPIO/SPI/UART/I2C Header		
27	MCU JTAG	54	BootFromUSB Header		

3 Design block diagram

3.1 System Diagram

3.1.1 EBC7702 Development Board Chip Diagram

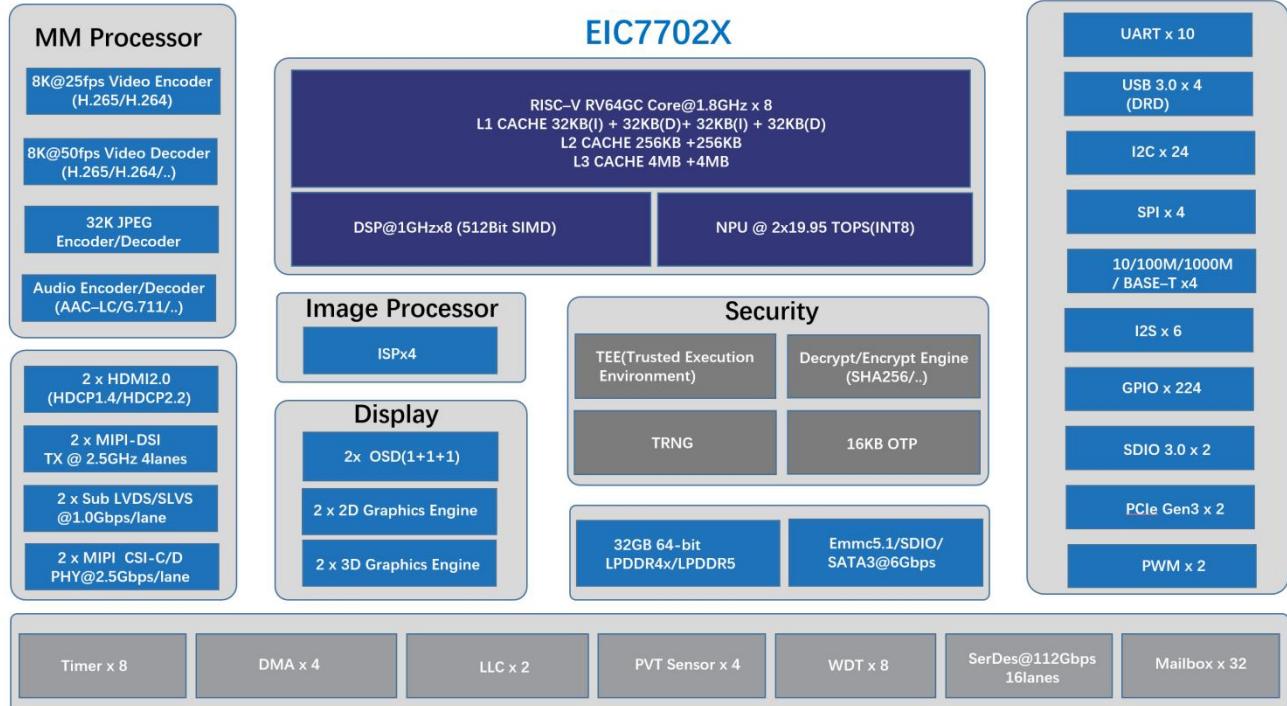


Figure 3-1 EIC7702X Chip Diagram

3.1.2 EBC7702 System Diagram

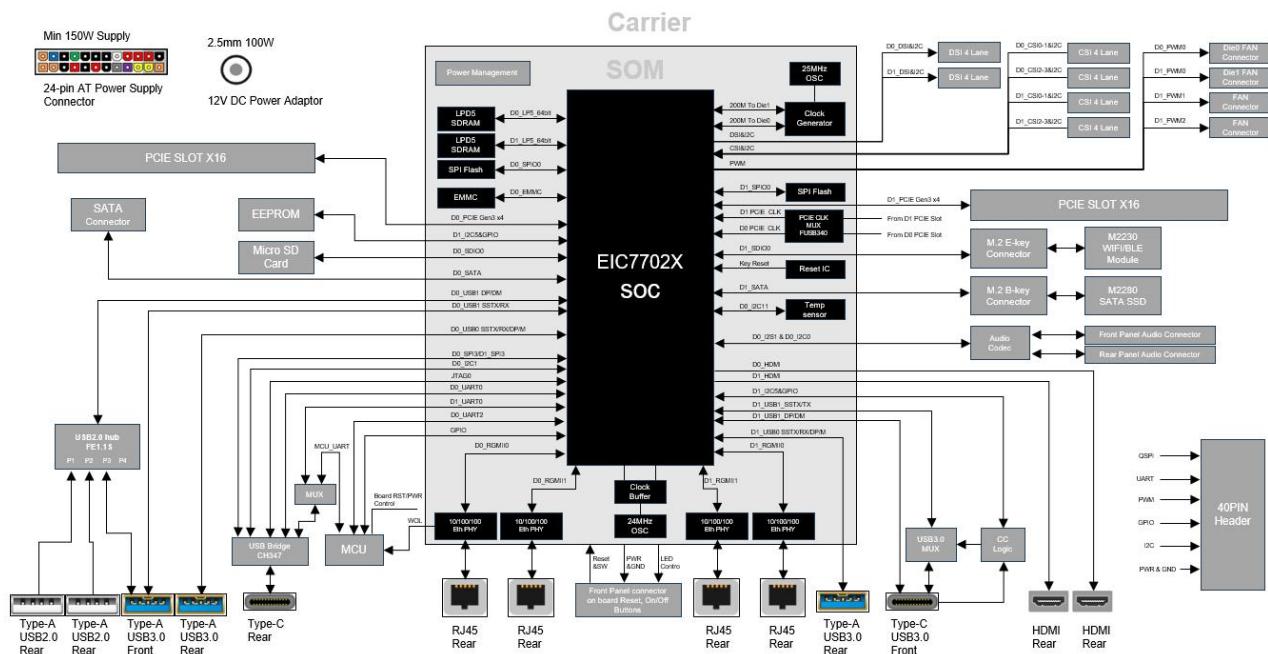


Figure 3-2 EBC7702 Development Board System Diagram

3.1.3 EBC7702 Development Board Clock Diagram

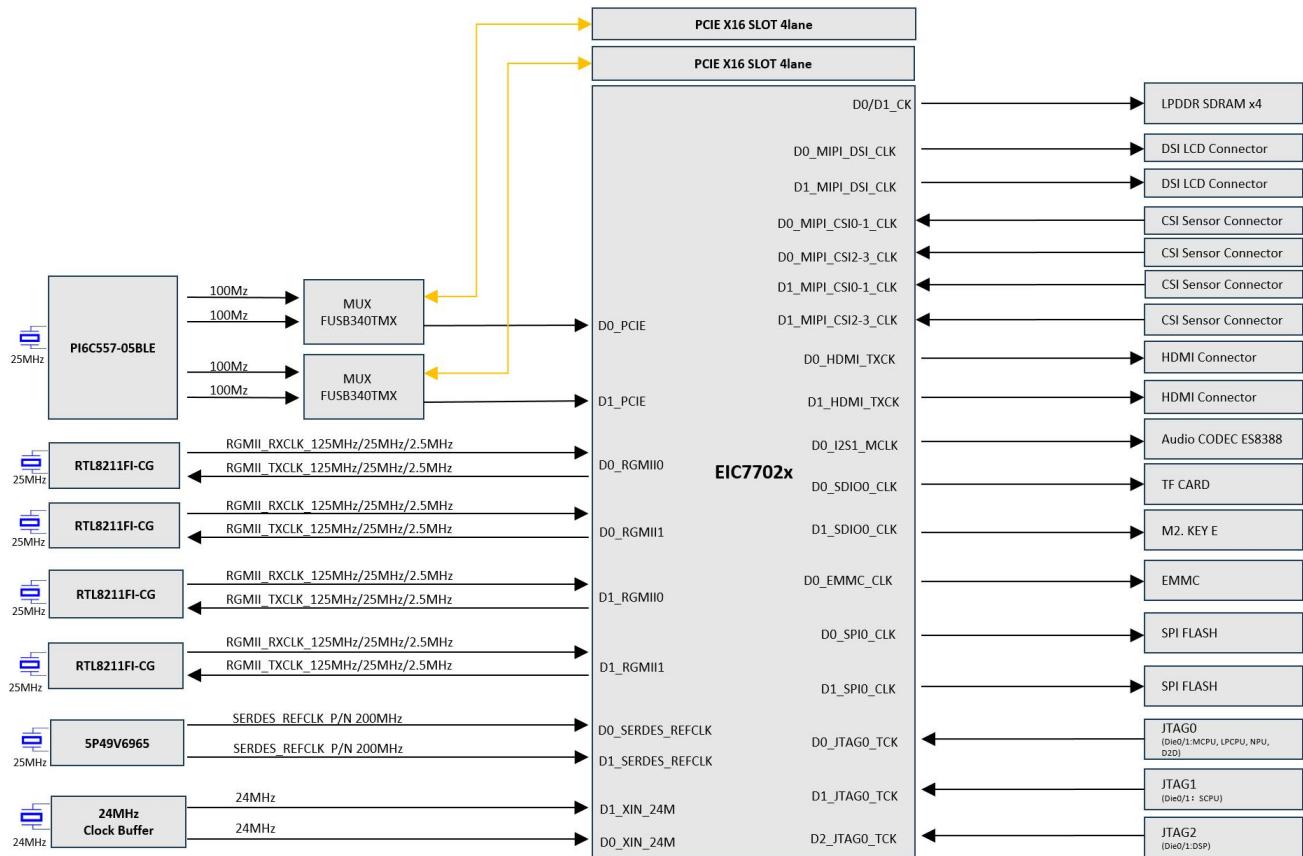


Figure 3-3 EBC7702 Development Board Clock Diagram

3.2 EBC7702 Development Board Power Diagram

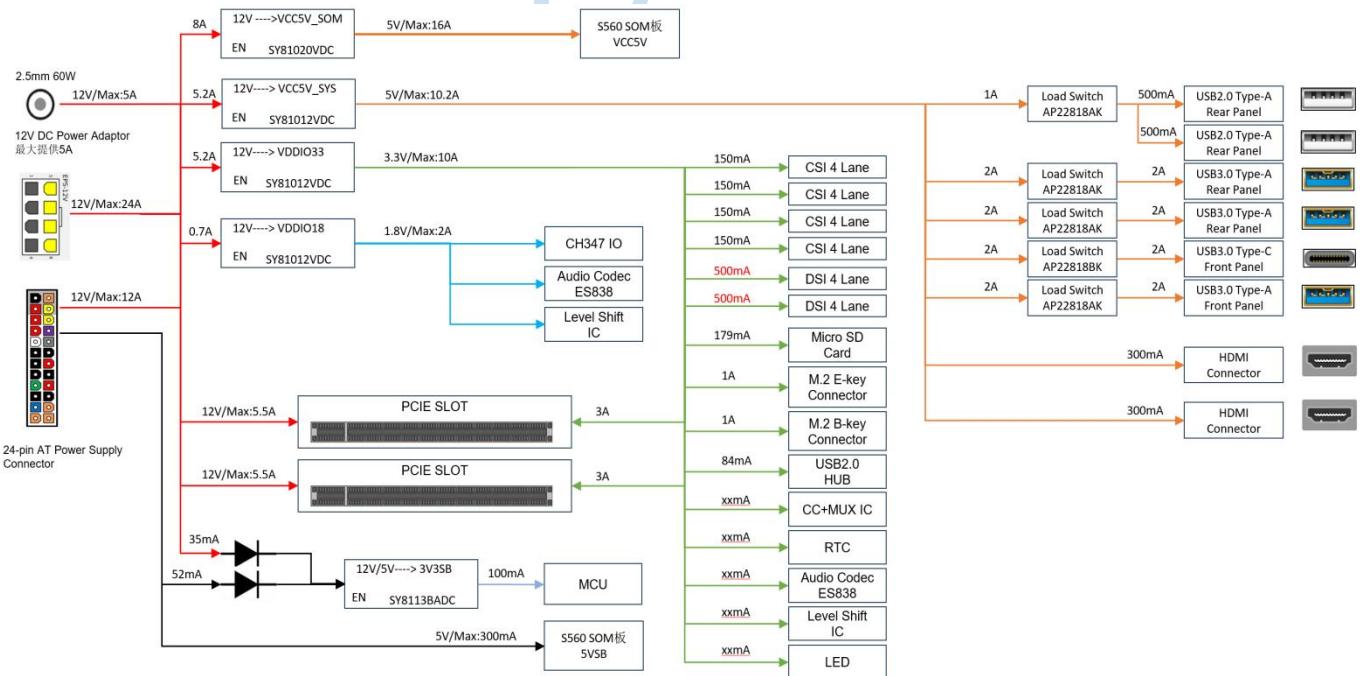


Figure 3-4 EBC7702 Development Board Power Tree

3.3 EBC7702 Development Board Effect Picture

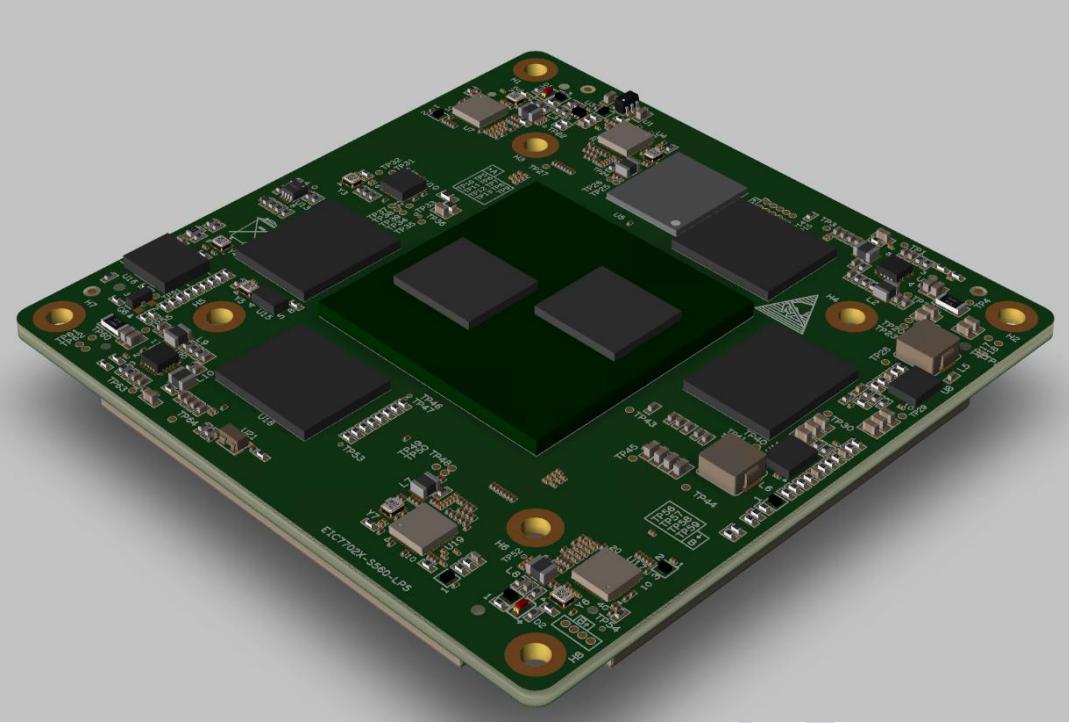


Figure 3-5 EIC7702X Core Board Effect Picture

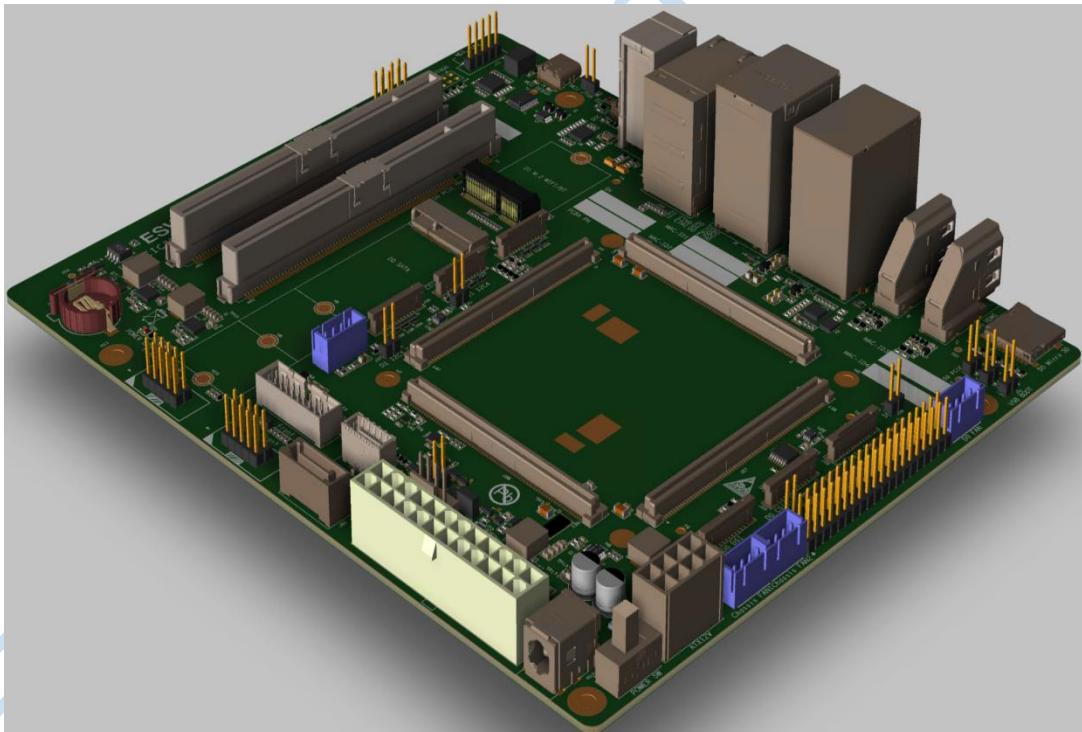


Figure 3-6 EBC7702 Development Board Carrier Board Effect Picture

3.4 EBC7702 Development Board Parts List

Table 3-1 EBC7702 Development Board Parts List

Item	Name	Nnumber	Parameters
1	EBC7702 Core Board EBC7702 Carrier Board	1	Includes fan + radiator
2	USB Type-A to Type-C Cable	1	
3	USB Type-A to Type-A Cable	1	
4	HDMI Cable	1	
5	NetWork Cable	1	
6	Screw/Stud	4	Screw: M3 x5mm, Stud: M3x15mm
7	IO Shield	1	

3.5 EBC7702 Development Board Key Features

Table 3-2 EBC7702 Development Board Key Features

Item	EBC7702 Development Board Core Board
CPU	RISC-V RV64GC 8 cores, Support cache consistency The clock frequency reaches up to 1.8GHz. L1 Cache 64KB(I) + 64KB(D) (privately-owned) L2 Cache 512KB (privately-owned) L3 Cache 8MB (privately-owned) Cache supports ECC (supports SECDED)
DNN Accelerator	40TOPS (INT8) 20TOPS (FP16 Or INT16)
Vision DSP	Multi-core DSP dual cluster, 1040MHz, supporting 512-bit SIMD and single-cycle 512 INT8 MAC operations
Memory	4X8GB LPDDR5 SDRAM
External Storage	32GB EMMC5.1 2X16MB SPI flash
RAS	SRAM DDR supports parity check and ECC
Video Encode	Video encoding supports up to 8K at 50fps or 26 channels of 1080P at 30fps. H.265 (HEVC): ITU-T Rec. H.265 (04/2013), ISO/IEC 23008-2 Main Profile, Level 5.1, High Tier Main10 profile, Level 5.1, High Tier Main Still Profile

	<p>H.264 (AVC):</p> <p>Spec Version 12:ISO/IEC 14496-10 / ITU-T Rec. H.264 (03/2010)</p> <ul style="list-style-type: none"> Baseline Profile, levels 1 - 5.2 Main Profile, levels 1 - 5.2 High Profile, levels 1 - 5.2 High 10 Profile, levels 1 - 5.2
Video Decode	<p>Video decoding supports up to 8K@100fps or 64 channels of 1080P@30fps</p> <p>H.265 (HEVC):</p> <p>ITU-T Rec. H.265 (04/2013), ISO/IEC 23008-2</p> <ul style="list-style-type: none"> Main Profile, up to Level 5.1, High Tier Main10 profile, up to Level 5.1, High Tier Main Profile, Level 6, High Tier Main10 profile, Level 6, High Tier Main Still Profile <p>H.264 (AVC):</p> <p>Spec Version 12:ISO/IEC 14496-10 / ITU-T Rec. H.264 (03/2010)</p> <ul style="list-style-type: none"> Baseline Profile, Levels 1 – 5.2 (up to 4K) Main Profile, Levels 1 – 5.2 (up to 4K) High Profile, Levels 1 – 5.2 (up to 4K) Constrained Baseline, levels 1 – 5.2 (up to 4K) Progressive High profile, levels 1 – 5.2 (up to 4K) High 10 profile (progressive only), levels 1 – 5.2 (up to 4K) High 10 Intra profile (progressive only), levels 1 – 5.2 (up to 4K) Constrained Baseline, level 6 (up to 8K) Progressive High profile, level 6 (up to 8K) High 10 profile (progressive only), level 6 (up to 8K) High 10 Intra profile (progressive only), level 6 (up to 8K)
JPEG Codec	<p>JPEG ISO/IEC 10918-1, ITU-T T.81.</p> <p>Supports the maximum resolution of 32K x 32K</p> <p>Baseline process (Internal Supports Huffman coding Interleaved YUV420, YUV422, Monochrome)</p> <p>Lossless process (Internal Supports 8-bit with Huffman coding Interleaved YUV420, Monochrome)</p> <p>MJPEG format (T.81 Annex H) in AVI container</p>
Vision Engine	2 x HAE (2D Blit, Crop, Resize, Normalization)
GPU	2 x 3D GPU (support OpenGL-ES 3.2、 EGL 1.4、 OpenCL 1.2/2.1 EP2、 Vulkan 1.2、 Android NN HAL)
Display System	2 x OSD (Three image layers, namely the video layer, the mouse layer, and the background layer)
DIE2DIE	Bidirection 16 x lanes 56Gbps per lane
LLC	2 x 4MB (128Bytes cache line, 16 ways associativity) ECC(Support SECDED)
Security	TEE (Trusted Execution Environment), TRNG, ECDSA RSA4096, AES, SM4, DES, HMAC(MD5,SHA-1,SHA-224,SHA256,SM-3,SHA512/224,SHA-512/256,SHA-384),CRC32,Dual-core hardware acceleration 32KB OTP Dual-core hardware acceleration

	2 x RV32I monokaryon L1 CACHE 64KB(I) + 64KB(D)
SATA	Standard interface 1 M.2 M key SATA interface 1
EEPROM	2Kbit storage capacity
SD Card	Micro SD card slot 1
USB Interface	USB2.0 interface 2 USB3.0 interface 2
RJ45	Four Gigabit network ports, among which D0_ETH0 has the function of remote wake-up.
TypeC	Type-C EIC7702X debug interface and MCU debug interface (J2)
HDMI	HDMI two interface terminals
CSI Interface	Four Physical connector interface D1 CSI0~1(J34) D1 CSI2~3(J28);D0 CSI0~1(J22) D0 CSI2~3(J25)
DSI Interface	D1 DSI1(J21) D0 DSI1(J35)
PCIE X16	D0 PCIE (J30) D1 PCIE (J29)
FAN	Four fan interfaces (D0/1 fan, bottom board 0/1 fan)
WIFI/BT	M.2 E key interface connects to WIFI/BT
AudioJack	Headphone/Microphone interface (J4)
Connector (J23)	GPIO PWM UART...
EBC7702 Dimensions	170X203mm
Working temperature	TBD

4 Hardware Introduction

4.1 Power On/Off

EBC7702 Development Board 24pin ATX Power and DC12V simultaneously power supply.

When the load power is relatively high, an ATX power supply should be used, and the 8-pin ATX CPU 12V power header should be connected simultaneously to increase power supply capacity.

After plugging in the power, turn on/off the device through the Main Switch. Remember to install the heat sink before use, and connect the fan plug as shown in the picture below.

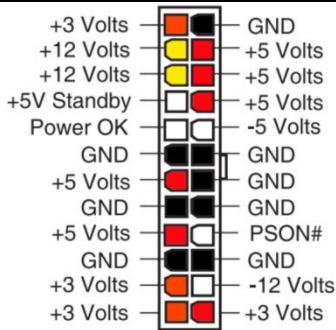


Figure 4-1 ATX Power pin out

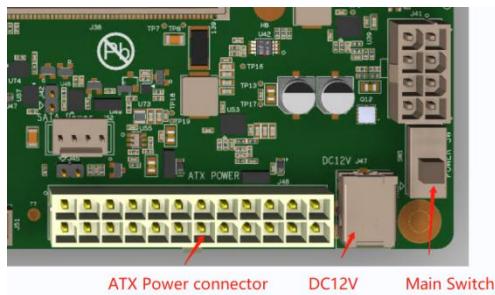


Figure 4-2 ATX Power connector

Power-on sequence:

1. Plug in the USB Type-C data cable, HDMI cable, USB interface mouse, keyboard, network cable, audio cable and other external accessories on the rear panel.
2. Plug in the DC12V adapter or ATX power supply, flip the 'Main Switch' to the ON position, and the POWER LED D54 on the EBC7702 development board will light up green.
3. Press 'POWER ON/OFF' to turn on the device. The SOC will power up, and you will see LED D54 light up. The power light on the SOM will turn green. After the SOC has finished starting, the power light on the SOM will blink green.
4. Press 'Power ON/OFF' again for 10 seconds to turn off the device.

4.2 Button

EBC7702 Development Board supports three on-board buttons: Power Button, SOC reset Button, and MCU Reset Button. The positions of the buttons are shown in the following diagram.

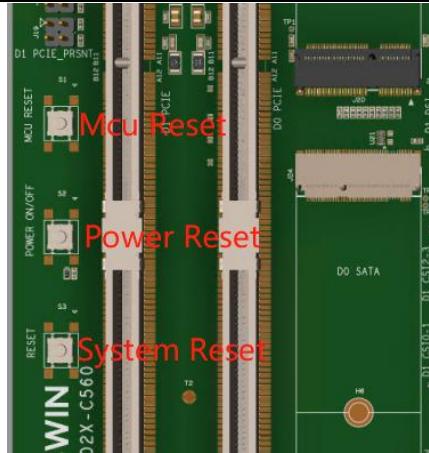


Figure 4-3 Button

4.3 Boot Pin Selection

EBC7702 Development Board startup method is fixed on the SOM board. As shown in the figure below, set bit 0100, and it defaults to starting from SPI Flash. On the carrier board, USB0 startup can be achieved by short-circuiting the J13 jumper through configuration.

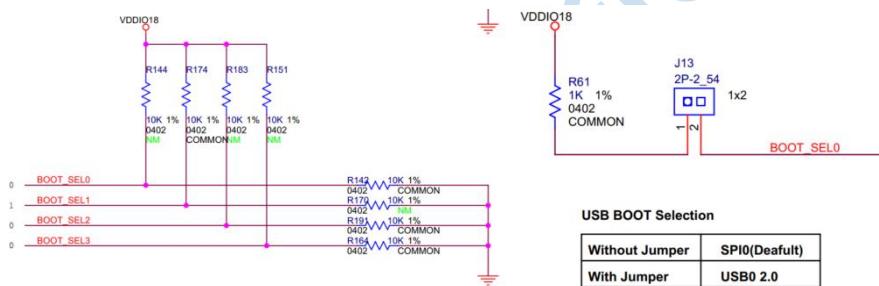


Figure 4-4 BOOT Selection

Table 4-1 SW1 Net table

Net	Net Define	Levle	Note
BOOT_SEL0	Boot selection 0	0	
BOOT_SEL1	Boot selection 1	1	
BOOT_SEL2	Boot selection 2	0	
BOOT_SEL3	Boot selection 3	0	

Table 4-2 BOOT_SEL Config table

security mode	boot_sel[3]	boot_sel[2]	boot_sel[1:0]	bootrom src	boot CPU	uboot src
1'b1	X	X	2'b00	ROM	SCPU	UART
			2'b10			Boot-SPI

security mode	boot_sel[3]	boot_sel[2]	boot_sel[1:0]	bootrom src	boot CPU	uboot src
			2'b11			USB
1'b0	1'b0	1'b0	2'b00	ROM	SCPU	UART
			2'b10			Boot-SPI
			2'b11			USB
	1'b1	1'b1	2'b00	Boot-SPI	MCPU	UART
			2'b10			Boot-SPI
			2'b11			USB
	1'b1	X	2'b00	Boot-SPI	MCPU	UART
			2'b10			Boot-SPI
			2'b11			USB

4.4 Rear Panel Interface(include DEBUG Interface)

The rear panel interface diagram is shown in the following figure.



Figure 4-5 Rear Panel Interface



Figure 4-6 DEBUG interface

SOC Type-C is mainly used for debugging EIC7702X.

D0 ETH0(J8) has a network wake-up function. The EBC7702 development board can be woken up via the network when in standby mode. Its design principle utilizes the WOL function of the PHY chip, and in terms of hardware design, VSB provides standby constant power.

Note:

- D1 USB3.0 Up interface, physical connection D1_USB0-3.0, D1_USB0-2.0
- D1 USB3.0 Down interface, physical connection D0_USB0-3.0, D0_USB0-2.0

- D0 USB2.0 Up interface, physical connection D0_USB1-2.0
- D0 USB2.0 Down interface, physical connection D0_USB1-2.0

4.5 Front Panel Interface

EBC7702 Development Board supports Mini-DTX chassis. The chassis comes with a connecting cable for the front panel interface. EBC7702 Development Board is equipped with the corresponding sockets for these connecting cables, including Front Audio (J49), Front Panel (J50), Front USB (J44), and Front Type-E (J43) sockets. The position diagram is as follows.

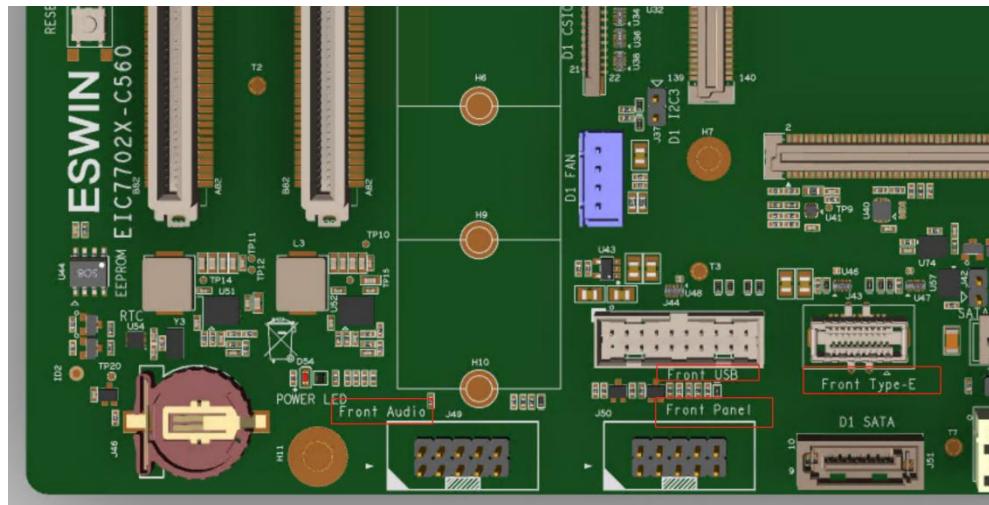


Figure 4-7 Front panel interface diagram

4.5.1 Front Audio Pin Out

Pin definitions of Front Audio (J49) are shown in the following figure.

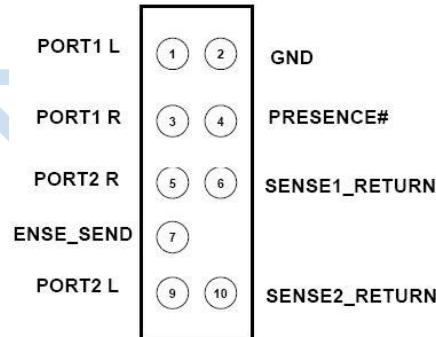


Figure 4-8 Front Audio pin out

4.5.2 Front Panel Pin Out

Front Panel Header(J50) pins defined as shown in the figure below

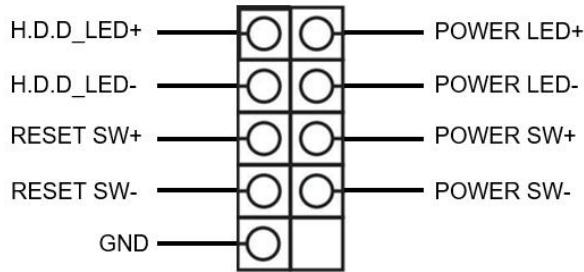


Figure 4-9 Front Panel pin out

4.5.3 Front USB Pin Out

Pin definitions of the Front USB Header (J44) are shown in Figure 19. The corresponding plug is shown in Figure 20.

Pin No.	Definition	Pin No.	Definition
1	VBUS	11	D2+
2	SSRX1-	12	D2-
3	SSRX1+	13	GND
4	GND	14	SSTX2+
5	SSTX-	15	SSTX2-
6	SSTX+	16	GND
7	GND	17	SSRX2+
8	D1-	18	SSRX2-
9	D1+	19	VBUS
10	NC		

Figure 4-10 Front USB Header(J44) pin out



Figure 4-11 Front Panel USB Plug

4.5.4 Front Type-E Connector

The pin definitions of the Front Type-E (J43) Connector are shown in below Figure. Most chassis with Type-C interfaces on the front panel are equipped with a Type-E to Type-C adapter cable. The Type-E plug is shown in below Figure.

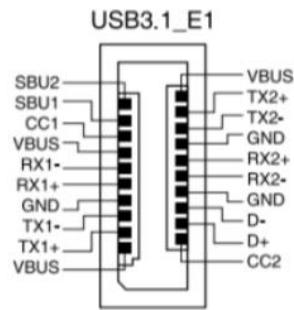


Figure 4-12 Type-E(J43) Connector pin out



Figure 4-13 Type-E plug

4.6 SDIO WiFi/BT Module

WiFi/BT Module interface of the M.2 E-KEY interface is located as follows:

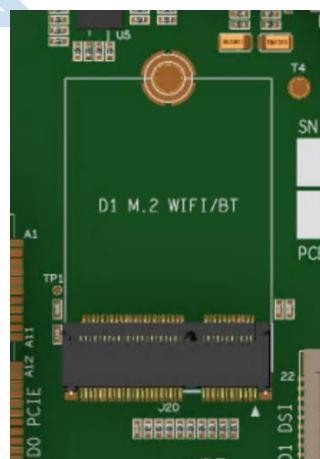


Figure 4-14 SDIO WiFi/BT

4.7 PCIE

Provides two PCIe x16 connectors, but in actual use they operate as 4 lanes, corresponding to Die0 and Die1 respectively;

This slot has a load capacity of 12V 5.5A. When the load power is relatively high, for example exceeding 60W, it is recommended to use an ATX 24-pin power supply. When the total load power exceeds 144W, an

ATX 24-pin power supply should be used along with connecting the 8-pin ATX CPU 12V power header to increase power supply capacity.

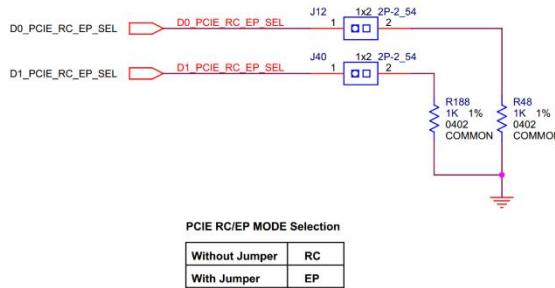


Figure 4-15 J12/J40 PCIE RC,EP mode select Header

Switch modes by installing jumper caps on J12/J40; installing them sets EP mode, not installing sets RC mode.

NOTE: PCIE_RC_EP_SEL are used to determine whether the EBC7702 operates in RC mode or EP mode.

In J12/J40, PIN1 and PIN2 are connected. PCIE_RC_EP_SEL is grounded. At this time, EBC7702 Development Board operates in EP mode, and the PCIE CLK is provided by an external RC.

In J12/J40, PIN1 and PIN2 are disconnected. The PCIE_RC_EP_SEL is defaulted to be pulled high. At this time, EBC7702 Development Board operates in RC mode and provides the PCIE CLK signal.

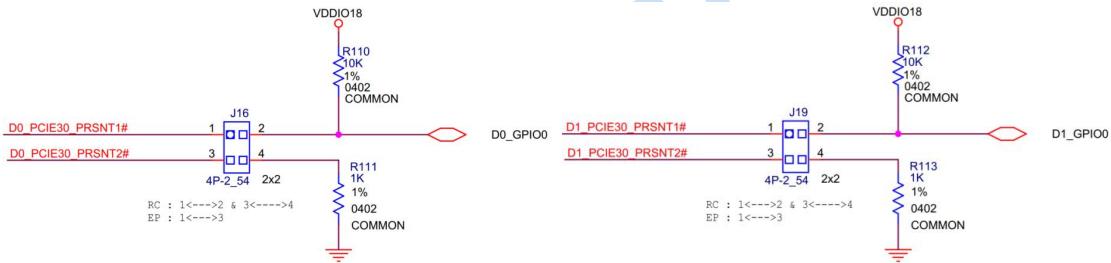


Figure 4-16 Present detection Configuration Header

NOTE: PCIE30_PRSNT1#, PCIE30_PRSNT2# are connected to PRSNT1#, PRSNT2# in the PCIE SLOT respectively.

The EBC7702 operates in RC mode. In the above diagram, PIN1 and PIN2 are connected, PCIE30_PRSNT1# is connected to GPIO0 for hot-plug event detection, and GPIO0 has interrupt functionality. In the above diagram, PIN3,PIN4 are connected to ground the PCIE30_PRSNT2#.

EBC7702 operates in EP mode. In the picture above, PIN1, PIN3 are connected; no PCIE insertion detection is performed.

4.8 SATA

When powered by a 12V DC power supply, the onboard SATA power connector can be used to power SATA loads, with the pinout shown in the diagram:

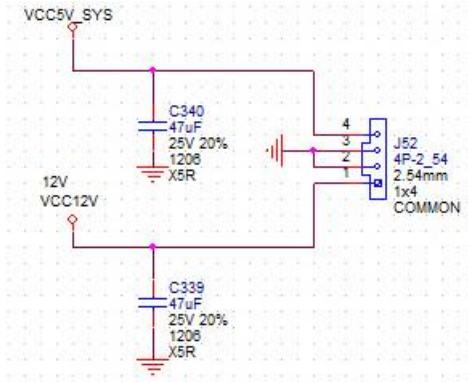


Figure 4-17 SATA power

When installing an ATX power supply inside the case, you can use the case power cables to supply power to SATA loads. When the load power is high, refer to the power supply rules in section 4.7 and connect the power appropriately.

4.9 LED

Table 4-3 LED Define

LED Define	Part	Description
MCU System Power LED	D1	The MCU system powers on and displays bright green light.
System Power LED	D54	SOC system powers on, displays bright green color

4.10 40 Pin Header

The EBC7702 connects the extra signals to the 40-pin header. The positions and pin definitions are shown in the following figure.

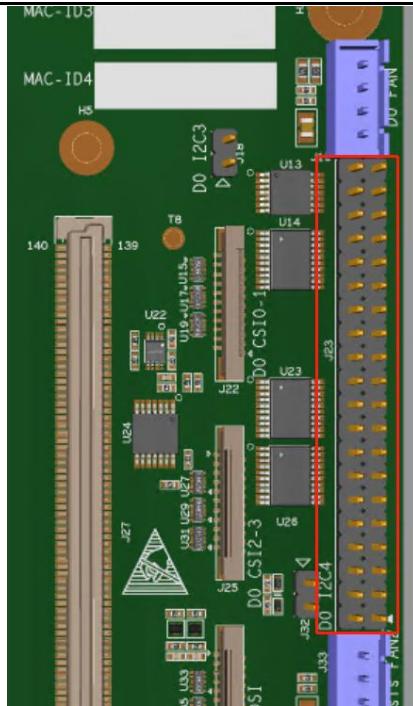


Figure 4-18 EBC7702 Carrier board 40 PIN Header

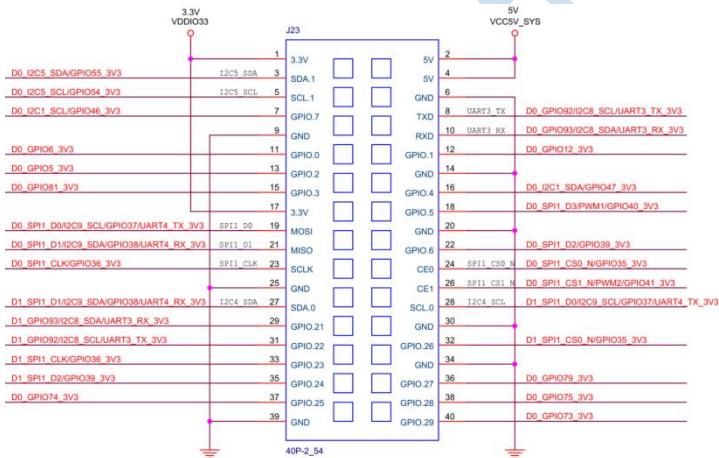


Figure 4-19 40 Pin Header schematic

4.11 Test Headers

Some test pins have been reserved on the EBC7702 Development Board. The positions and functions are as shown in the table below.

Table 4-4 Test Headers

Header	Function Description
J11	MCU SWD signal, MCU flash is used
J14	D0/D1 SPI3 selects between D0 or D1 SPI3 by using D1_SPI3_CS_N. D0/D1 JTAG0 connected together in a daisy chain
J16	D0 EP,RC mode selection, RC:1<-->2,3<-->4; EP: 1<-->3
J19	D1 EP,RC mode selection, RC:1<-->2,3<-->4; EP: 1<-->3
J1	JTAG, SWD default switch selection

J45	1<-->2 auto Power on header
J42	1<-->2 cancel auto Power on header
J41	ATX CPU 12V power
J15	D0 Fan SOM board fan
J39	D1 Fan SOM board fan
J36	Chassis Fan1 Carrier board fan
J33	Chassis Fan2 Carrier board fan
J31	D1 I2C4
J37	D1 I2C3
J12	D0 PCIE RC,EP mode selection
J40	D1 PCIE RC,EP mode selection
J13	USB Boot configuration selection ; J13 jumper connection: USB0 enable; J13 jumper no connection: SPI enable
J18	D0 I2C3
J32	D0 I2C4

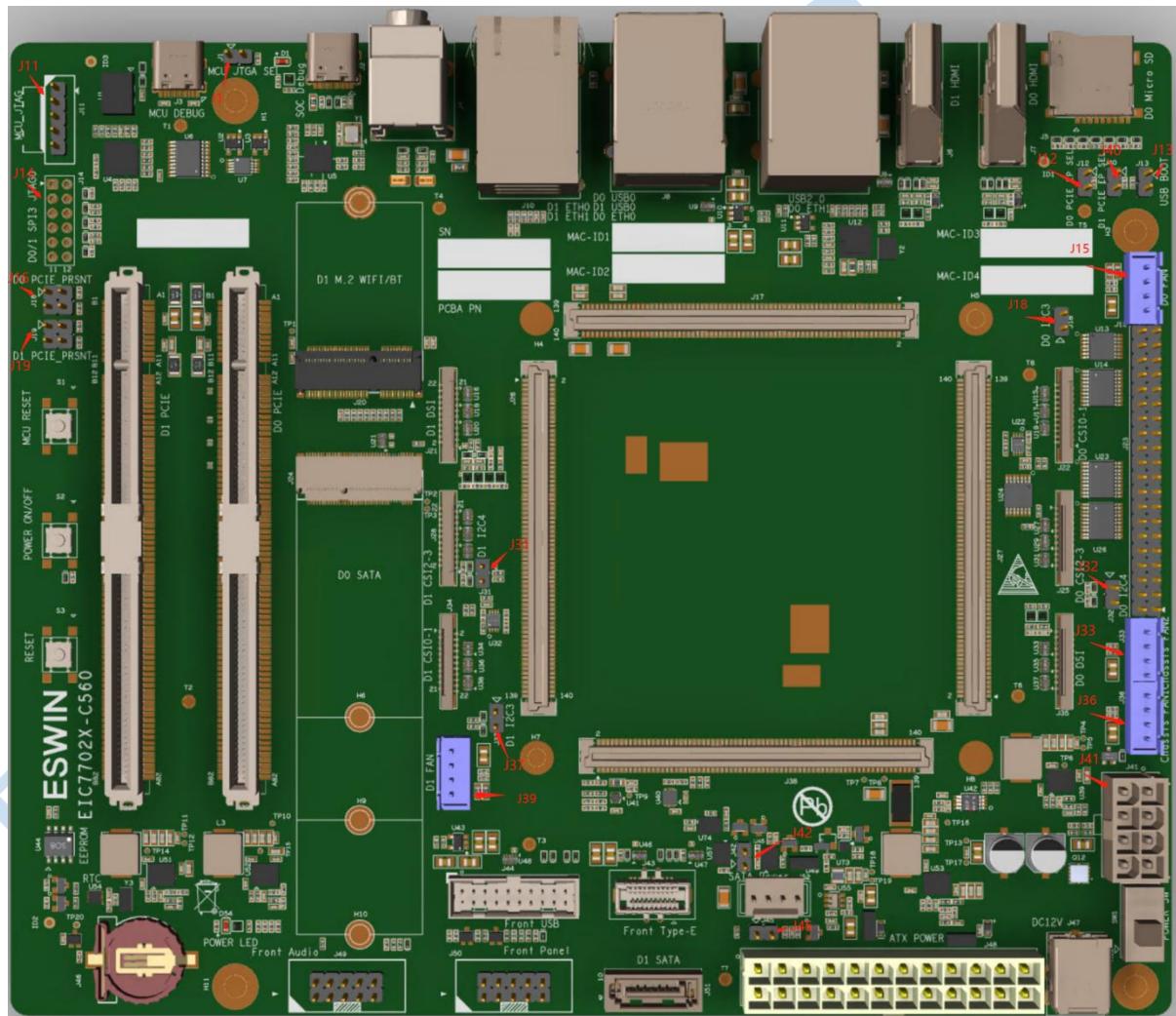
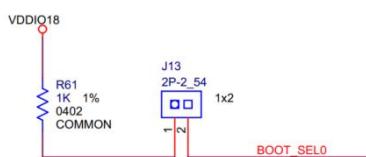
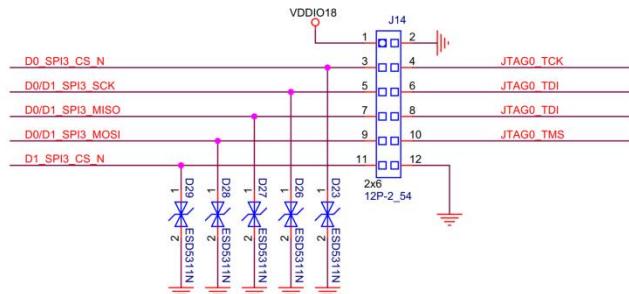
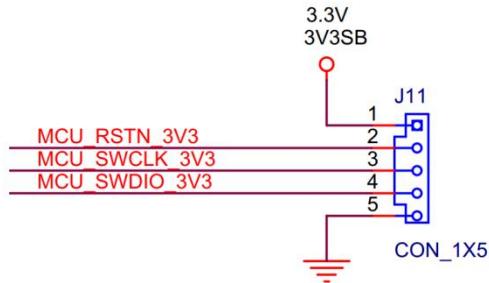


Figure 4-20 Test Pin



Without Jumper	SPI0(Default)
With Jumper	USB0 2.0

Figure 4-23 USB Boot Configure Header

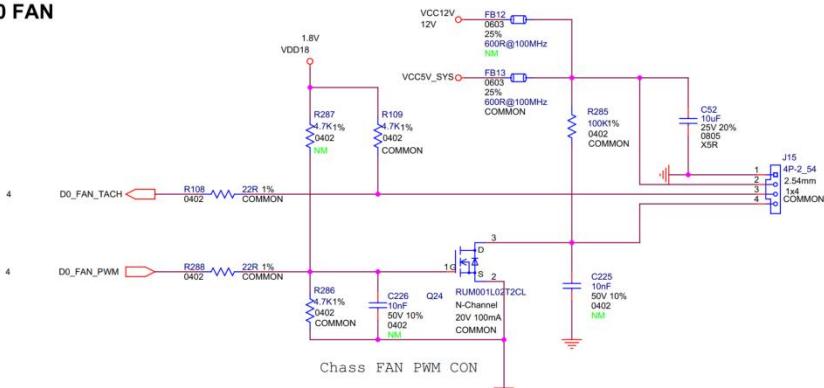
D0 FAN

Figure 4-24 J15 D0 FAN

D1 FAN

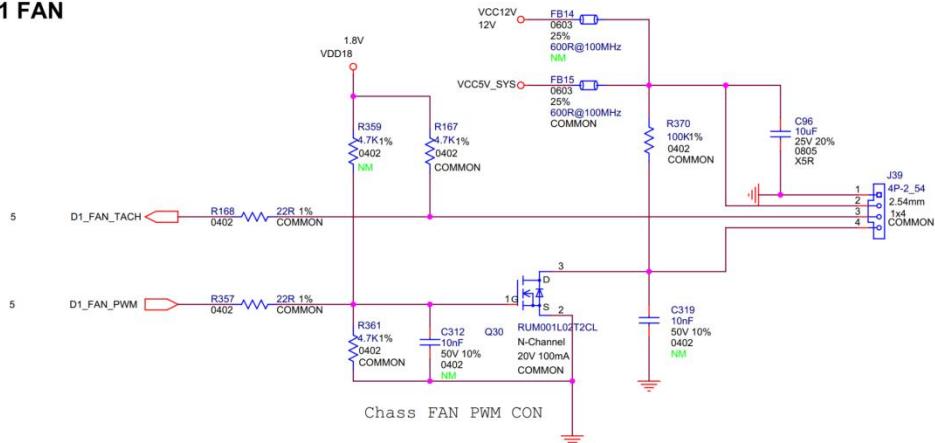


Figure 4-25 J39 D1 FAN

Chassis FAN1

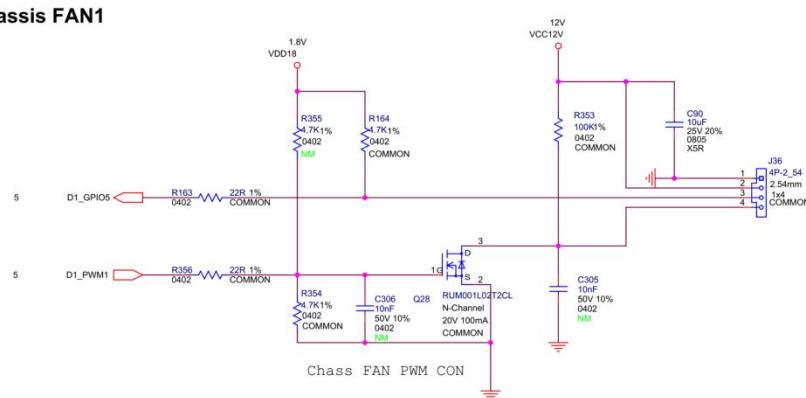


Figure 4-26 J36 Chassis FAN1

Chassis FAN2

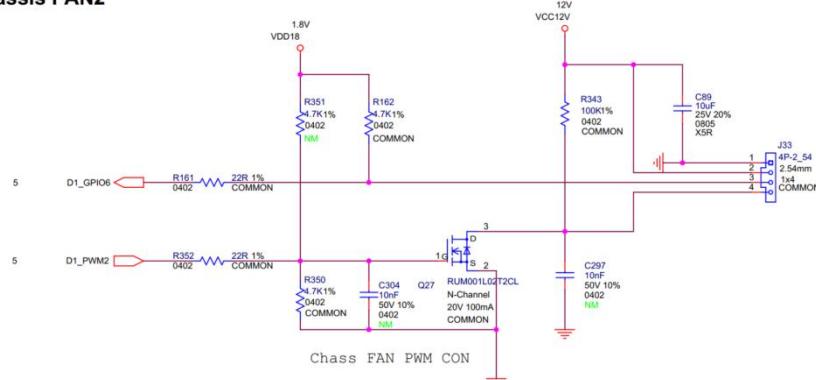


Figure 4-27 J33 Chassis FAN1

MUX

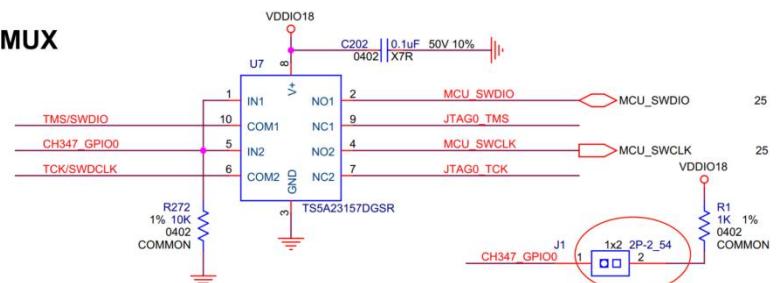


Figure 4-28 J1 JTAG0 SW default select Header

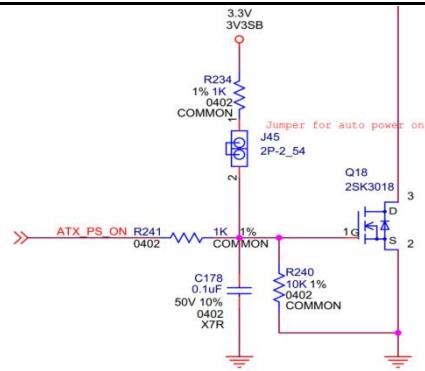


Figure 4-29 J45 Auto Power on Header

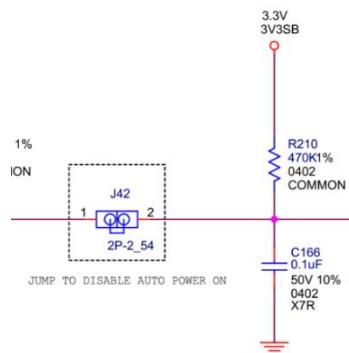


Figure 4-30 J42 Cancel Auto Power on Header

4.12 GND Test Pin

T1, T2, T3, T4, T5, T6, T7, T8 are GND test point

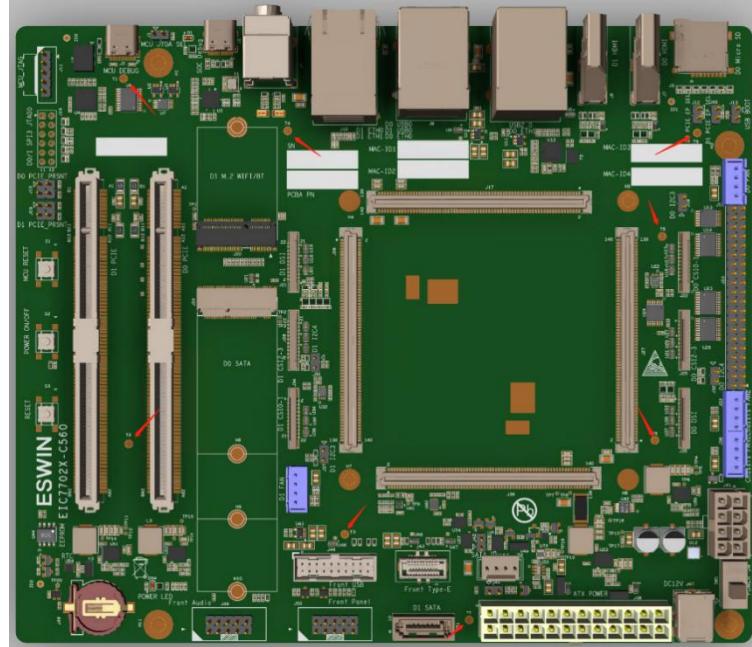


Figure 4-31 GND Test Pins

5 Software stack usage and board-level image Flashing

Users should refer to both the "Software Getting Started Guide" and the "Development Board Image Installation and Upgrade Manual" when using the EIC7x series AI SoC software stack.

When users are familiarizing themselves with the EIC7x software stack on the development board, they can directly use the "Software Getting Started Guide" for development, compilation, and related tasks. When users need to flash the images generated from development and compilation onto the development board, they should refer to the "EIC7700X Development Board Image Installation and Upgrade Manual".

6 Flashing Program and Board Information

6.1 MCU

The MCU program is a customer-customized program used for managing the development board's status, allowing customers to easily access or manage the operational status of the development board locally. When customers use the MCU to view and manage the EBC7702 development board's status, they can refer directly to the "EBC7702 Development Board MCU User Manual".

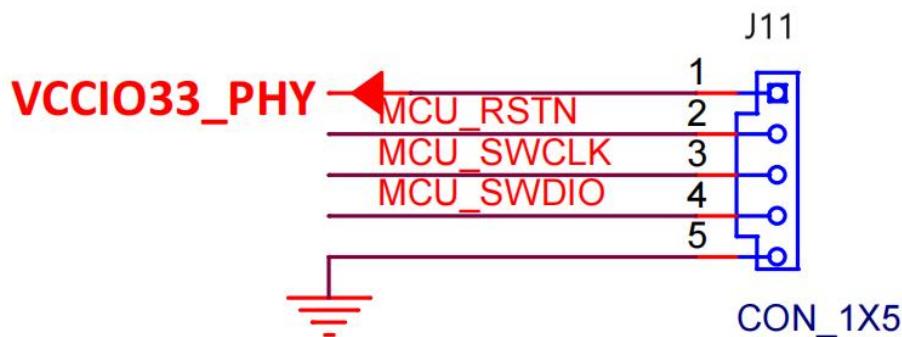


Figure 6-1 MCU_JTAG Wiring Reference

6.1.1 Flashing MCU Program on Linux

6.1.1.1 Environment Preparation - Installing stlink-tools

- 1) Port Connections
 - PortA: SOC UART1 connected to Type-C
 - PortB: MCU JTAG connected to ST-Link

Note 1: Since Die1 UART0 and the MCU debug UART share PortA, it is necessary to switch via shorting or disconnecting jumper J54 when in use.

- 2) Linux tool installation

```
sudo apt update
```

```
sudo apt install stlink-tools
```

- 3) Check the MCU Device

```
:$ st-info --probe
Found 1 stlink programmers
  serial: 373f71064e5734363f5e1143
  openocd: "\x37\x3f\x71\x06\x4e\x57\x34\x36\x3f\x5e\x11\x43"
    flash: 65536 (pagesize: 1024)
    sram: 20480
    chipid: 0x0410
    descr: F1 Medium-density device
```

Figure 6-2 The detected MCU device information

6.1.2 Flashing the MCU Program

1. Open the terminal and confirm the bin file: CV070_EC_FW_20241129_V1.bin
2. Execute: st-flash --reset write CV070_EC_FW_20241129_V1.bin 0x08000000

```
$ st-flash --reset write CV070_EC_FW_20241129_V1.bin 0x08000000
st-flash 1.6.0
2025-09-24T19:53:15 INFO common.c: Loading device parameters....
2025-09-24T19:53:15 INFO common.c: Device connected is: F1 Medium-density device, id 0x21050410
2025-09-24T19:53:15 INFO common.c: SRAM size: 0x5000 bytes (20 KiB), Flash: 0x10000 bytes (64 KiB) in pages of 1024 bytes
2025-09-24T19:53:15 INFO common.c: Attempting to write 15828 (0x3dd4) bytes to stm32 address: 134217728 (0x80000000)
Flash page at addr: 0x08003c00 erased
2025-09-24T19:53:15 INFO common.c: Finished erasing 16 pages of 1024 (0x400) bytes
2025-09-24T19:53:15 INFO common.c: Starting Flash write for VL/F0/F3/F1_XL core id
2025-09-24T19:53:15 INFO flash_loader.c: Successfully loaded flash loader in sram
16/16 pages written
2025-09-24T19:53:16 INFO common.c: Starting verification of write complete
2025-09-24T19:53:16 INFO common.c: Flash written and verified! jolly good!
```

Figure 6-3 MCU Successfully Flashed

6.1.3 Flashing MCU Program on Windows

6.1.3.1 Environment Preparation - Installing J-Flash

- 1) Port Connections
 - PortA : SOC UART1 connected to Type-C
 - PortB : MCU JTAG connected to J-Link
 - 2) Windows Software Installation
 - Official download link: <https://www.segger.com/downloads/jlink/>
1. Open the installer JLink_Windows_V872a_x86_64.exe, click Next -> I Agree.

2. Select all default options and complete the installation.
3. Open Device Manager. As shown in the red box, the JLink driver installation is successful.

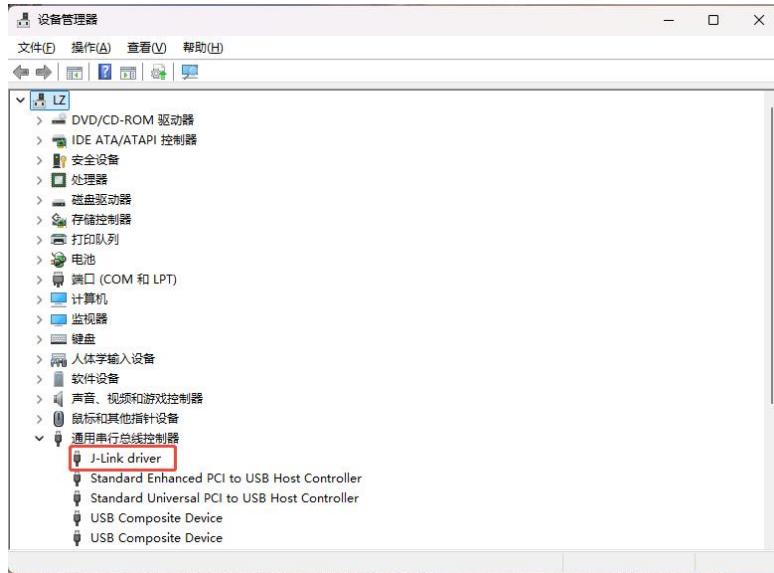


Figure 6-4 J-Link Device Driver

6.1.3.2 Flashing the MCU Program

1. Create a Project: Select Create new project, then click Start J-Flash.

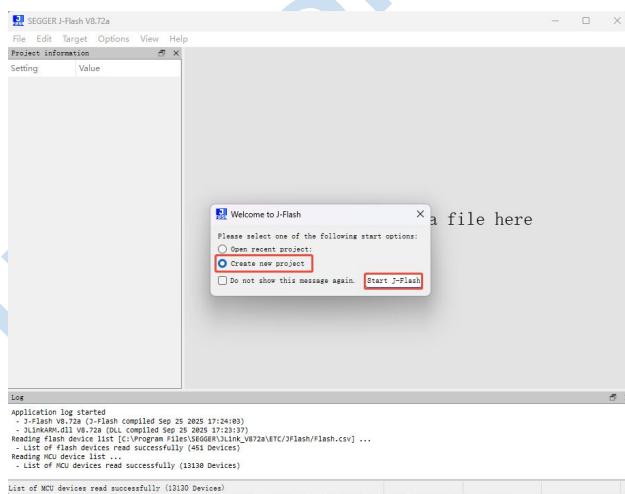


Figure 6-5 J-Flash start

2. Configure the Chip Model: Click the "... " button in Target device, then enter the chip model to be flashed in the Device field.

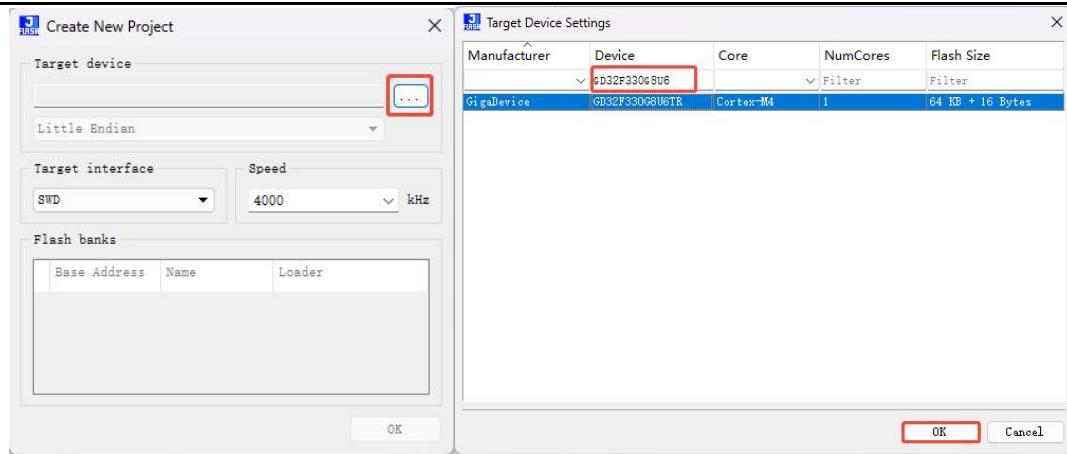


Figure 6-6 Configure the Chip Model

3. Verify Flash Address: Click OK to complete the project creation.

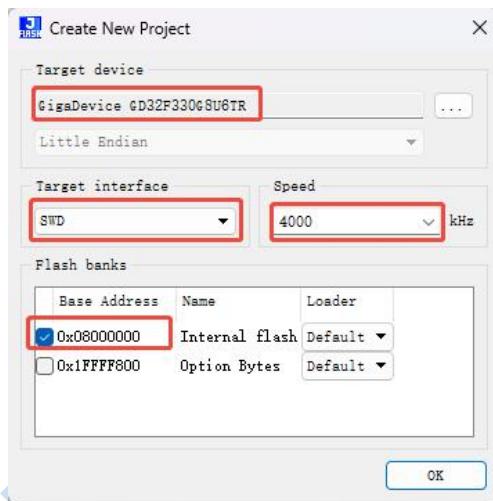


Figure 6-7 Verify Flash Address

4. Connect the Device: After setting up the hardware environment, click Target -> Connect.

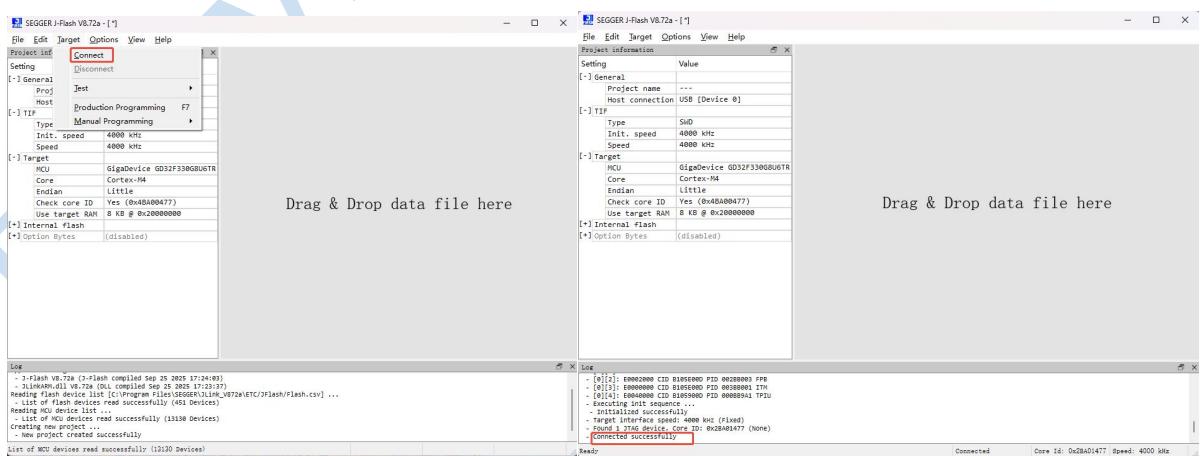


Figure 6-8 Connect the Device

5. Load the Bin File: Click File -> Open data File, locate and open the .bin or .hex file to be flashed.

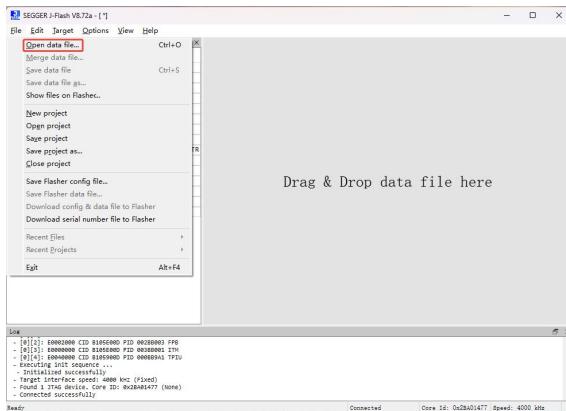


Figure 6-9 Open the File

6. Start Flashing: Click Target -> Production Programming to start the flashing program.

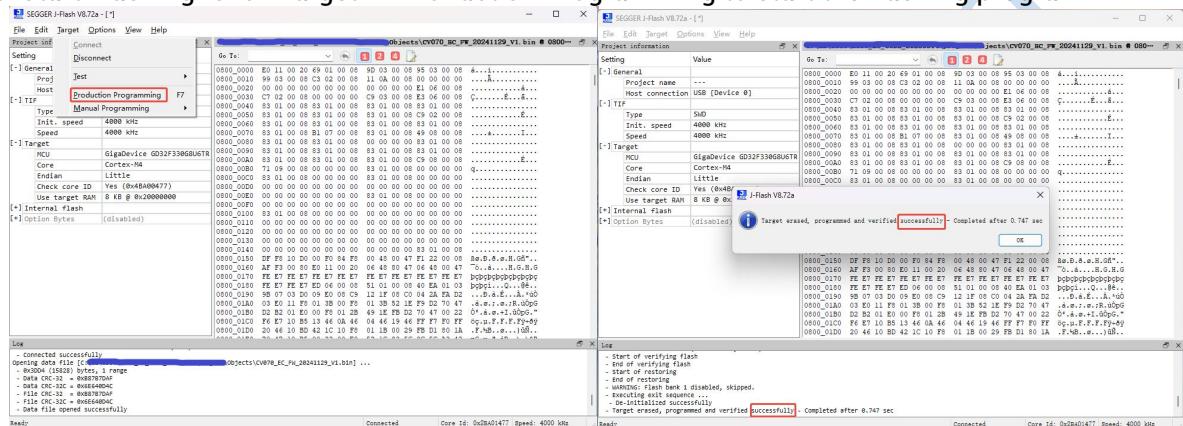


Figure 6-10 MCU Successfully Flashed

6.2 SOM

6.2.1 Development Board Flashing Configuration

When flashing the EBC7702 development board image, different peripheral interfaces are required depending on the flashing method. Users should ensure correct hardware connections as described in this section before use.

Table 6-1 Hardware Description

Number	Name	Description
23	Debug Uart	Used for Connecting EIC7702X Debug UART (Die0 UART0 & Die1 UART0), EIC7702X JTAG, I2C, SPI, MCU debug UART, MCU JTAG
20	USB3.0/ETH0	<ul style="list-style-type: none"> Die0 USB0 (Bottom Port): In USB boot mode, this acts as a device for loading the Die0 bootchain or as a device in Uboot for fastboot USB flashing. Die1 USB0 (Middle Port): In USB boot mode, this acts as a device for loading the Die0 bootchain or as a host in Uboot for USB device flashing. Die0 ETH0 (Top Port): Network interface, used in Uboot for fastboot via UDP to download image files.
19	USB2.0	USB Host: Used in Uboot for USB device flashing.

17	SD	SD Card Slot: Used for SD card flashing.
54	Bootsel	Used to select the boot mode for the EBC7702 development board. Detailed instructions can be found in BootSel and System Boot.

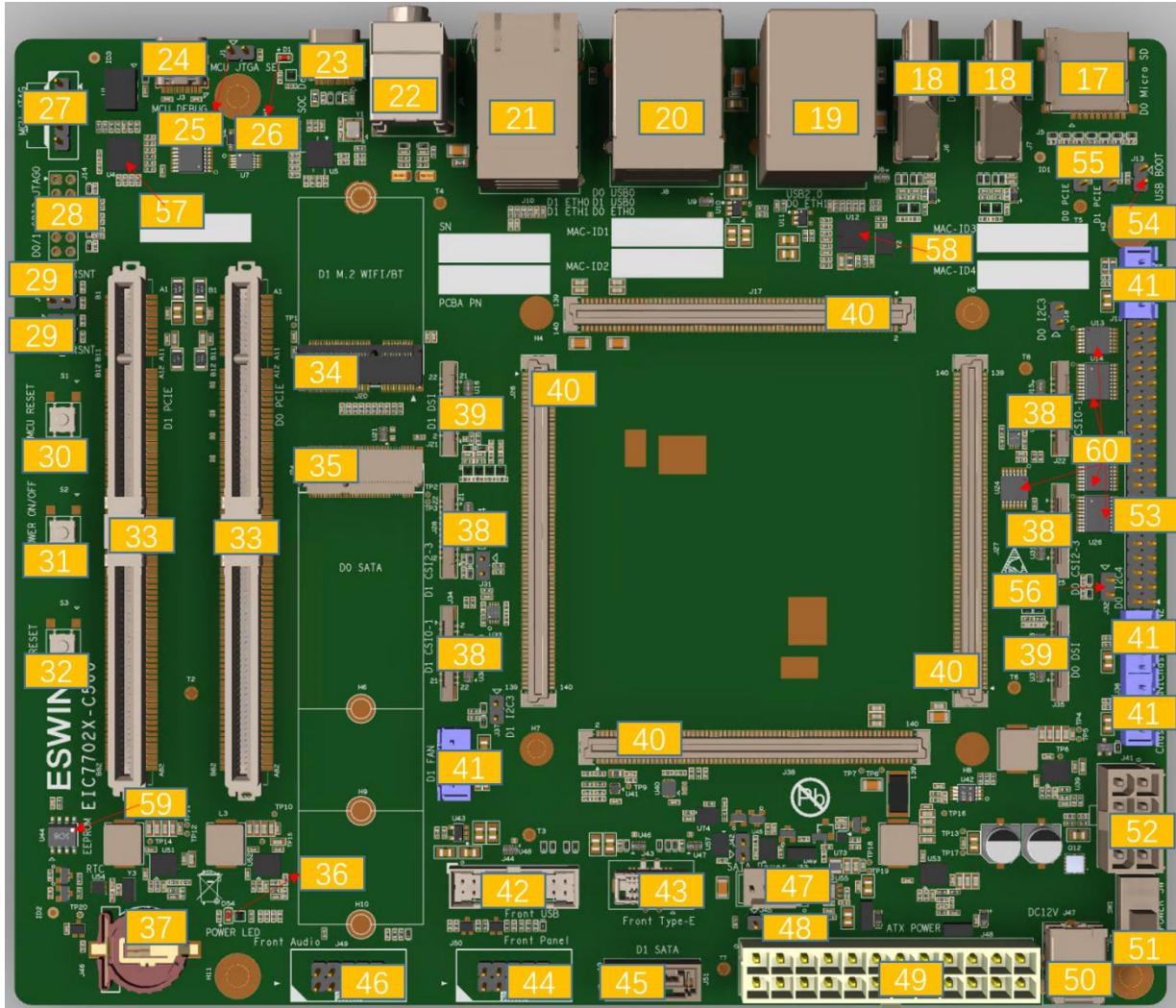


Figure 6-11 EBC7702 Development Board Hardware Description

6.2.1.1 Bootsel

Bootsel switch J13 (# 54), supports SPI and USB boot selection.

Table 6-2 Bootsel sw Signal List

Connection Status	Signal Definition	Notes
Not Shorted	Boot from SPI	
Shorted	Boot from USB	

6.2.1.2 System Boot

When performing image installation and upgrades, the boot mode corresponding to the J13(#54) pin header can be selected based on the figure below.

state	:	boot cpu	:	first boot	:	second boot
dis-connect	:	SCPU	:	ROM	:	SPI NOR
connect	:	SCPU	:	ROM	:	USB

Figure 6-12 BOOT SEL Configuration

6.2.2 Flashing the Image

6.2.2.1 Installing the BOOTLOADER Image

This section is applicable for the initial flashing or normal update of the bootloader.

6.2.2.1.1 Creating the Bootloader Flash Disk

To install or upgrade the image, the corresponding image file must be obtained from a USB flash drive. Therefore, the USB drive needs to be partitioned and formatted to either FAT or EXT4 file system in advance.

If the bootable USB drive has already been created, this step can be skipped.

6.2.2.1.2 Insert the USB drive into the host machine, in the terminal, enter `dmesg` command to obtain the device path of the USB drive.

```
53] usb 2-6: SerialNumber: ABCDEF0123456789AB
91] usb-storage 2-6:1.0: USB Mass Storage device detected
90] scsi host10: usb-storage 2-6:1.0
30] scsi 10:0:0:0: Direct-Access           SD Card Reader 1.00 PQ: 0 ANSI: 6
91] sd 10:0:0:0: Attached scsi generic sg0 type 0
53] sd 10:0:0:0: [sdc] 61074432 512-byte logical blocks: (31.3 GB/29.1 GiB)
99] sd 10:0:0:0: [sdc] Write Protect is off
02] sd 10:0:0:0: [sdc] Mode Sense: 03 00 00 00
45] sd 10:0:0:0: [sdc] No Caching mode page found
55] sd 10:0:0:0: [sdc] Assuming drive cache: write through
74] sdc: sdc1
75] sd 10:0:0:0: [sdc] Attached SCSI removable disk
```

Figure 6-13 Obtain the device path of the USB drive

6.2.2.1.3 Partition and format the USB drive, then create a partition table.

Use the following command:

```
$ sudo fdisk /dev/sdc
```

Welcome to fdisk (util-linux 2.31.1).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Command (m for help): g

Created a new GPT disklabel (GUID: 044AD8EA-A200-BE43-B53F-731998FD7092).

The old dos signature will be removed by a write command.

Command (m for help): n

Partition number (1-128, default 1): 1

First sector (2048-61074398, default 2048): 2048

Last sector, +sectors or +size{K,M,G,T,P} (2048-61074398, default 61074398): 61074398

Created a new partition 1 of type 'Linux filesystem' and of size 29.1 GiB.

Command (m for help): w

The partition table has been altered.

Calling ioctl() to re-read partition table.

Syncing disks.

The host terminal output is shown in the following figure:

```
$ sudo fdisk /dev/sdc

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): g
Created a new GPT disklabel (GUID: 044AD8EA-A200-BE43-B53F-731998FD7092).
The old dos signature will be removed by a write command.

Command (m for help): n
Partition number (1-128, default 1): 1
First sector (2048-61074398, default 2048): 2048
Last sector, +sectors or +size{K,M,G,T,P} (2048-61074398, default 61074398): 61074398

Created a new partition 1 of type 'Linux filesystem' and of size 29.1 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Figure 6-14 Create the Partition Table

6.2.2.1.4 Check if the partition was created successfully, and format partition number 1 as EXT4 file system

```
$ ls /dev/sdc*
/dev/sdc1

$ sudo mkfs.ext4 /dev/sdc1
Creating filesystem with 7680000 4k blocks and 1921360 inodes
Filesystem UUID: 40cae407-67bd-4474-858d-0ad0c07640c9
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

6.2.2.1.5 Copying the Flashing Image to the USB Drive

Copy the prepared bootloader image (e.g., nsign_bootloader_secboot_ddr.bin) to the USB drive. Once the copy is complete, safely eject the USB drive.

6.2.2.1.6 Partition Check and Settings

In Uboot, use the command mmc part to view the partition and edit partitions to modify the partition information. The command run gpt_partition to repartition(repartitioning is not required by default).

```

5 => mmc part
6
7 Partition Map for MMC device 0 -- Partition Type: EFI
8
9 Part      Start LBA        End LBA        Name
10
11       Attributes
12       Type GUID
13       Partition GUID
14 1 0x00000800      0x000327ff      "EFI System"
15     attrs: 0x0000000000000000
16     type: c12a7328-f81f-11d2-ba4b-00a0c93ec93b
17         (system)
18     guid: 8ce42ad9-d127-492c-b115-d4b13e7560fb
19 2 0x00032800      0x000347ff      "CIDATA"
20     attrs: 0x0000000000000000
21     type: 0fc63daf-8483-4772-8e79-3d69d8477de4
22         (linux)
23     guid: e19f7503-cbe0-47a6-908b-15e887e2a721
24 3 0x00034800      0x03a3ffde      "Root Partition"
25     attrs: 0x0000000000000000
26     type: 0fc63daf-8483-4772-8e79-3d69d8477de4
27         (linux)
28     guid: 19d29085-dcd1-44bc-8c63-e88c08d73564
29
30 => mode9end:2052ze 4 KiB, total 16 MiB

```

Figure 6-15 Edit Partition Information and Repartition in Uboot

Update the Uboot screenshot and confirm that the model is ESWIN EIC7702 EBC7702 Development Board.



Figure 6-16 Mode Confirmation

6.2.2.2 Upgrading the BOOTLOADER Image

This section applies to upgrading the bootloader image. The bootloader image can be upgraded via command line.

Note: Formatting and preparing the image USB drive can be referenced in the Creating the Bootloader Flash Disk section.

Connect the Debug UART and insert the image USB drive into either the USB2.0 port or Die1 USB0 (the upper port on USB 3.0). Power on the board and wait for it to enter the Uboot command-line interface (if you enter uboot and insert the USB drive again, you need to enter USB reset command to rescan the USB devices).

```

=>
=>
=> usb reset
resetting USB...
Bus usb1@50490000: Register 2000140 NbrPorts 2
Starting the controller
USB XHCI 1.10
Bus d1_usb0@70480000: Register 2000140 NbrPorts 2
Starting the controller
USB XHCI 1.10
Bus d1_usb1@70490000: Register 2000140 NbrPorts 2
Starting the controller
USB XHCI 1.10
scanning bus usb1@50490000 for devices... 2 USB Device(s) found
scanning bus d1_usb0@70480000 for devices... 2 USB Device(s) found
scanning bus d1_usb1@70490000 for devices... 1 USB Device(s) found
    scanning bus for storage devices... 1 Storage Device(s) found
=> ls usb 0
<DIR>      4096 .
<DIR>      4096 ..
<DIR>      16384 lost+found
      524288000 boot-EBC7702-D01-20250905-142731.ext4
      7516192768 root-EBC7702-D01-20250905-142731.ext4
      7771207168 d560-ubuntu-24.04-preinstalled-server-riscv64.img
<DIR>      4096 0804
<DIR>      4096 ubuntuversion
<DIR>      4096 json_outputs_C560_v2
=>

```

Figure 6-17 USB Rescan and Recognition

Mount the USB drive to download and install the image. Then switch to SPI boot mode.

Command1: env set stdout serial

The serial command directs the output to the serial port. After execution, system logs, command feedback, and other outputs will be sent through the serial port (which is convenient for monitoring via serial terminal tools on a computer).

Command2: ext4load usb 0 0x100000000 ubuntuversion/bootloader_EBC7702-D01_die0.bin

Load the bootloader file for die0 from the USB device's ext4 file system into the specified memory address

Command3: es_burn write 0x100000000 flash

Flash the data from the specified memory address into the flash storage

Command4: ext4load usb 0 0x100000000 ubuntuversion/bootloader_EBC7702-D01_die1.bin

Load the bootloader file for die1 from the USB device's ext4 file system into the specified memory address.

Command5: es_burn write 0x100000000 flash 1

Flash the data from the specified memory address into the flash storage.

```
setenv stdout serial
ext4load usb 0 0x100000000 ubuntuversion/bootloader_EBC7702-D01_die0.bin
5567208 bytes read in 69 ms (76.9 MiB/s)
es_burn write 0x100000000 flash
Bootspi flash write protection disabled
SF: 4096 bytes @ 0x1000 Read: OK
Erase progress: 100%:#####
SF: 4096 bytes @ 0x1000 Erased: OK
Write progress: 100%:#####
SF: 0x1000 bytes @ 0x1000 Written: OK
SF: 4096 bytes @ 0x0 Read: OK
FIRMWARE writing...
Erase progress: 100%:#####
SF: 2362 bytes @ 0x50000 Erased: OK
Write progress: 100%:#####
SF: 0x93a bytes @ 0x50000 Written: OK
DDR writing...
Erase progress: 100%:#####
SF: 709120 bytes @ 0x51000 Erased: OK
Write progress: 100%:#####
SF: 0xad200 bytes @ 0x51000 Written: OK
D2D writing...
Erase progress: 100%:#####
SF: 743852 bytes @ 0xff000 Erased: OK
Write progress: 100%:#####
SF: 0xb59ac bytes @ 0xff000 Written: OK
BOOTLOADER writing...
Erase progress: 100%:#####
SF: 4110792 bytes @ 0x1b5000 Erased: OK
Write progress: 100%:#####
SF: 0x3eb9c8 bytes @ 0x1b5000 Written: OK
BOOTCHAIN HEAD writing...
Erase progress: 100%:#####
SF: 4096 bytes @ 0x0 Erased: OK
Write progress: 100%:#####
SF: 0x1000 bytes @ 0x0 Written: OK
bootloader write OK
Bootspi flash write protection enabled
```

Figure 6-18 Die0 download and upgrade the bootloader image

```

es_burn write 0x10000000 flash <INTERRUPT>
=> <INTERRUPT>
=> ext4load usb 0 0x10000000 ubuntuversion/bootloader_EBC7702-D01_die1.bin
1447436 bytes read in 26 ms (58.1 MiB/s)
=> es_burn write 0x10000000 flash 1
Bootspi flash write protection disabled
SF: 4096 bytes @ 0x1000 Read: OK
Erase progress: 100%:+++++ooooooooooooooo
SF: 4096 bytes @ 0x1000 Erased: OK
Write progress: 100%:+++++ooooooooooooooo
SF: 0x1000 bytes @ 0x1000 Written: OK
SF: 4096 bytes @ 0x0 Read: OK
FIRMWARE writing...
Erase progress: 100%:+++++ooooooooooooooo
SF: 2418 bytes @ 0x50000 Erased: OK
Write progress: 100%:+++++ooooooooooooooo
SF: 0x972 bytes @ 0x50000 Written: OK
DDR writing...
Erase progress: 100%:+++++ooooooooooooooo
SF: 709120 bytes @ 0x51000 Erased: OK
Write progress: 100%:+++++ooooooooooooooo
SF: 0xad200 bytes @ 0x51000 Written: OK
D2D writing...
Erase progress: 100%:+++++ooooooooooooooo
SF: 735084 bytes @ 0xff000 Erased: OK
Write progress: 100%:+++++ooooooooooooooo
SF: 0xb376c bytes @ 0xff000 Written: OK
BOOTCHAIN HEAD writing...
Erase progress: 100%:+++++ooooooooooooooo
SF: 4096 bytes @ 0x0 Erased: OK
Write progress: 100%:+++++ooooooooooooooo
SF: 0x1000 bytes @ 0x0 Written: OK
bootloader write OK
Bootspi flash write protection enabled
=>

```

Figure 6-19 Die1 download and upgrade the bootloader image

6.2.2.2.1 Recover Mode Upgrade

This section applies to the initial flashing, normal update, or recovery of the bootloader image when the system fails to boot properly due to corruption.

Preparation Requirements: Development board, Two USB male-to-male cables, Jumper caps, Host machine (e.g., a Linux host)

As shown in the figure below, connect the development board's Debug UART to the host using a serial cable. Connect Die0 USB0 and Die1 USB0 of the development board to the host using USB male-to-male cable in sequence. Finally, short the Bootsel header pins (shorting means inserting a jumper cap as shown in the USB boot position).

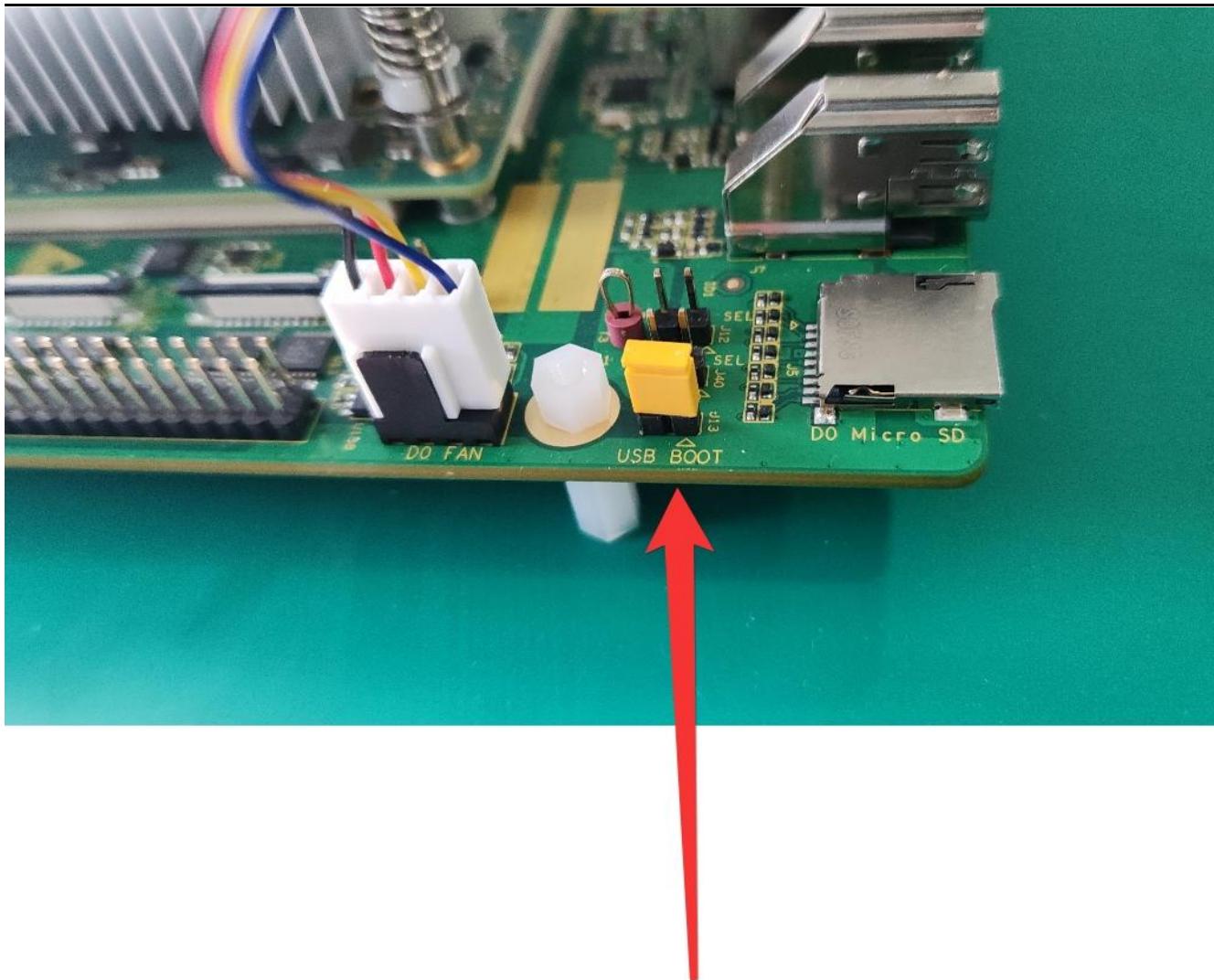


Figure 6-20 USB boot connection position

After powering on, the development board will create a USB storage device number in the host system (e.g., /dev/sd{x} and /dev/sd{y}). Use the dmesg command to obtain the related information (e.g., Die0 mounts to /dev/sda and Die1 mounts to /dev/sdb. For example, run sudo mount /dev/sda /home/eswin/sda. The purpose of this is to copy the recovery_bootloader_EBC7702-xxx_die0.bin file to die0 sda, and the bootloader_EBC7702-xxx_die1.bin file to die1 sdb).

```

usb 1-2-3: new high-speed USB device number 17 using xhci_hcd
usb 1-2-3: New USB device found, idVendor=0000, idProduct=0000, bcdDevice= 0.00
usb 1-2-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-2-3: Product: Eswin-storage
usb 1-2-3: Manufacturer: Eswincomputing.co
usb 1-2-3: SerialNumber:
usb-storage 1-2-3:1.0: USB Mass Storage device detected
scsi host6: usb-storage 1-2-3:1.0
scsi 6:0:0:0: Direct-Access      ESWIN      WIN-2030          PQ: 0 ANSI: 2
sd 6:0:0:0: Attached scsi generic sg1 type 0
sd 6:0:0:0: Power-on or device reset occurred
sd 6:0:0:0: [sdb] 8192000 512-byte logical blocks: (4.19 GB/3.91 GiB)
sd 6:0:0:0: [sdb] Write Protect is off
sd 6:0:0:0: [sdb] Mode Sense: 0c 00 00 08
sd 6:0:0:0: [sdb] No Caching mode page found ← D0 USB0
sd 6:0:0:0: [sdb] Assuming drive cache: write through
sd:
sd 6:0:0:0: [sdb] Attached SCSI removable disk
usb 1-2-4: new high-speed USB device number 18 using xhci_hcd
usb 1-2-4: New USB device found, idVendor=0000, idProduct=0000, bcdDevice= 0.00
usb 1-2-4: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-2-4: Product: Eswin-storage
usb 1-2-4: Manufacturer: Eswincomputing.co
usb 1-2-4: SerialNumber:
usb-storage 1-2-4:1.0: USB Mass Storage device detected
scsi host7: usb-storage 1-2-4:1.0
scsi 7:0:0:0: Direct-Access      ESWIN      WIN-2030          PQ: 0 ANSI: 2
sd 7:0:0:0: Attached scsi generic sg2 type 0
sd 7:0:0:0: Power-on or device reset occurred
sd 7:0:0:0: [sdc] 8192000 512-byte logical blocks: (4.19 GB/3.91 GiB)
sd 7:0:0:0: [sdc] Write Protect is off
sd 7:0:0:0: [sdc] Mode Sense: 0c 00 00 08
sd 7:0:0:0: [sdc] No Caching mode page found ← D1 USB0
sd 7:0:0:0: [sdc] Assuming drive cache: write through
sd:
sd 7:0:0:0: [sdc] Attached SCSI removable disk

```

Figure 6-21 Board Corresponding Block Device Number

Note that Die0 and Die1 require different bootloaders to be flashed. The USB 3.0 upper port corresponds to Die1, while the USB 3.0 lower port corresponds to Die0.



Figure 6-22 Distinction between die0 and die1 interfaces

In the release image package, find the files `recovery_bootloader_EBC7702-xxx_die0.bin` and `recovery_bootloader_EBC7702-xxx_die1.bin`. Mount both Die0 and Die1 USB devices, then copy the recovery bootloader images to the corresponding directories for Die0 and Die1. After the copying is complete, the bootchain flashing will be successful. (If the D0/D1 HDMI outputs are connected to monitors, the message Upgrade Finished will appear on the screen.)

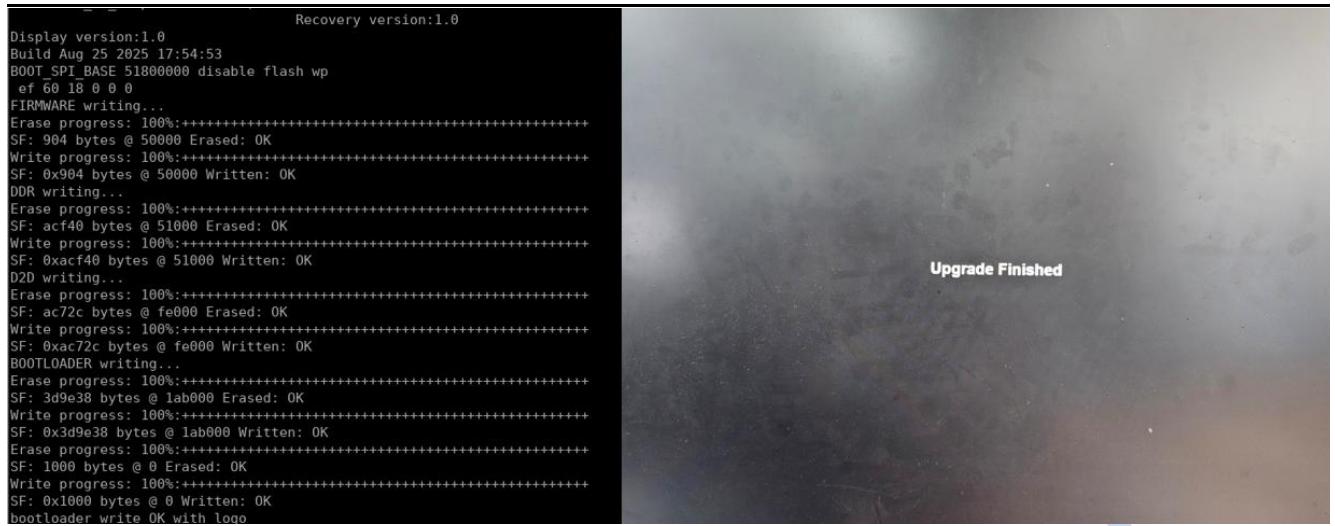


Figure 6-22 Die0 is flashed using the recovery file corresponding to its display

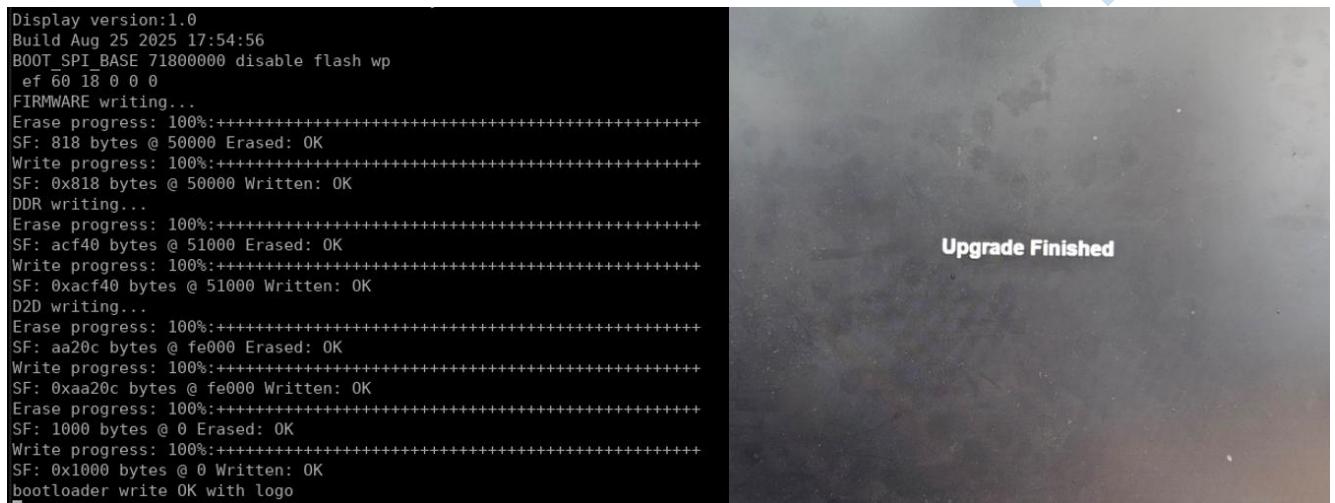


Figure 6-24 Die1 is flashed using the recovery file corresponding to its display

6.2.2.3 System Image Upgrade

This section mainly covers how to upgrade the boot and root file system images. The upgrade can be done via USB, allowing for the upgrade of both the boot and root file system images.

This section is only applicable when the bootloader can start normally, and the goal is to upgrade the boot and root file system images.

6.2.2.3.1 USB Upgrade

After the development board boots up, enter the command run gpt_partition in U-Boot, followed by the flashing command (usb 0 reads the pre-copied system from the USB drive. If a directory exists, for example, a folder named `ubantuversion/` in the USB drive, modify the command to: `es_fs write usb 0 ubantuversion/EBC7702-ubuntu-24.04-preinstalled-server-riscv64.img mmc 0`).

Flash Command: `es_fs write usb 0 EBC7702-ubuntu-24.04-preinstalled-server-riscv64.img mmc 0`

```

Erasing SPI flash...Writing to SPI flash...done
OK
=> es_fs write usb 0 ubantuversion/d560-ubuntu-24.04-preinstalled-server-riscv64.img mmc 0
Write progress: 100%:+++++=====
ubantuversion/d560-ubuntu-24.04-preinstalled-server-riscv64.img has been successfully written in mmc 0
=> boot

```

Figure 6-25 Start flashing the Ubuntu version

After inputting the command reset, the device will boot up.

```

=> es_fs write usb 0 ubantuversion/d560-ubuntu-24.04-preinstalled-server-riscv64.img mmc 0
Write progress: 100%:+++++=====
ubantuversion/d560-ubuntu-24.04-preinstalled-server-riscv64.img has been success
=> ly written in mmc 0
=> D2D NOTICE: ΔT 2.067°C (prev 41.535, curr 43.602)
=>
=> reset
resetting ...
eic770x_cold_reset
Secboot Firmware Version: 1.4
ESWIN SOM560 board
Set max SOC voltage
Power on fan
pll config ok
die_num:2,die_ordinal:0
Firmware version:1.24; disable ECC
Display version:1.0
boot start temperature:42.916C
DDR type:LPDDR5_D2D_X16;Size:16GB,Data Rate:6400MT/s
DDR self test OK
D2D Firmware Version: 1.48 auto 7
d2d_pmix_init!elp | 115200 8N1 | NOR | Minicom 2.8 | VT102 | Offline | ttyACM10
T: degree 41.996
d2d_pmix_need_sweep: cur_degree: 41.996 min_degree: 38.047 fitted_min_degree: 28d2d_pmix_need_sweep: sweep_full_smooth: 0
d2d_pmix_need_sweep: sweep_full_smooth: 0
d2d_pmix_need_sweep: sweep_full: 0
d2d_pmix_need_sweep: sweep_full: 0
d2d_pmix_need_sweep: sweep_incr: 0
d2d_pmix_need_sweep: sweep_incr: 0
===== d2d start      0ms      0ms =====
===== d2d init       4ms      4ms =====
T: degree 41.535
[Thermal] Cur: 41.535 Target: 28.500 Heat: 50.000 Cool: 60.000
[Thermal] Cur <= Heat, FAN[off] NPU[off]
d2d_init: retry cnt 0
d2d_init: Deassert [default subsys] rst 0x80
d2d_init: Aclk full speed
d2d_init: Deassert [prst|cfg] rst

```

Figure 6-26 Device boots up

6.2.2.3.2 Firmware and Software Upgrade Operations

To upgrade the firmware and software, insert the USB drive into either the development board's USB2.0 port or the Die1 USB0 (the upper port on USB 3.0), and proceed with the upgrade after entering the uboot command line.

- 1) Insert the USB drive: Insert the prepared USB drive into the development board's USB2.0 port or Die1 USB0 as shown in the figure below.
- 2) Power Up for Upgrade: Power on the development board and enter uboot to perform the upgrade by typing the following command:

setenv stdout serial ---- This command ensures that no characters are lost during copy-paste.

```

Erasing SPI flash...Writing to SPI flash...done
OK
=> es_fs write usb 0 ubantuversion/d560-ubuntu-24.04-preinstalled-server-riscv64.img mmc 0
Write progress: 100%:+++++=====
ubantuversion/d560-ubuntu-24.04-preinstalled-server-riscv64.img has been successfully written in mmc 0
=> boot

```

Figure 6-27 Ubuntu version upgrade

The following screenshot shows the process of updating the bootloader.

```
=> setenv stdout serial
=> ext4load usb 0 0x10000000 ubuntuversion/bootloader_EBC7702-D01_die0.bin
5567208 bytes read in 69 ms (76.9 MiB/s)
=> es_burn write 0x10000000 flash
Bootspi flash write protection disabled
SF: 4096 bytes @ 0x1000 Read: OK
Erase progress: 100%:+++++
SF: 4096 bytes @ 0x1000 Erased: OK
Write progress: 100%:+++++
SF: 0x1000 bytes @ 0x1000 Written: OK
SF: 4096 bytes @ 0x0 Read: OK
FIRMWARE writing...
Erase progress: 100%:+++++
SF: 2362 bytes @ 0x50000 Erased: OK
Write progress: 100%:+++++
SF: 0x93a bytes @ 0x50000 Written: OK
DDR writing...
Erase progress: 100%:+++++
SF: 709120 bytes @ 0x51000 Erased: OK
Write progress: 100%:+++++
SF: 0xad200 bytes @ 0x51000 Written: OK
D2D writing...
Erase progress: 100%:+++++
SF: 743852 bytes @ 0xff000 Erased: OK
Write progress: 100%:+++++
SF: 0xb59ac bytes @ 0xff000 Written: OK
BOOTLOADER writing...
Erase progress: 100%:+++++
SF: 4110792 bytes @ 0x1b5000 Erased: OK
Write progress: 100%:+++++
SF: 0x3eb9c8 bytes @ 0x1b5000 Written: OK
BOOTCHAIN HEAD writing...
Erase progress: 100%:+++++
SF: 4096 bytes @ 0x0 Erased: OK
Write progress: 100%:+++++
SF: 0x1000 bytes @ 0x0 Written: OK
bootloader write OK
Bootspi flash write protection enabled
```

Figure 6-28 Die0 bootloader upgrade

```

es_burn write 0x100000000 flash <INTERRUPT>
=> <INTERRUPT>
=> ext4load usb 0 0x100000000 ubuntuversion/bootloader_EBC7702-D01_die1.bin
1147436 bytes read in 26 ms (58.1 MiB/s)
=> es_burn write 0x100000000 flash 1
Bootspi flash write protection disabled
SF: 4096 bytes @ 0x1000 Read: OK
Erase progress: 100%:+++++ooooooooooooooo
SF: 4096 bytes @ 0x1000 Erased: OK
Write progress: 100%:+++++ooooooooooooooo
SF: 0x1000 bytes @ 0x1000 Written: OK
SF: 4096 bytes @ 0x0 Read: OK
FIRMWARE writing...
Erase progress: 100%:+++++ooooooooooooooo
SF: 2418 bytes @ 0x50000 Erased: OK
Write progress: 100%:+++++ooooooooooooooo
SF: 0x972 bytes @ 0x50000 Written: OK
DDR writing...
Erase progress: 100%:+++++ooooooooooooooo
SF: 709120 bytes @ 0x51000 Erased: OK
Write progress: 100%:+++++ooooooooooooooo
SF: 0xad200 bytes @ 0x51000 Written: OK
D2D writing...
Erase progress: 100%:+++++ooooooooooooooo
SF: 735084 bytes @ 0xff000 Erased: OK
Write progress: 100%:+++++ooooooooooooooo
SF: 0xb376c bytes @ 0xff000 Written: OK
BOOTCHAIN HEAD writing...
Erase progress: 100%:+++++ooooooooooooooo
SF: 4096 bytes @ 0x0 Erased: OK
Write progress: 100%:+++++ooooooooooooooo
SF: 0x1000 bytes @ 0x0 Written: OK
bootloader write OK
Bootspi flash write protection enabled
=>

```

Figure 6-29 Die1 bootloader upgrade

6.2.2.4 D2D Phase Adjustment (D2D pmix)

When the device boots up after flashing, it will undergo phase adjustment.

Principle Overview:

D2D phase adjustment data is used to optimize the D2D transmission function. The phase adjustment data is stored at a fixed location in the bootspi flash and must be generated through a phase adjustment program.

After the adjustment data is generated, during normal system operation, firmware running on the low power CPU (lpcpu) will periodically read the phase adjustment data from bootspi flash and configure the D2D link based on the current environment temperature to achieve optimal transmission performance. (The phase adjustment has already been completed for new boards. However, updating the bootloader might cause a shift in the phase adjustment storage location, necessitating a re-adjustment to suit the current temperature range. The minimum temperature for phase adjustment is 40 ° C. If the startup temperature is below 30 ° C, incremental phase adjustment will be triggered, which will require a temperature rise. However, only a temperature increase from 30° C to 40° C is required.)

EBC77 Series Development Board User Manual

```
d2d_ber_test_and_sync: Local BER test: PASS (total errors: 0x0)
d2d_ber_test_and_sync: Dual-die BER test success!
d2d initCheck pass
sweep target degree 85, pid_control 1, fan_polarity 0, incremental 0, ber_time: Setting NPU system voltage to 750mV (reg config: eswin som568)
NPU system voltage set successfully to 748mV
CPU CLK SAVE: offset 0x208 (0x51828208) 0xff800000 -> 0xff800002, CPU clk 24M
Fan set to 100% PWM with polarity 0
Sweep required start temperature: 40.000°C, Curr Die: 41.535°C, need cooling!
Sweep required start temperature: 40.000°C, Curr Die: 40.842°C, need cooling!
Sweep required start temperature: 40.000°C, Curr Die: 40.147°C, need cooling!
Sweep required start temperature: 40.000°C, Curr Die: 39.682°C, need cooling!
Curr die cooled below target (40.000°C). Waiting for another die
Both dies finished cooling. Good!
CPU CLK RESTORE: offset 0x208 (0x51828208) 0xff800002 -> 0xff800000, CPU clk 140
Temperature: 0x13a(38.516), Valid: 1, ΔT: 65650.252
Stored in pmix_data[0]:
    one_test Temperature: 0x13a(38.516), Valid: 1
    PMIX[0]: Phase0=0x5c, Phase90=0x5b, Phase180=0x5c, Phase270=0x5b, width=      PMIX[1]: Phase0=0x3a, Phase90=0x3a, Phase180=0x3a, Phase270=0x3a, width=      PMIX[2]
2]: Phase0=0x5d, Phase90=0x5d, Phase180=0x5d, Phase270=0x5d, width=      PMIX[3]: Phase0=0x3c, Phase90=0x3c, Phase180=0x3c, Phase270=0x3d, width=      PMIX[4]: Phase0=0x3
54, Phase90=0x54, Phase180=0x55, Phase270=0x56, width=      PMIX[5]: Phase0=0x36, Phase90=0x36, Phase180=0x36, Phase270=0x36, width=      PMIX[6]: Phase0=0x35, Phase90=0x3
x36, Phase180=0x33, Phase270=0x34, width=      PMIX[7]: Phase0=0x39, Phase90=0x38, Phase180=0x37, Phase270=0x39, width=d2d_set_target_temperature 452, target_temperature 4
0.016
d2d_set_target_temperature 457, npu_task_cnt: 0
elapsed time: 0 hours 0 minutes 7 seconds, halt notify 0
```

Figure 6-30 If the phase adjustment temperature exceeds 40° C, cooling is required first.

6.2.2.4.1 Specific phase adjustment method

The uboot has been updated to a version that supports phase adjustment functionality. The latest bootloader versions support automatic phase adjustment.

After the device restarts, press p quickly to print the previous phase adjustment data. For the EBC7702 development board, phase adjustment requires temperature rise (temperature rise is crucial); If the ambient temperature is too low, the heating process will be very slow. The target temperature for phase adjustment is from the current temperature to 90° C. To speed up the process, you can cover the device, as the fan will remain off during the heating process.)

```
NPU system voltage set successfully to 748mV
CPU CLK SAVE: offset 0x208 (0x51828208) 0xff800000 -> 0xff800002, CPU clk 24M
Fan set to 100% PWM with polarity 0
Sweep required start temperature: 40.000°C, Curr Die: 41.535°C, need cooling!
Sweep required start temperature: 40.000°C, Curr Die: 40.842°C, need cooling!
Sweep required start temperature: 40.000°C, Curr Die: 40.147°C, need cooling!
Sweep required start temperature: 40.000°C, Curr Die: 39.682°C, need cooling!
Curr die cooled below target (40.000°C). Waiting for another die
Both dies finished cooling. Good!
CPU CLK RESTORE: offset 0x208 (0x51828208) 0xff800002 -> 0xff800000, CPU clk 140
Temperature: 0x13a(38.516), Valid: 1, ΔT: 65650.252
Stored in pmix_data[0]:
    one_test Temperature: 0x13a(38.516), Valid: 1
    PMIX[0]: Phase0=0x5c, Phase90=0x5b, Phase180=0x5c, Phase270=0x5b, width=      PMIX[1]: Phase0=0x3a, Phase90=0x3a, Phase180=0x3a, Phase270=0x3a, width=      PMIX[2]
2]: Phase0=0x5d, Phase90=0x5d, Phase180=0x5d, Phase270=0x5d, width=      PMIX[3]: Phase0=0x3c, Phase90=0x3c, Phase180=0x3c, Phase270=0x3d, width=      PMIX[4]: Phase0=0x3
54, Phase90=0x54, Phase180=0x55, Phase270=0x56, width=      PMIX[5]: Phase0=0x36, Phase90=0x36, Phase180=0x36, Phase270=0x36, width=      PMIX[6]: Phase0=0x35, Phase90=0x3
x36, Phase180=0x33, Phase270=0x34, width=      PMIX[7]: Phase0=0x39, Phase90=0x38, Phase180=0x37, Phase270=0x39, width=d2d_set_target_temperature 452, target_temperature 4
0.016
d2d_set_target_temperature 457, npu_task_cnt: 0
elapsed time: 0 hours 0 minutes 7 seconds, halt notify 0

Temperature: 0x138(38.047), Valid: 1, ΔT: 0.469
Stored in pmix_data[1]:
    one_test Temperature: 0x138(38.047), Valid: 1
    PMIX[0]: Phase0=0x5c, Phase90=0x5b, Phase180=0x5c, Phase270=0x5b, width=      PMIX[1]: Phase0=0x3a, Phase90=0x3a, Phase180=0x3a, Phase270=0x3b, width=      PMIX[2]
2]: Phase0=0x5d, Phase90=0x5d, Phase180=0x5d, Phase270=0x5d, width=      PMIX[3]: Phase0=0x3c, Phase90=0x3c, Phase180=0x3c, Phase270=0x3d, width=      PMIX[4]: Phase0=0x3
55, Phase90=0x54, Phase180=0x56, Phase270=0x55, width=      PMIX[5]: Phase0=0x36, Phase90=0x37, Phase180=0x37, Phase270=0x35, width=      PMIX[6]: Phase0=0x35, Phase90=0x3
x37, Phase180=0x34, Phase270=0x36, width=      PMIX[7]: Phase0=0x39, Phase90=0x38, Phase180=0x37, Phase270=0x39, width=d2d_set_target_temperature 452, target_temperature 3
0.547
d2d_set_target_temperature 457, npu_task_cnt: 0
elapsed time: 0 hours 0 minutes 15 seconds, halt notify 0

Temperature: 0x13b(38.749), Valid: 1, ΔT: 0.702
Stored in pmix_data[2]:
    one_test Temperature: 0x13b(38.749), Valid: 1
    PMIX[0]: Phase0=0x5c, Phase90=0x5b, Phase180=0x5c, Phase270=0x5b, width=      PMIX[1]: Phase0=0x3a, Phase90=0x3a, Phase180=0x3a, Phase270=0x3a, width=      PMIX[2]
2]: Phase0=0x5d, Phase90=0x5d, Phase180=0x5d, Phase270=0x5d, width=      PMIX[3]: Phase0=0x3c, Phase90=0x3d, Phase180=0x3d, Phase270=0x3d, width=      PMIX[4]: Phase0=0x3
55, Phase90=0x54, Phase180=0x56, Phase270=0x55, width=      PMIX[5]: Phase0=0x36, Phase90=0x37, Phase180=0x36, Phase270=0x35, width=      PMIX[6]: Phase0=0x35, Phase90=0x3
x36, Phase180=0x33, Phase270=0x35, width=      PMIX[7]: Phase0=0x39, Phase90=0x38, Phase180=0x36, Phase270=0x39, width=d2d_set_target_temperature 452, target_temperature 4
0.249
```

Figure 6-31 Phase adjustment in progress

Figure 6-31

6.2.2.5 Installing Serial Drivers in Windows

1. On Windows, you may need to install the serial driver program CH347 SER.EXE.

2. Use a serial cable to reconnect the development board to the computer.

3. Use MobaXterm software to open the serial terminal.

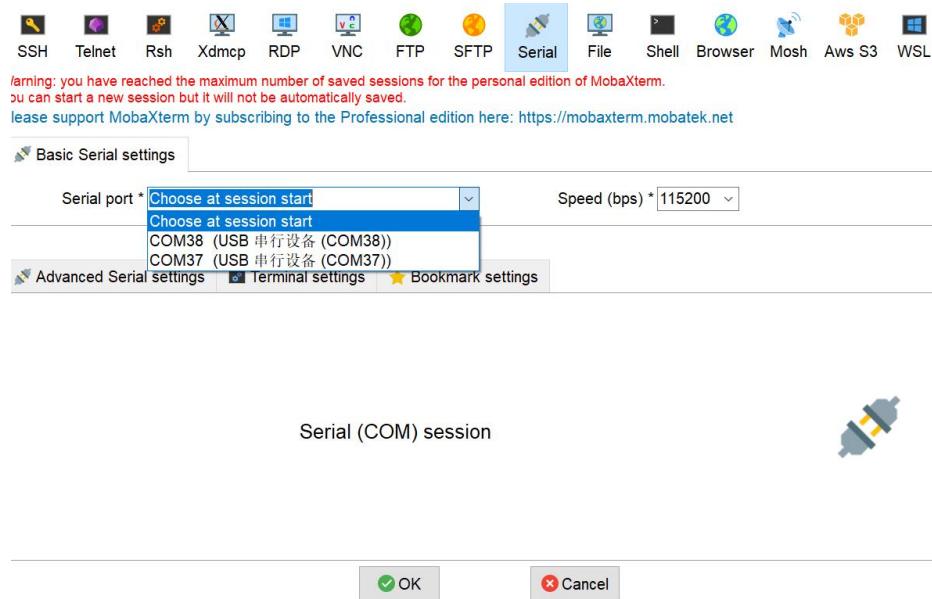


Figure 6-32 Screenshot of the Serial Terminal Tool MobaXterm

4. Once connected to the serial terminal, follow the previous steps to operate and use the development board.

6.2.2.6 FQA

Q: How to become familiar with uboot related functions?

A: Use the help command to query uboot commands, and each command will also provide detailed help information.

Q: What should I do if the network is not working?

A: Ensure the network has the necessary permissions. Then, use print to check if the ethaddr parameter is set and if the MAC address is read correctly. If not, the MAC address can be configured using the command env set ethaddr aa:bb:cc:11:22:33, followed by the saveenv command to persist the setting.

For the EBC7702 development board, there are four MAC address settings. (Below is an example, you need to input your own MAC address).

```
env set eth1addr 94:47:B0:18:5C:XX
env set eth2addr 94:47:B0:18:5C:XX
env set eth3addr 94:47:B0:18:5C:XX
env set ethaddr 94:47:B0:18:5C:XX
```

The following configurations must also be set:

```
setenv fdtfile 'eswin/eic7702-EBC7702.dtb'
env save
```

Q: Why does serial port pasting only accept a fixed length, losing subsequent characters?

A: Use the command setenv stdout serial. This will stop uboot from outputting to HDMI.

Q: I flashed it, but the device still shows an error during startup?

A: The environment configuration from the previous version may have been left behind, affecting the

startup of the new version. After flashing the bootloader, input reset, then run env default -a to clear the old configurations. Use saveenv to save the current configuration, but any manually added settings from the previous version will be lost. (This applies after startup.)

Q: The flash process gives an error like "partition does not found" or "partition is too small"?

A: Use the command mmc part to check partition information. First, flash the correct bootloader, then reset and run env default -a to clear old configurations. Use saveenv to save the current configuration, then reset and run run gpt_partition to repartition. After running Env default, check the print command to verify the dtb informations. If missing, set it with setenv fdtfile 'eswin/eic7702-EBC7702.dtb' and then run env save.

Q: How can I repartition if the default partition size is not enough?

A: Use the print command to view the partitions parameter. Use the edit partitions command to modify it, or env set partitions to reset it. Afterward, run gpt_partition to repartition.

Q: How can I confirm if the flash was successful?

A: During uboot startup, there will be a line indicating the uboot compilation time. Similarly, the kernel startup will also display the kernel compilation time.

Q: Both SD and eMMC have systems, how do I make the system boot from the SD card?

A: We use UUIDs to identify the file system partitions. To ensure booting from the SD card, you need to erase the eMMC data with the following command: mmc erase 0 0xa00000. If you need to erase the SD card data, switch the MMC device to dev 1 in advance with mmc dev 1.

Q: Does uboot support hot-swapping of devices?

A: USB devices require a usb reset after hot-swapping. SD cards do not support hot-swapping.