User's Manual

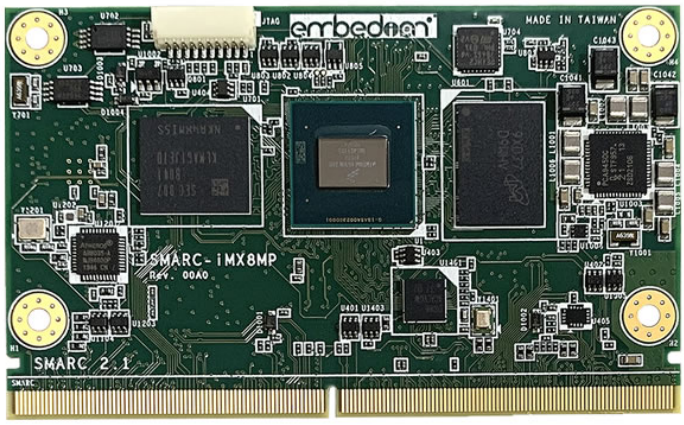
SMARC Computer on Module

NXP *i.MX8M Plus* 4/2 x Cortex-A53 and Cortex-M7
Neural Processing Unit (NPU) operating at up to 2.3 TOPS
1 x 24bits dual-channel LVDS LCD
HDMI 2.0a
MIPI-DSI
4 x COM Ports
1 x SDHC
1 x USB OTG 2.0, 1 x USB 3.0, 4 x USB Host 2.0
2 x 10/100/1000M Gigabit Ethernet with TSN
2 x CAN-FD, 2 x SPIs, 4 x I2Cs, PCIe 3.0,
2 x MIPI_CSI and 2 x ISP

SMARC-iMX8P

Quad and Dual Cores with NPU

(SMARC 2.0 Specification Compliant)


embedian
MADE IN TAIWAN
SMARC- iMX8MP
Rev. 00A0
SMARC 2.1

Revision History

Revision	Date	Changes from Previous Revision
1.0	2021/ 07/ 20	Initial Release

USER INFORMATION

About This Manual

This document provides information about products from EMBEDIAN, INC. No warranty of suitability, purpose, or fitness is implied. While every attempt has been made to ensure that the information in this document is accurate, the information contained within is supplied “as-is” and is subject to change without notice.

For the circuits, descriptions and tables indicated, EMBEDIAN assumes no responsibility as far as patents or other rights of third parties are concerned.

Copyright Notice

Trademarks

The following lists the trademarks of components used in this board.

- ARM is a registered trademark of ARM Limited.
- Android is a registered trademark of Google
- Linux is a registered trademark of Linus Torvalds.
- WinCE is a registered trademark of Microsoft
- Qualcomm is a registered trademark of Qualcomm
- All other products and trademarks mentioned in this manual are trademarks of their respective owners.

Standards

EMBEDIAN is ISO 9001:2008 and ISO14001-certified manufacturer. SMARC is an SGET standard for ARM computer on module.

Warranty

This EMBEDIAN product is warranted against defects in material and workmanship for the warranty period from the date of shipment. During the warranty period, *EMBEDIAN* will at its discretion, decide to repair or replace defective products.

Within the warranty period, the repair of products is free of charge as long as warranty conditions are observed.

The warranty does not apply to defects resulting from improper or inadequate maintenance or handling by the buyer, unauthorized modification or misuse, operation outside of the product's environmental specifications or improper installation or maintenance.

EMBEDIAN will not be responsible for any defects or damages to other products not supplied by *EMBEDIAN* that are caused by a faulty *EMBEDIAN* product.

Technical Support

Technicians and engineers from *EMBEDIAN* and/or its subsidiaries and official distributors are available for technical support. We are committed to making our product easy to use and will help you use our products in your systems.

Before contacting EMBEDIAN technical support, please consult our Web site for the latest product documentation, utilities, and drivers. If the information does not help solve the problem, contact us by e-mail or telephone.

Table of Contents

Using this Manual

This guide provides information about the Embedian *SMARC-iMX8MP* for NXP *i.MX8M Plus* embedded *SMARC* core module family.

Conventions used in this guide

This table describes the typographic conventions used in this guide:

This Convention	Is used for
Italic type	Emphasis, new terms, variables, and document titles.
monospaced type	Filenames, pathnames, and code examples.

Embedian Information

Document Updates

Please always check the product specific section on the Embedian support website at www.embedian.com/ for the most current revision of this document.

Contact Information

For more information about your Embedian products, or for customer service and technical support, contact Embedian directly.

To contact Embedian by	Use
Mail	Embedian, Inc. 9F-4. 432 Keelung Rd. Sec. 1, Taipei 11051, Taiwan
World Wide Web	http://www.embedian.com/
Telephone	+ 886 2 2722 3291

Additional Resources

Please also refer to the most recent *NXP i.MX8M Plus* processor reference manual and related documentation for additional information.

Introduction

This Chapter gives background information on the *SMARC-iMX8MP*
Section include：

- Features and Functionality
- Module Variant
- Differences between Module Variants
- Block diagram
- Software Support / Hardware Abstraction
- Module Variant
- Document and Standard References

Chapter 1 Introduction

The *SMARC-iMX8MP* enables system integrators and manufacturers to create and deploy advanced and power-efficient solutions at the edge. It provides high performance and processing speed, while keeping the power consumption to a minimum.

Based on *NXP i.MX8M Plus quad and dual* cores, the *SMARC-iMX8MP* carries quad/dual 1.8GHz *ARM Cortex-A53* and 800MHz real-time *Cortex-M7* co-processor. The built-in *NPU* (up to 2.3TOPs) provides customers the option to perform machine learning inference directly on the edge, reducing cloud dependency, latency and is able to perform highly complex neural network functions like face and emotion detection, object detection and surveillance, which are key to applications such as smart and safe cities, retail, smart home and much more.

Featuring additional high-speed interfaces such as 2 x Gigabit Ethernet, TSN, PCIe Gen 3, USB 3.0, LVDS, HDMI, MIPI-DSI and CAN-FD capabilities, offering industrial temperature grade and NXP's longevity program, the i.MX8M Plus SMARC 2.0 module is perfect for industrial IoT and HMI applications.

The module is the ideal choice for a broad range of target markets including

- Industrial
  - Anesthesia Unit Monitor
  - Anomaly Detection
  - Hospital Admission Machine
  - Electricity Grid and Distribution
  - Factory Automation
  - Powered Patient Beds
  - Avionics
- Smart City
  - Fleet Analytics and Driver Monitors
  - Safety, Security, and Surveillance
  - Automatic Vehicle Identification
- Smart Home
  - Home Patient and Elderly Monitors
  - Home Security and Surveillance
- And more

Complete and cost-efficient Embedian evaluation kits for Yocto build, Debian bullseye, Ubuntu 18.04 and Android 11 allow immediate and professional

embedded product development with dramatically reduced design risk and time-to-market.

1.1 Features and Functionality

The *SMARC-iMX8MP* module is based on the *i.MX8M Plus* processor with quad and dual cores from NXP. This processor offers a high number of interfaces. The module has the following features:

- *SMARC* 2.0 compliant in an 82mm x 50mm form factor.
- *NXP i.MX8M* Plus Processor:
  - *Dual/Quad x 1.8GHz ARM Cortex™-A53*
  - *Real-time 800Mhz ARM Cortex™-M7*
  - Neon Media Processor Engine (*MPE*)
  - AI/ML NPU 2.3 TOPS
  - 2D/3D GPU GC7000UL/ GC520L
- Memory: Onboard 16GB *eMMC* Flash
- Onboard 4GB or 6GB *LPDDR4*
- TPM 2.0
- Networking: 2 x 10/100/1000 Mbps Ethernet (1 Gbit Ethernet QoS with TSN supports)
- Display:
  - One 24-bit Single/Dual channel *LVDS* up to 1920x1200p60 if no more than 2 display instances used
  - *HDMI* 2.0a (up to 3840 x 2160p30)
  - *MIPI-DSI* (up to 1920x1440 at 60 Hz)
- Expansion: 1 x *SDHC/SDIO*, 5x *USB* 2.0 (one OTG), 2 x *USB* 3.0, 1 x *PCIe* x 1 Gen 3.0
- USB: 4 x *USB* 2.0 Host, 1 x *USB* 2.0 OTG, 2 x *USB* 3.0
- A single 4KB *EEPROM* is provided on I2C1 that holds the board information. This information includes board name, serial number, and revision information.
- Additional Interface:
  - 4 x UARTs
  - 2 x SPI
  - 5 x I2C
  - 2 x I2S
  - 2 x CAN-FD[Note1]
  - 2 x PWM
  - 1 x 4-Lane MIPI CSI and 1 x 2-Lane MIPI CSI (Camera Interface)
  - 12 x GPIOs

  - WDT
- SW Support: Linux, Yocto Build, Ubuntu 18.04, Debian 10, Android 10
- Power Consumption (Typcal)
  - TBD
- Thermal:
  - Commercial Temperature: 0°C ~ 80°C
  - Industrial Temperature: -40° ~85°C
- Power Supply
- 3V to 5.25V
- 1.8V module IO support (SMARC 2.0 compliant)

Note:

1. CAN-FD if processor is industrial temperature one. CAN bus if processor is commercial temperature one.

1.2 Module Variant

The *SMARC-iMX8MP* module is available with various options based on processors in this family from *NXP, LPDDR4* memory configuration, and operating temperature ranges.

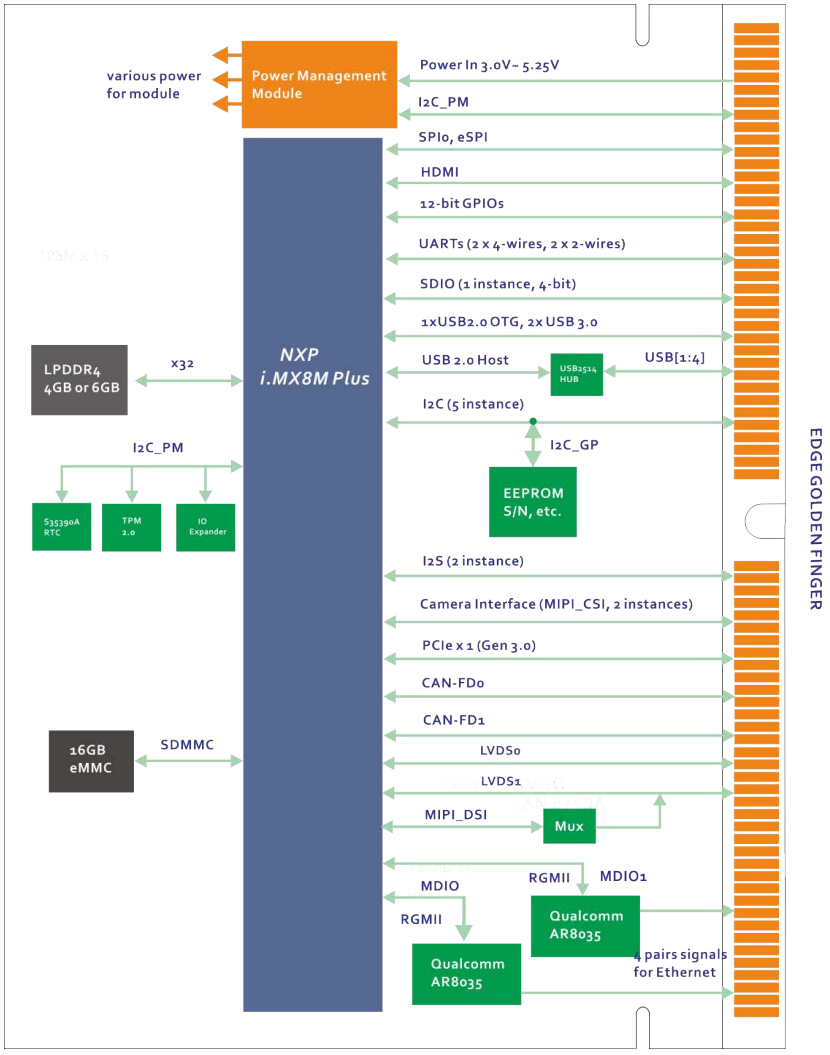
1. "Q" (CPU is Quad Core with NPU)
   "D" (CPU is Dual Core with NPU)
2. "4G" (4GB LPDDR4 memory)
   "6G" (6GB LPDDR4 memory)
3. "*I*" Industrial temperature - Leave it blank if commercial temperature.
4. "C" (Conformal coating) – Leave it blank if no needs of conformal coating.

For example, *SMARC-iMX8MP-Q-4G* stands for Quad core *i.MX8M Plus* processor and *4GB LPDDR4* memory in normal operating temperature without conformal coating.

1.3 Block Diagram

The following diagram illustrates the system organization of the *SMARC-iMX8MP*. Arrows indicate direction of control and not necessarily signal flow.

Figure 1 SMARC-iMX8MP Block Diagram



Details for this diagram will be explained in the following chapters.

1.4 Software Support / Hardware Abstraction

The Embedian *SMARC-iMX8MP* Module is supported by Embedian BSPs (Board Support Package). The first *SMARC-iMX8MP* BSP targets Linux (Ubuntu 18.04 LTS, Debian Bulleye, Yocto Build) and Android 11 support. BSPs for other operating systems are planned. Check with your Embedian contact for the latest BSPs.

This manual goes into a lot of detail on I/O particulars – information is provided on exactly how the various *SMARC* edge fingers tie into the NXP *i.MX8M Plus* SoC and to other Module hardware. This is provided for reference and context. Almost all of the I/O particulars are covered and abstracted in the BSP and it should generally not be necessary for users to deal with I/O at the register level.

1.5 Document and Standard References

1.5.1. External Industry Standard Documents

- ***eMMC (Embedded Multi-Media Card)*** the eMMC electrical standard is defined by JEDEC JESD84-B45 and the mechanical standard by JESD84-C44 *(www.jedec.org)*.
- ***The I2C Specification,*** Version 2.1, January 2000, Philips Semiconductor (now NXP) *(www.nxp.com)*.
- ***I2S Bus Specification,*** Feb. 1986 and Revised June 5, 1996, Philips Semiconductor (now NXP) (www.nxp.com).
- ***JTAG (Joint Test Action Group*** defined by IEEE 1149.1-2001 - IEEE Standard Test Access Port and Boundary Scan Architecture (www.ieee.org).
- ***MXM3 Graphics Module Mobile PCI Express Module Electromechanical Specification,*** Version 3.0, Revision 1.1, © 2009 NVIDIA Corporation (www.mxm-sig.org).
- ***PICMG® EEEP Embedded EEPROM Specification,*** Rev. 1.0, August 2010 (www.picmg.org).
- ***SD Specifications Part 1 Physical Layer Simplified Specification,*** Version 3.01, May 18, 2010, © 2010 SD Group and SD Card Association (Secure Digital) (www.sdcard.org).
- ***SPI Bus –*** "Serial Peripheral Interface" - de-facto serial interface standard defined by Motorola. A good description may be found on Wikipedia (http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus).
- ***USB Specifications*** (www.usb.org).
- ***PCI Express Specifications (www.pci-sig.org)***
- ***SPDIF (aka S/PDIF) ("Sony Philips Digital Interface)-*** *IEC 60958-3*
- ***eSPI ("Enhanced Serial Peripheral Interface") T****he eSPI Interface Base Specification is defined by Intel https://downloadcenter.intel.com/de/download/22112)*
- ***GBE MDI ("Gigabit Ethernet Medium Dependent Interface")*** *defined by IEEE 802.3. The 1000Base-T operation over copper twisted pair cabling defined by IEEE 802.3ab (www.ieee.org).*
- ***RS-232 (EIA "Recommended Standard 232")*** *this standard for asynchronous serial port data exchange dates from 1962. The original standard is hard to find. Many good descriptions of the standard can be*

found on-line, e.g. at Wikipedia, and in text books.

- ***CSI-2 (Camera Serial Interface version 2)*** *The CSI-2 standard is owned and maintained by the MIPI Alliance ("Mobile Industry Processor Interface Alliance") (www.mipi.org).*
- ***CSI-3 (Camera Serial Interface version 3)*** *The CSI-3 standard is owned and maintained by the MIPI Alliance ("Mobile Industry Processor Alliance") (www.mipi.org)*
- ***CAN FD ("Controller Area Network Flexible Data-Rate")*** *Bus Standard – ISO 11898-1*
- ***DisplayPort and Embedded DisplayPort*** *These standards are owned and maintained by VESA ("Video Electronics Standards Association") (www.vesa.org)*

1.5.2. SGET Documents

- ***SMARC_Hardware_Specification_V200***, version 2.0, June 2nd, 2016.
- ***SMARC_Hardware_Specification_V1p1***, version 1.1, May 29, 2014.

1.5.3. Embedian Documents

The following documents are listed for reference. The Module schematic is not usually available outside of Embedian, without special permission. The other schematics will be available. Contact your Embedian representative for more information. The *SMARC* Evaluation Carrier Board Schematic is particularly useful as an example of the implementation of various interfaces on a Carrier board.

- ***SMARC Evaluation Carrier Board Schematic***, PDF and OrCAD format
- ***SMARC Evaluation Carrier Board User's Manual***
- ***SMARC-iMX8MP User's Manual***
- ***PinMux file for SMARC-iMX8MP***
- ***SMARC-iMX8MP Schematic Checklist***

1.5.4. NXP Documents

- ***IMX8MPRM, i.MX 8M Plus Applications Processor Reference Manual,*** *11/2021 (rev. 1)*
- ***IMX8MPIEC, i.MX 8M Plus Applications Processor Datasheet for Industrial Products,*** *03/2021 (rev. 1)*
- ***IMX8MPCEC, i.MX 8M Plus Applications Processor Datasheet for Consumer Products,*** *03/2021 (rev. 1)*

1.5.5. NXP Development Tools

- ***IOMUX_TOOL v9 for ARM® i.MX8M Plus Microprocessors***

1.5.6. NXP Software Documents

- ***Linux lf-5.10.y_1.0.0***
- ***Android 11.0.0_1.2.1 Documentation***

1.5.7. Embedian Software Documents

- ***Embedian Linux BSP for SMARC-iMX8MP Module***
- ***Embedian Android BSP for SMARC-iMX8MP Module***
- ***Embedian Linux BSP User's Guide***
- ***Embedian Android BSP User's Guide***

1.5.8. NXP Design Network

- ***SABRE***
- ***Wandboard***
- ***Nucleus***
- ***QNX***

Specifications

This Chapter provides *SMARC-iMX8MP* specifications.
Section include：

- *SMARC-iMX8MP* General Functions
- *SMARC-iMX8MP* Debug
- Mechanical Specifications
- Electrical Specification
- Environment Specification

Chapter 2 Specifications

2.1 SMARC-iMX8MP General Functions

2.1.1. SMARC-iMX8MP Feature Set

This section lists the complete feature set supported by the *SMARC-iMX8MP* module.

SMARC Feature Specification	SMARC 2.0 Specification Maximum Number Possible	SMARC-iMX8MP Feature Support	SMARC-iMX8MP Feature Support Instances
LVDS LCD Display Support	2	Yes	2(dual channel)
DP/eDP	1	Yes	N/A
HDMI Display Support	1	Yes	1
Serial Camera Support	2	Yes	2 (1 x 4-lane and 1 x 2-lane)
USB Interface	6	Yes	5 (1 x USB 2.0 OTG, 4 x USB 2.0, 2 x USB 3.0)
PCIe Interface	4	Yes	1 (1-lane Gen 3.0)
SATA Interface	1	N/A	N/A
GbE Interface	1	Yes	1
2nd GBE Interface	1	Yes	1
SDIO Interface (4bit)	1	Yes	1
SPI Interface	2	Yes	2
I2S Interface	2	Yes	2
I2C Interface	6	Yes	5
Serial	4	Yes	4

SMARC Feature Specification	SMARC 2.0 Specification Maximum Number Possible	SMARC-iMX8MP Feature Support	SMARC-iMX8MP Feature Support Instances
CAN	2	Yes	2 (CAN-FD)[Note1]
VDDIO	1.8V	1.8V	1.8V

Note:

1. CAN-FD if processor is industrial temperature. CAN bus if processor is commercial temperature.

2.1.2. Form Factor

The *SMARC-iMX8MP* module complies with the *SMARC* General Specification module size requirements in an 82mm x 50mm form factor.

2.1.3. CPU

The *SMARC-iMX8MP* implements *NXP's i.MX8M ARM Cortex-A53* and *ARM* Cortex-M7 processor.

NXP CPU	i.MX8M Plus
ARM Cortex-A53 cores	*4/2 x 1.8GHz Cortex-A53*
ARM Cortex-M7 cores	*1x 800Mhz Cortex-M7*
Memory Speed	*LPDDR4-4000* *Inline ECC on the DDR bus*
L2 Cache	*512KB L2*
GPU	• *GC7000UL with OpenCL and Vulkan support* • *2 shader* • *166 million triangles/sec* • *1.0 giga pixel/sec* • *16 GFLOPs 32-bit* • *Supports OpenGL ES 1.1, 2.0, 3.0, OpenCL 1.2, Vulkan* • *Core clock frequency of 1000 MHz* • *Shader clock frequency of 1000 MHz* • *GC520L for 2D acceleration* • *Render target compatibility between 3D and 2D GPU (super tile status buffer)*

NXP CPU	i.MX8M Plus
VPU	*Video Decode* *• 1080p60 HEVC/H.265 Main, Main 10 (up to level 5.1)* *• 1080p60 VP9 Profile 0, 2* *• 1080p60 VP8* *• 1080p60 AVC/H.264 Baseline, Main, High decoder* *Video Encode* *• 1080p60 AVC/H.264 encoder* *• 1080p60 HEVC/H.265 encoder*
NPU	*2.3 TOP/s Neural Network performance* *• Keyword detect, noise reduction, beamforming* *• Speech recognition (i.e. Deep Speech 2)* *• Image recognition (i.e. ResNet-50)*

2.1.4. Onboard Storage

The *SMARC-iMX8MP* module supports a 16GB *eMMC* flash memory device, and a 32Kb I2C serial *EEPROM* on the Module *I2C_GP* (I2C3) bus. The device used is an On Semiconductor 24C32 equivalent. The Module serial EEPROM is intended to retain Module parameter information, including a module part number, revision number and serial number. The Module serial EEPROM data structure conforms to the PICMG® EEEP Embedded *EEPROM* Specification.). The onboard 32GB *eMMC* flash is used as boot media and operating systems. The module will always boot up from the onboard *eMMC* flash first. The firmware in *eMMC* flash will read the *BOOT_SEL* configuration from the boot selection and boot up the devices from that selected.

2.1.5. Clocks

A 24 MHz oscillator is used as the primary clock source for the PLLs to generate the clock for CPU, BUS, and high-speed interfaces. For fractional PLLs, the 24 MHz clock from the oscillator can be directly used as the PLL reference clock.
A 32.768 KHz clock is required for the *i.MX8M Plus* CPU RTC (Real Time Clock) and external (S-35390A) RTC.
A 24Mhz crystal is used on on-module *USB2514* USB hub.
A 25 MHz *HCSL* oscillator is used as the reference clock for *PCIe clock generator*.
The Qualcomm AR8035 PHY, *PCIe HCSL* clock generator is provided with a 25 MHz clock using a crystal in normal oscillation mode.

2.1.6 LVDS Interface

The *SMARC-iMX8MP* implements one 24 bit dual channel *LVDS* output streams.

The LVDS Display Bridge (*LDB*) from the NXP® *i.MX8M Plus* processor found on the *SMARC-iMX8MP* offers two *LVDS* channels, with up to 1920x1200 @60Hz[Note]. Each channel consists of one clock pair and four data pairs.

The LVDS ports support the following configurations:

- Two single channel output
- One dual channel output

Note:

There are three *LCFIF* controllers in *i.MX8M Plus* processor. Support up to 1920x1200p60 display per LCDIF if no more than 2 instances used simultaneously, or 1x 1080p60 + 2x 720p60 if all 3 instances used simultaneously.

• One *LCDIF* drives *MIPI DSI*
• One *LCDIF* drives *LVDS Tx*
• One *LCDIF* drives *HDMI*

The following figure shows the *LVDS LCD* block diagram.

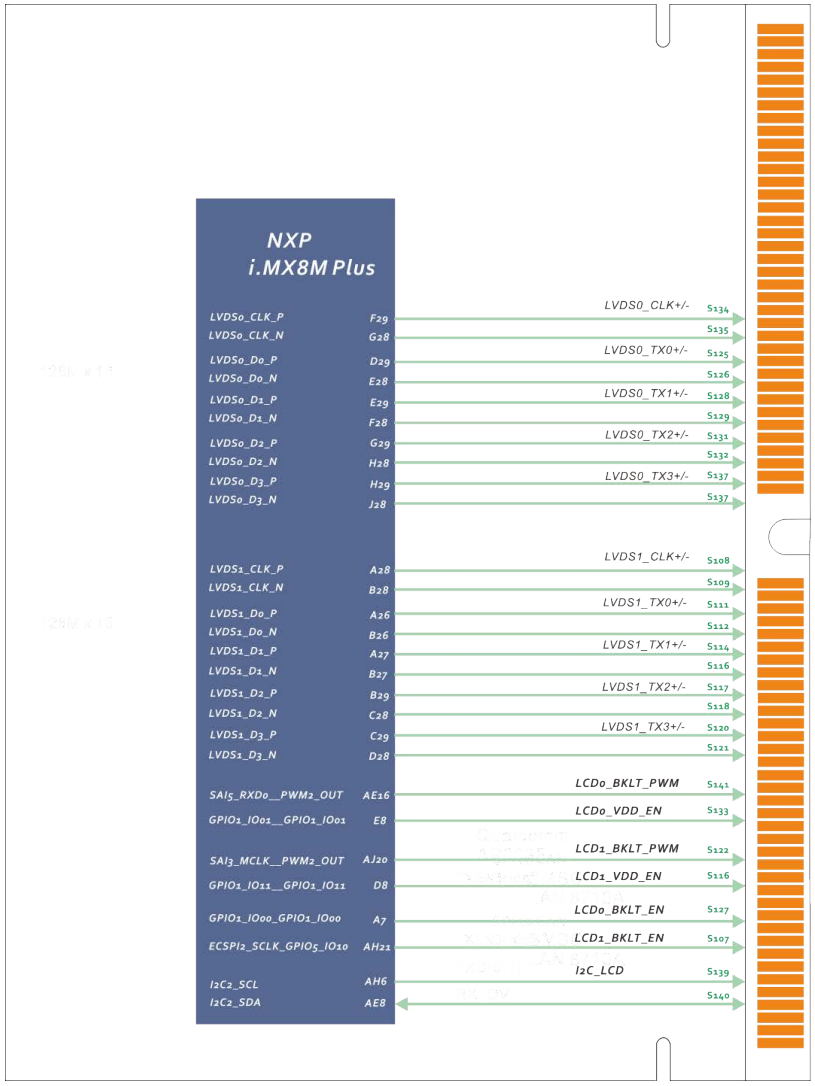


Figure 2 SMARC-iMX8MP LVDS LCD Diagram

2.1.6.1 LVDS Signals

The LVDS signals data flow from i.MX8M Plus processor to the golden finger connector is shown in the following table:

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Note
Ball	**Mode**	**Pin Name**	**Pin#**	**Net Name**	
LVDS Channel 0					
D29	*N/A*	*LVDS0_D0_P*	*S125*	*LVDS0_D0+*	*LVDS0 LCD data channel differential pairs 1*
E28	*N/A*	*LVDS0_D0_N*	*S126*	*LVDS0_D0-*	
E29	*N/A*	*LVDS0_D1_P*	*S128*	*LVDS0_D1+*	*LVDS0 LCD data channel differential pairs 2*
F28	*N/A*	*LVDS0_D1_N*	*S129*	*LVDS0_D1-*	
F29	*N/A*	*LVDS0_D2_P*	*S131*	*LVDS0_D2+*	*LVDS0 LCD data channel differential pairs 3*
H28	*N/A*	*LVDS0_D2_N*	*S132*	*LVDS0_D2-*	
H29	*N/A*	*LVDS0_D3_P*	*S137*	*LVDS0_D3+*	*LVDS0 LCD data channel differential pairs 4*
J28	*N/A*	*LVDS0_TX3_N*	*S138*	*LVDS0_D3-*	
F29	*N/A*	*LVDS0_CLK_P*	*S134*	*LVDS0_CK+*	*LVDS0 LCD differential clock pairs*
G28	*N/A*	*LVDS0_CLK_N*	*S135*	*LVDS0_CK-*	

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Note
Ball	Mode	Pin Name	Pin#	Net Name	
LVDS Channel 1					
A26	N/A	LVDS1_D0_P	S111	LVDS1_D0+	LVDS1 LCD data channel differential pairs 1
B26	N/A	LVDS1_D0_N	S112	LVDS1_D0-	
A27	N/A	LVDS1_D1_P	S114	LVDS1_D1+	LVDS1 LCD data channel differential pairs 2
B27	N/A	LVDS1_D1_N	S115	LVDS1_D1-	
B29	N/A	LVDS1_D2_P	S117	LVDS1_D2+	LVDS1 LCD data channel differential pairs 3
C28	N/A	LVDS1_D2_N	S118	LVDS1_D2-	
C29	N/A	LVDS1_D3_P	S120	LVDS1_D3+	LVDS1 LCD data channel differential pairs 4
D28	N/A	LVDS1_D3_N	S121	LVDS1_D3-	
B28	N/A	LVDS1_CLK_P	S108	LVDS1_CK+	LVDS1 LCD differential clock pairs
A28	N/A	LVDS1_CLK_N	S109	LVDS1_CK-	

A 24 bit dual channel *LVDS* implementation comprises 10 differential pairs: 4 pairs for odd pixel and control data; 1 pair for the *LVDS* clock for the odd data; 4 pairs for the even pixel data and control data, and 1 pair for the even *LVDS* clock. To use the dual channel *LVDS* mode, you need a display supporting the dual channel *LVDS* mode in order to receive odd and even pixel data.

2.1.6.2 Other LCD Control Signals

The signals in the table below support the *LVDS LCD* interfaces (as these are created from the same *i.MX8M Plus* source).

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
LCD0_VDD_EN	Output	CMOS 1.8V	High enables LVDS0 panel VDD
LCD0_BKLT_EN	Output	CMOS 1.8V	High enables LVDS0 panel backlight
LCD0_BKLT_PWM	Output	CMOS 1.8V	LVDS0 display backlight PWM control
LCD1_VDD_EN	Output	CMOS 1.8V	High enables LVDS1 panel VDD
LCD1_BKLT_EN	Output	CMOS 1.8V	High enables LVDS1 panel backlight
LCD1_BKLT_PWM	Output	CMOS 1.8V	LVDS1 display backlight PWM control
I2C_LCD_DAT	Bi-Dir OD	CMOS 1.8V	I2C data – to read LCD display EDID EEPROMs
I2C_LCD_CK	Output	CMOS 1.8V	I2C clock – to read LCD display EDID EEPROMs

Below list *LCD* control signals that mapping to *CPU* iomux and *SMARC* edge connector.

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
A7	ALT0	GPIO1_IO00__ GPIO1_IO00	S127	LCD0_BKLT_EN	LCD0_ BKLT_EN	High enables lvds0 panel backlight
E8	ALT0	GPIO1_IO01__ GPIO1_IO01	S133	LCD0_VDD_EN	LCD0_ VDD_EN	High enables lvds0 panel VDD
AE16	ALT2	SAI5_RXD0__ PWM2_OUT	S141	LCD0_BKLT_PWM	LCD0_ BKLT_ PWM	Lvds0 display backlight PWM control
AH21	ALT5	ECSPI2_SCLK__ GPIO5_IO10	S107	LCD1_BKLT_EN	LCD1_ BKLT_EN	High enables lvds1 panel backlight
D8	ALT0	GPIO1_IO11__ GPIO1_IO11	S116	LCD1_VDD_EN	LCD1_ VDD_EN	High enables lvds1 panel VDD
AJ20	ALT1	SAI3_MCLK__ PWM4_OUT	S141	LCD1_BKLT_PWM	LCD1_ BKLT_ PWM	Lvds1 display backlight PWM control
AH6	ALT0	I2C2_SCL__ I2C2_SCL	S139	I2C_LCD_CK	I2C_ LCD_CK	I2C data – to read LCD display EDID EEPROMs
AE8	ALT0	I2C2_SDA__ I2C2_SDA	S140	I2C_LCK_DAT	I2C_ LCD_DAT	I2C data – to read LCD display EDID EEPROMs

2.1.7. HDMI Interface

High-Definition Multimedia Interface *(HDMI)* is a licensable compact audio/video connector interface for transmitting uncompressed digital streams. *HDMI* encodes the video data into *TMDS* for digital transmission and is backward-compatible with the single-link Digital Visual Interface (*DVI*) carrying digital video. *i.MX8M Plus HDMI* Video quality can reach 3840 x 2160p30 resolutions with maximum pixel clock up to 297 MHz.

The *SMARC-iMX8MP* provides *HDMI* connection directly from the *NXP® i.MX8M Plus* processor. Video data is provided through three differential *TMDS* data pairs (*HDMI_D0±* to *HDMI_D2±*) and one differential clock pair (*HDMI_CLK±*). In addition, the *SMARC-iMX8MP* includes one standard *I2C* interface (*HDMI_CTRL_SDA* and *HDMI_CTRL_SCL*) for configuring and testing the *HDMI 3D Tx PHY* and a pin (*HDMI_HPD*) for *HDMI* hot plug detection support.

The following figure shows the *HDMI* block diagram.

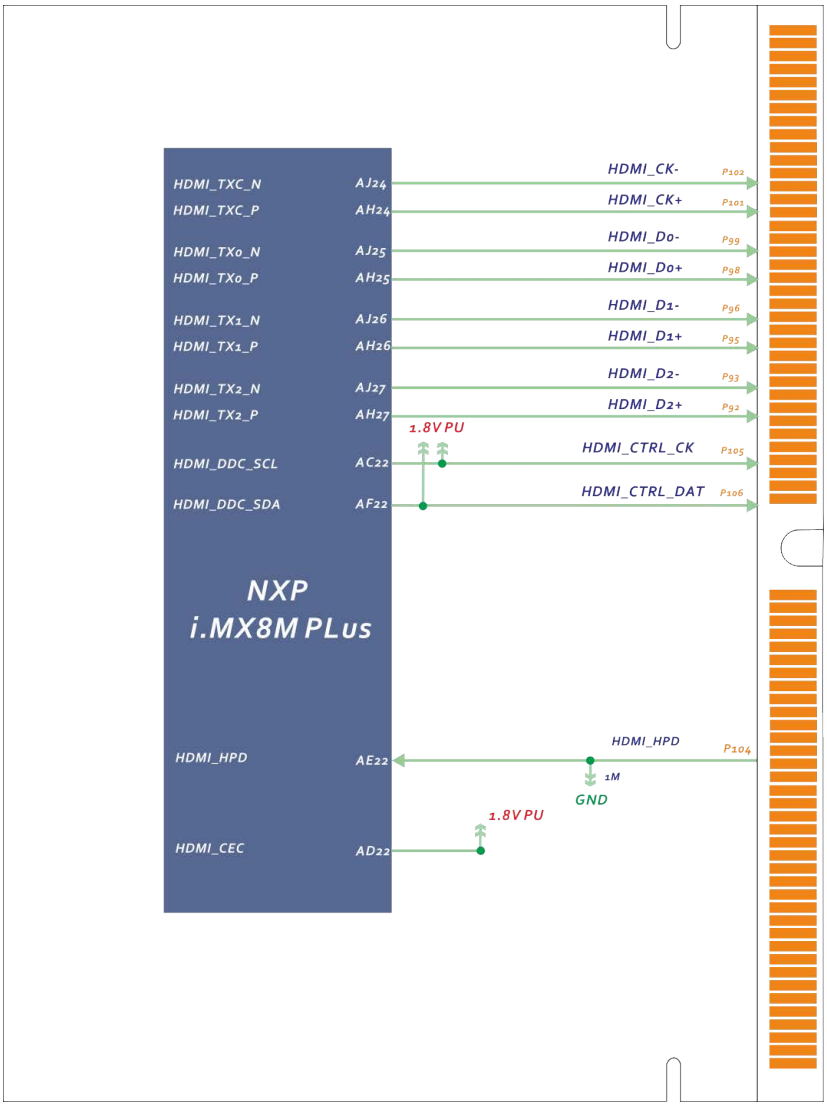


Figure 3 SMARC-iMX8MP HDMI Diagram

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Note
Ball	Mode	Pin Name	Pin#	Net Names	
HDMI					
AJ25	N/A	HDMI_TXo_N	P99	HDMI_D0-	TMDS / HDMI data differential pair 0
AH25	N/A	HDMI_TX0_P	P98	HDMI_D0+	
AJ26	N/A	HDMI_TX1_N	P96	HDMI_D1-	TMDS / HDMI data differential pair 1
AH26	N/A	HDMI_TX1_P	P95	HDMI_D1+	
AJ27	N/A	HDMI_TX2_N	P93	HDMI_D2-	TMDS / HDMI data differential pair 2
AH27	N/A	HDMI_TX2_P	P92	HDMI_D2+	
AJ24	N/A	HDMI_TXC_N	P102	HDMI_CK-	HDMI differential clock output pair
AH24	N/A	HDMI_TXC_P	P101	HDMI_CK+	
AE22	N/A	HDMI_HPD	P104	HDMI_HPD	HDMI Hot Plug Detect input
I2C Dedicate for HDMI					
AF22	N/A	HDMI_DDC_SDA	P106	HDMI_CTRL_DAT	I2C Data
AC22	N/A	HDMI_DDC_SCL	P105	HDMI_CTRL_CLK	I2C Clock

2.1.7.1 HDMI Signals

The table below shows the HDMI related signals.

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
HDMI_D[0:2]+ *HDMI_D[0:2]-*	*Output*	*TDMS*	*TMDS / HDMI data differential pairs*
HDMI_CK+ *HDMI_CK-*	*Output*	*TDMS*	*HDMI differential clock output pair*
HDMI_HPD	*Input*	*CMOS* *1.8V*	*HDMI Hot Plug Detect input*
HDMI_CTRL_DAT	*Bi-Dir* *OD*	*CMOS* *1.8V*	*I2C data line dedicated to HDMI*
HDMI_CTRL_CK	*Bi-Dir* *OD*	*CMOS* *1.8V*	*I2C clock line dedicated to HDMI*

HDMI displays uses *5V I2C* signaling. The Module *HDMI_CTRL_DAT* and *HDMI_CTRL_CK* signals are level translated on the Carrier from the Module *1.8V* level. A similar consideration applies to the *HDMI_HPD* signal. There are a number of single chip devices on the market that perform *ESD* protection and control signal level shifting for HDMI interfaces. The Texas Instruments *TPD12S016* is one such device.

2.1.8 USB Interface

The *Embedian SMARC-iMX8MP* module supports five *USB 2.0* ports (*USB 0:4*) and two *USB 3.0* ports (*USB[1:2]_SS*). The *USB 2.0* and *USB 3.0 IP* in *i.MX8M Plus* processor are independent. A Microchip *USB2514* is used to expand four USB 2.0 ports from *i.MX8M Plus USB 2.0* Host Port. Per the *SMARC* specification, the module supports a USB "*On-The-Go*" (*OTG*) port capable of functioning either as a client or host device, on the *SMARC USB0* port.

The following figure shows the *USB 0:4 (USB2.0)* and *USB[1:2]_SS (USB3.0)* block diagram.

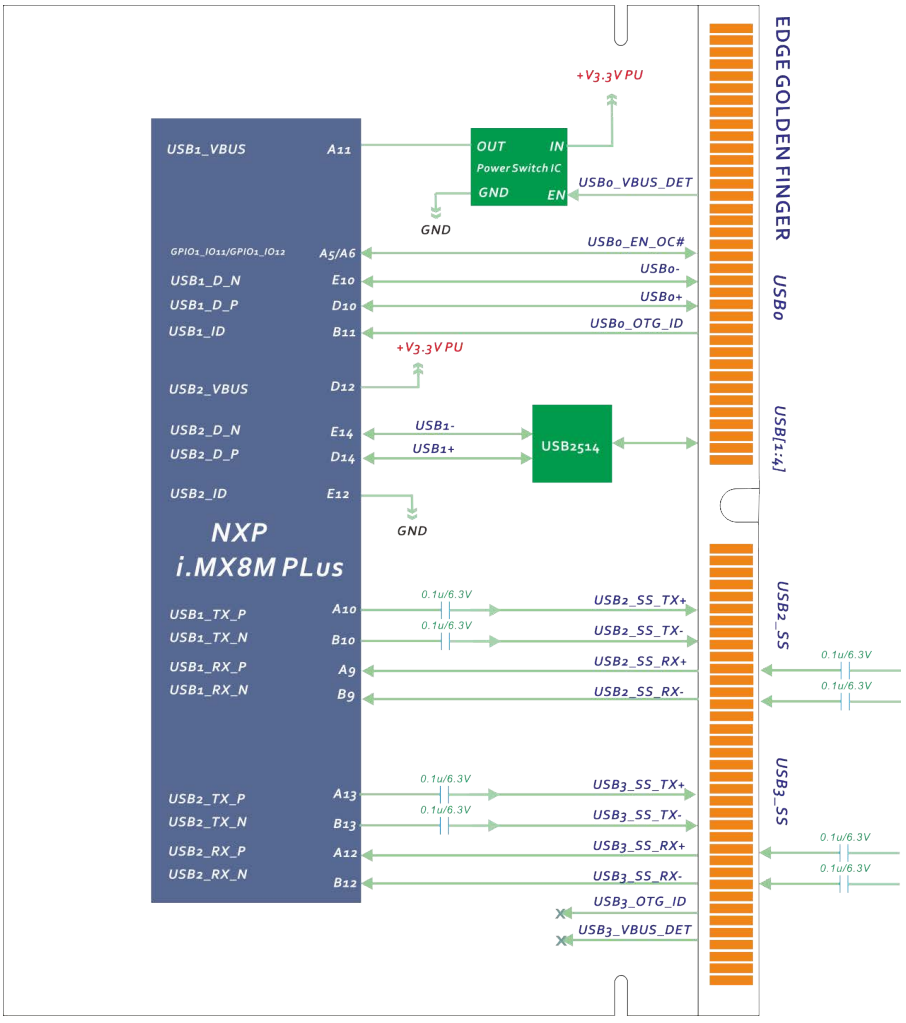


Figure 4 USB Block Diagram

USB interface signals are exposed on the *SMARC-iMX8MP* edge connector as shown below:

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	**Mode**	**Pin Name**	**Pin#**	**Pin Name**		
USB0 Port (USB 2.0 OTG)						
D10		*USB1_D_P*	*P60*	*USB0+*	*USB0+*	*USB0 data pair*
E10		*USB1_D_N*	*P61*	*USB0-*	*USB0-*	
A5 *A6*	*ALT0*	*GPIO1_IO11__ GPIO1_IO11 (EN)* *GPIO1_IO12__ GPIO1_IO12(OC#)*	*P62*	*USB0_EN_OC#*	*USB0_EN_OC#*	*USB0 power enable/over current indication signal*
A11		*USB1_VBUS (Turn on USB_OTG_VBUS)*	*P63*	*USB0_VBUS_ DET*	*USB0_VBUS_ DET*	*USB0 host power detection, when this port is used as a device.*
B11		*USB_ID*	*P64*	*USB0_OTG_ID*	*USB0_OTG_ID*	*USB0 OTG ID input, active high*

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
USB[1:4] Port (USB 2.0 Host)						
			P65	USB1+	USB1+	USB_DN3 of USB2514
			P66	USB1-	USB1-	
					VBUS_DET#	USB2514 HUB VBUS Detect
			P69	USB2+	USB2+	USB_DN1 of USB2514
			P70	USB2-	USB2-	
From USB2514			P71	USB2_EN_OC#	USB2_EN_OC#	USB2 power enable/over current indication signal
			S68	USB3+	USB3+	USB_DN2 of USB2514
			S69	USB3-	USB3-	
From USB2514			P74	USB3_EN_OC#	USB3_EN_OC#	USB3 power enable/over current indication signal
			S35	USB4+		USB_DN4 of USB2514
			S36	USB4-		
From USB2514			P76	USB4_EN_OC#	USB4_EN_OC#	USB4 power enable/over current indication signal

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
USB2_SS Port (USB 3.0 Host)						
A10		USB1_TX_P	S71	USB2_SSTX+	USB2_SSTX+	USB2 transmit signal differential pair positive
B10		USB1_TX_N	S72	USB2_SSTX-	USB2_SSTX-	USB2 transmit signal differential pair negative
A9		USB1_TX_P	S74	USB2_SSRX+	USB2_SSRX+	USB2 receive signal differential pair positive
B9		USB1_TX_N	S75	USB2_SSRX-	USB2_SSRX-	USB2 receive signal differential pair negative

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
USB3_SS Port (USB3.0 OTG)						
A13		USB2_TX_P	S62	USB3_SSTX+	USB3_SSTX+	USB3 transmit signal differential pair positive
B13		USB2_TX_N	S63	USB3_SSTX-	USB3_SSTX-	USB3 transmit signal differential pair negative
A12		USB2_TX_P	S65	USB3_SSRX+	USB3_SSRX+	USB3 receive signal differential pair positive
B12		USB2_TX_N	S66	USB3_SSRX-	USB3_SSRX-	USB3 receive signal differential pair negative

Note:

1. *USB0 OTG* role switch in *i.MX8M Plus* is implemented via *Vbus* switch from software driver. The *USB0_VBUS_DET (P63)* and *USB0_OTG_ID (P64)* pins defined in *SMARC 2.0* specification are not used.
2. If using *USB Type-C* connector, a *PTN5110* cc logic needs to be added in your carrier board. Please refer to *i.MX8M Plus EVK* evaluation board. The *USB Type-C* specification describes how the *USB* device uses pull-down/pull-up resistors on configuration channel pins to signify that it is a device or host.
3. If implementing *USB 3.0* function, *USB 2.0* signal lines need to be routed to *USB 3.0* connector to make it downward compatible.

2.1.8.1 USB Signals

The table below shows the USB related signals.

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
USB[0:4]+ *USB[0:4]-*	*Bi-Dir*	*USB*	*Differential US 2.0 Data Pair*
USB[0:4]_EN_OC#	*Bi-Dir* *OD*	*CMOS* *3.3V*	*Pulled low by Module OD driver to disable USB0 power.* *Pulled low by Carrier OD driver to indicate over-current situation.* *A 10k pull-up is present on the Module to a 3.3V rail. The pull-up rail may be switched off to conserve power if the USB port is not in use. Further details may be found in Section 2.1.8.2 USBx_EN_OC# Discussion below.*
USB0_VBUS_DET *USB1_VBUS_DET*	*Input*	*USB VBUS 5V*	*USB host power detection, when this port is used as a device.*
USB0_OTG_ID	*Input*	*CMOS* *3.3V*	*USB OTG ID input, active high.*
USB2_SSRX- *USB2_SSRX+*	*Input*	*USB SS*	*Receive signal differential pairs for SuperSpeed USB data Coupling caps for RX pairs are **off-Module (on Carrier Board)***
USB2_SSTX- *USB2_SSTX+*	*Output*	*USBSS*	*Transmit signal differential pairs for SuperSpeed USB data Coupling caps for RX pairs are **on-Module***

2.1.8.2 USB[0:4]_EN_OC# Discussion

The Module *USB[0:4]_EN_OC#* pins are multi-function Module pins, with a *10k* pull-up to a 3.3V rail on the Module, an OD driver on the Module, and, if the OC# (over-current) monitoring function is implemented on the Carrier, an *OD* driver on the Carrier. The use is as follows:

1) On the Carrier board, for external plug-in *USB* peripherals (*USB* memory sticks, cameras, keyboards, mice, etc.) *USB* power distribution is typically handled by USB power switches such as the Texas Instruments *TPS2052B* or the *Micrel MIC2026-1* or similar devices. The Carrier implementation is more straightforward if the Carrier *USB* power switches have active-high power enables and active low open drain *OC#* outputs (as the *TI* and *Micrel* devices referenced do). The USB power switch Enable and *OC#* pins for a given *USB* channel are tied together on the Carrier. The *USB* power switch enable pin must function with a low input current. The *TI* and *Micrel* devices referenced above require 1 microampere or less, at a *3.3V* enable voltage level.
2) The Module drives *USB[0:4]_EN_OC#* low to disable the power delivery to the *USBx* device.
3) The Module floats *USB[0:4]_EN_OC#* to enable power delivery. The line is pulled to *3.3V* by the Module pull-up, enabling the Carrier board *USB* power switch.
4) If there is a *USB* over-current condition, the Carrier board USB power switch drives the *USB[0:4]_EN_OC#* line low. This removes the over-current condition (by disabling the *USB* switch enable input), and allows Module software to detect the over-current condition.
5) The Module software should look for a falling edge interrupt on *USB[0:4]_EN_OC#*, while the port is enabled, to detect the *OC#* condition. The *OC#* condition will not last long, as the *USB* power switch is disabled when the switch IC detects the *OC#* condition.
6) If the *USB* power to the port is disabled (*USB[0:4]_EN_OC#* is driven low by the Module) then the Module software is aware that the port is disabled, and the low input value on the port does not indicate an over-current condition (because the port power is disabled).

Carrier Board *USB* peripherals that are not removable often do not make use of *USB* power switches with current limiting and over-current detection. It is usually deemed un-necessary for non-removable devices. In these cases, the

USB[0:4]_EN_OC# pins may be left unused, or they may be used as *USB[0:4]* power enables, without making use of the over-current detect Module input feature.

The *SMARC-iMX8MP* Module *USB* power enable and over current indication logic implementation is shown in the following block diagram. There are 10k pull-up resistors on the Module on the *SMARC USB[0:4]_EN_OC#* lines. Outputs driving the *USBx_EN_OC#* lines are open-drain. The Carrier board USB power switch, if present, is enabled by *USB[0:4]_EN_OC#* after a device connection is detected on the *DP/DM* lines.

The Enable pin on the Carrier board *USB* power switch must be active high and the Over-Current pin (*OC#*) must be open drain, active low (these are commonly available). No pull-up is required on the USB power switch Enable or *OC#* line on carrier board; they are tied together on the Carrier and fed to the Module *USB[0:4]_EN_OC#* pin.

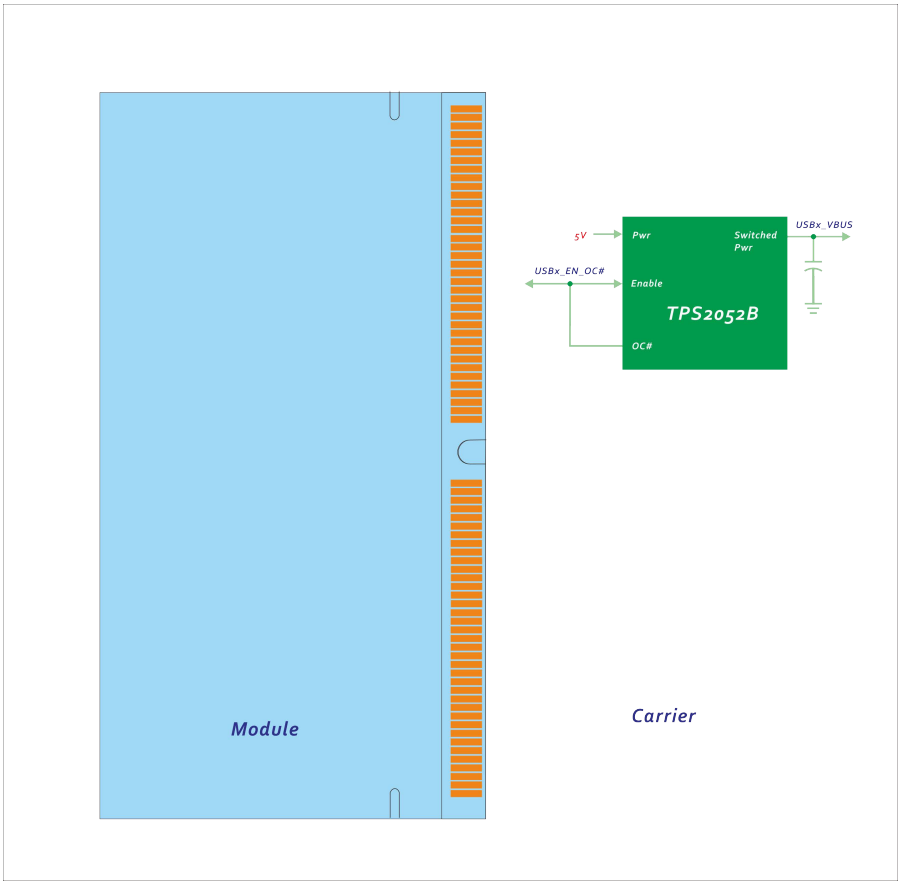


Figure 5 USB Power Distribution Implementation on Carrier

2.1.9. Gigabit Ethernet Controller (10/100/1000Mbps) Interface

The *SMARC-iMX8MP* module supports two Gigabit Ethernet (*10/100/1000*Mbps) interfaces, one (*GBE0*) supporting Time Sensitive Networking (*TSN*), drive gateway applications with low latency. The Gigabit Ethernet controller interfaces are accomplished by using the low-power Qualcomm Atheros AR8035 physical layer (PHY) transceiver with variable I/O voltage that is compliant with the *IEEE 802.3-2005* standards. The *AR8035* supports communication with an Ethernet MAC via a standard *RGMII* interface.

The Ethernet interface consists of 4 pairs of low voltage differential pair signals designated from *GBE0(1)_MDIO±* to *GBE0(1)_MDI3±* plus control signals for link activity indicators. These signals can be used to connect to a *10/100/1000* BaseT RJ45 connector with integrated or external isolation magnetics on the carrier board.

This is diagrammed below.

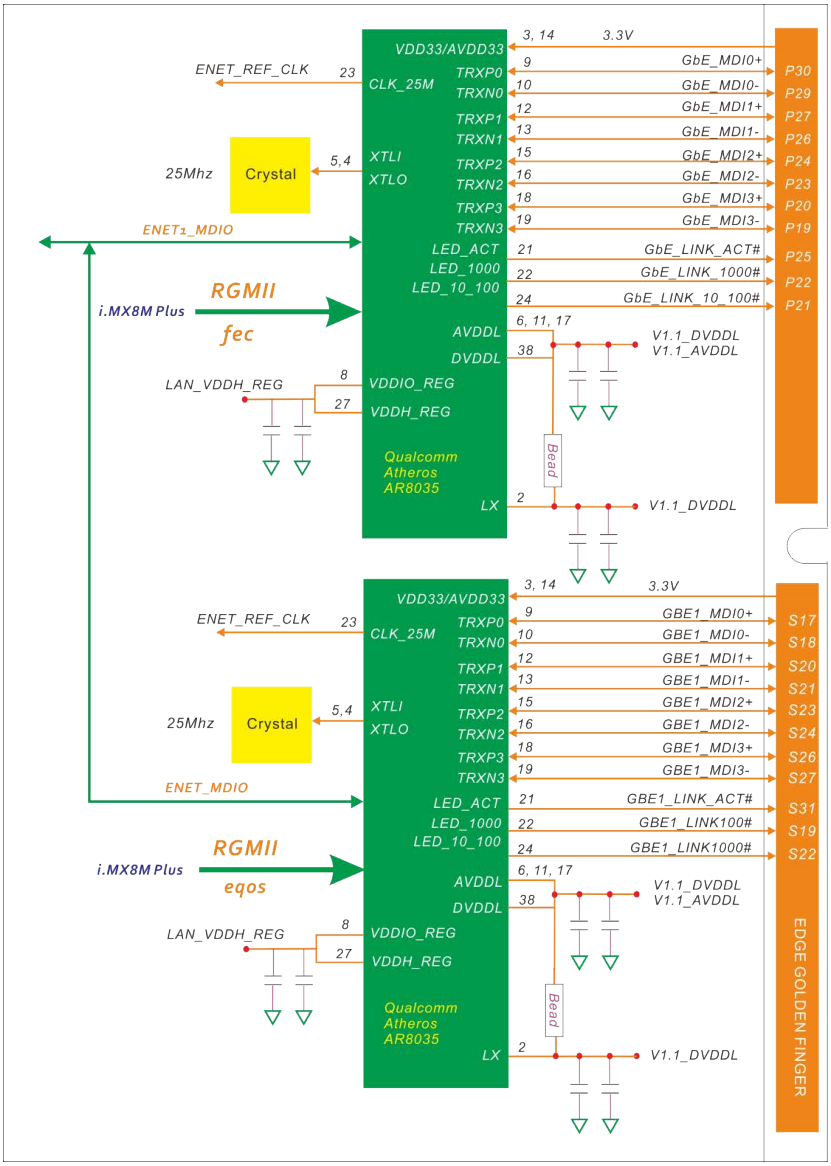


Figure 6 Gigabit Ethernet Connection from i.MX8M Plus to Qualcomm Atheros AR8035

2.1.9.1. Path of Gigabit LAN1

i.MX8M Plus processor and the first Qualcomm Atheros *AR8035* implementation is shown in the following table:

NXP i.MX8M Plus CPU			Qualcomm AR8035		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
Gigabit LAN1						
AJ8	ALT4	SAI1_RXD3__ ENET1_MDIO	39	MDIO	ENET1_MDIO	Serial Management Interface data input/output
AH9	ALT4	SAI1_RXD2__ ENET1_MDC	40	MDC	ENET1_MDC	Serial Management Interface clock
AD10	ALT4	SAI1_RXD4__ ENET1_RGMII_RD0	29	RXD0	ENET1_RD0	Bit 0 of the 4 data bits that are sent by the transceiver on the receive path.
AE10	ALT4	SAI1_RXD5__ ENET1_RGMII_RD1	28	RXD1	ENET1_RD1	Bit 1 of the 4 data bits that are sent by the transceiver on the receive path.
AH10	ALT4	SAI1_RXD6__ ENET1_RGMII_RD2	26	RXD2	ENET1_RD2	Bit 2 of the 4 data bits that are sent by the transceiver on the receive path.
AH12	ALT4	SAI1_RXD7__ENET1_ RGMII_RD3	25	RXD3	ENET1_RD3	Bit 3 of the 4 data bits that are sent by the transceiver on the receive path.
AJ12	ALT4	SAI1_TXC__ ENET1_RGMII_RXC	31	RX_CLK	ENET1_RXC	Reference clock
AF12	ALT4	SAI1_TXFS__ ENET1_RGMII_RX_CTL	30	RX_DV	ENET1_RX_CTL	Indicates both the receive data valid (RXDV) and receive error (RXER) functions per the RGMII specification.

NXP i.MX8M Plus CPU			Qualcomm AR8035		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
Gigabit LAN1						
AH14	ALT4	SAI1_TXD4__ ENET1_RGMII_TX_CTL	32	TX_EN	ENET1_TX_ CTL	Indicates that valid transmission data is present on TXD[3:0].
AJ11	ALT4	SAI1_TXD0__ ENET1_RGMII_TD0	34	TXD0	ENET1_TD0	The MAC transmits data to the transceiver using this signal.
AJ10	ALT4	SAI1_TXD1__ ENET1_RGMII_TD1	35	TXD1	ENET1_TD1	The MAC transmits data to the transceiver using this signal.
AH11	ALT4	SAI1_TXD2__ ENET1_RGMII_TD2	36	TXD2	ENET1_TD2	The MAC transmits data to the transceiver using this signal.
AD12	ALT4	SAI1_TXD3__ ENET1_RGMII_TD3	37	TXD3	ENET1_TD3	The MAC transmits data to the transceiver using this signal.
AH14	ALT4	SAI1_TXD5__ ENET1_RGMII_TXC	33	GTX_CLK	ENET1_TXC	Used to latch data from the MAC into the PHY. 1000BASE-T: 125MHz 100BASE-TX: 25MHz 10BASE-T: 2.5MHz
AF10	ALT5	SAI1_RXD1__ GPIO4_IO03				LAN1 interrupt pin
AH8	ALT5	SAI1_RXC__ GPIO4_IO01				IEEE 1588 Trigger Signal.

The path from *AR8035* to the golden finger edge connector is show in the following table.

Qualcomm AR8035		Golden Finger Edge Connector		Net Names	Note
Pin	Pin Name	Pin#	Pin Name		
AR8035 PHY1					
9	TRXP0	P30	GbE_MDI0+	GBE_MDI0+	Differential Transmit/Receive Positive Channel 0
10	TRXN0	P29	GbE_MDI0-	GBE_MDI0-	Differential Transmit/Receive Negative Channel 0
		P28	GbE_CTREF	GBE_CTREF	Center tap reference voltage
12	TRXP1	P27	GbE_MDI1+	GBE_MDI1+	Differential Transmit/Receive Positive Channel 1
13	TRXN1	P26	GbE_MDI1-	GBE_MDI1-	Differential Transmit/Receive Negative Channel 1
15	TRXP2	P24	GbE_MDI2+	GBE_MDI2+	Differential Transmit/Receive Positive Channel 2
16	TRXN2	P23	GbE_MDI2-	GBE_MDI2-	Differential Transmit/Receive Negative Channel 2
18	TRXP3	P20	GbE_MDI3+	GBE_MDI3+	Differential Transmit/Receive Positive Channel 3
19	TRXN3	P19	GbE_MDI3-	GBE_MDI3-	Differential Transmit/Receive Negative Channel 3

Qualcomm AR8035		Golden Finger Edge Connector		Net Names	Note
Pin	Pin Name	Pin#	Pin Name		
AR8035 PHY1					
21	LED_ACT	P25	GbE_LINK_ACT#	GBE_LINK_ACT#	Link / Activity Indication LED Driven low on Link (10, 100 or 1000 mbps) Blinks on Activity Could be able to sink 24mA or more Carrier LED current
24	LED_10_100	P21	GbE_LINK100#	GBE_LINK100#	Link Speed Indication LED for 100Mbps Could be able to sink 24mA or more Carrier LED current
22	LED_1000	P22	GbE_LINK1000#	GBE_LINK1000#	Link Speed Indication LED for 1000Mbps Could be able to sink 24mA or more Carrier LED current

2.1.9.2. Path of Gigabit LAN2

i.MX8M Plus processor and the second Qualcomm Atheros *AR8035* implementation is shown in the following table:

NXP i.MX8M Plus CPU			Qualcomm AR8035		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
Gigabit LAN 2						
AH29	ALT0	ENET_MDIO__ ENET_QOS_MDIO	39	MDIO	ENET_MDIO	Serial Management Interface data input/output
AH28	ALT0	ENET_MDC__ ENET_MDC	40	MDC	ENET_MDC	Serial Management Interface clock
AG29	ALT0	ENET_RD0__ ENET_QOS_RGMII_ RD0	29	RXD0	ENET_RD0	Bit 0 of the 4 data bits that are sent by the transceiver on the receive path.
AG28	ALT0	ENET_RD1__ ENET_QOS_RGMII_ RD1	28	RXD1	ENET_RD1	Bit 1 of the 4 data bits that are sent by the transceiver on the receive path.
AF29	ALT0	ENET_RD2__ ENET_QOS_RGMII_ RXD2	26	RXD2	ENET_RD2	Bit 2 of the 4 data bits that are sent by the transceiver on the receive path.
AF28	ALT0	ENET_RD3__ ENET_QOS_RGMII_ RXD3	25	RXD3	ENET_RD3	Bit 3 of the 4 data bits that are sent by the transceiver on the receive path.
AE29	ALT0	ENET_RXC__ CCM_ENET_QOS_ CLOCK_GENERATE_ RX_CLK	31	RX_CLK	ENET_RXC	Reference clock
AE28	ALT0	ENET_RX_CTL__ ENET_QOS_RGMII_ RX_CTL	30	RX_DV	ENET_RX_ CTL	Indicates both the receive data valid (RXDV) and receive error (RXER) functions per the RGMII specification.

NXP i.MX8M Plus CPU			Qualcomm AR8035		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
Gigabit LAN 2						
B48	ALT0	ENET_TX_CTL__ ENET_QOS_RGMII_TX_ CTL	32	TX_EN	ENET_TX_CTL	Indicates that valid transmission data is present on TXD[3:0].
AC25	ALT0	ENET_TD0__ ENET_QOS_RGMII_ TD0	34	TXD0	ENET_TD0	The MAC transmits data to the transceiver using this signal.
AE26	ALT0	ENET_TD1__ ENET_QOS_RGMII_ TD1	35	TXD1	ENET_TD1	The MAC transmits data to the transceiver using this signal.
AF26	ALT0	ENET_TD2__ ENET_QOS_RGMII_ TD2	36	TXD2	ENET_TD2	The MAC transmits data to the transceiver using this signal.
AD24	ALT0	ENET_TD3__ ENET_QOS_RGMII_ TD3	37	TXD3	ENET_TD3	The MAC transmits data to the transceiver using this signal.
AE24	ALT0	ENET_RGMII_TXC__ CCM_ENET_QOS_ CLOCK_GENERATE_ TX_CLK	33	GTX_CLK	ENET_TXC	Used to latch data from the MAC into the PHY. 1000BASE-T: 125MHz 100BASE-TX: 25MHz 10BASE-T: 2.5MHz
AH17	ALT5	SAI2_RXFS__ GPIO4_IO21				LAN2 interrupt pin
B8	ALT0	GPIO1_IO09__ GPIO1_IO09				IEEE 1588 Trigger Signal.

The path from the second *AR8035* to the golden finger edge connector is show in the following table.

Qualcomm AR8035		Golden Finger Edge Connector		Net Names	Note
Pin	Pin Name	Pin#	Pin Name		
AR8035 PHY 2					
9	TRXP0	S17	GBE1_MDI0+	GBE1_MDI0+	Differential Transmit/Receive Positive Channel 0
10	TRXN0	S18	GBE1_MDI0-	GBE1_MDI0-	Differential Transmit/Receive Negative Channel 0
		S28	GBE1_CTREF	GBE1_CTREF	Center tap reference voltage
12	TRXP1	S20	GBE1_MDI1+	GBE1_MDI1+	Differential Transmit/Receive Positive Channel 1
13	TRXN1	S21	GBE1_MDI1-	GBE1_MDI1-	Differential Transmit/Receive Negative Channel 1
15	TRXP2	S23	GBE1_MDI2+	GBE1_MDI2+	Differential Transmit/Receive Positive Channel 2
16	TRXN2	S24	GBE1_MDI2-	GBE1_MDI2-	Differential Transmit/Receive Negative Channel 2
18	TRXP3	S26	GBE1_MDI3+	GBE1_MDI3+	Differential Transmit/Receive Positive Channel 3
19	TRXN3	S27	GBE1_MDI3-	GBE1_MDI3-	Differential Transmit/Receive Negative Channel 3

Qualcomm AR8035		Golden Finger Edge Connector		Net Names	Note
Pin	Pin Name	Pin#	Pin Name		
AR8035 PHY 2					
21	LED_ACT	S31	GBE1_LINK_ACT#	GBE1_LINK_ACT#	Link / Activity Indication LED Driven low on Link (10, 100 or 1000 mbps) Blinks on Activity Could be able to sink 24mA or more Carrier LED current
24	LED_10_100	S19	GBE1_LINK100#	GBE1_LINK100#	Link Speed Indication LED for 100Mbps Could be able to sink 24mA or more Carrier LED current
22	LED_1000	S22	GBE1_LINK1000#	GBE1_LINK1000#	Link Speed Indication LED for 1000Mbps Could be able to sink 24mA or more Carrier LED current

2.1.9.3. Gigabit LAN Signals

The table below shows the Gigabit LAN related signals.

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
GBE0(1)_MDI0+ GBE0(1)_MDI0-	Bi-Dir	GBE_MDI	Bi-directional transmit/receive pair 0 to magnetics (Media Dependent Interface)
GBE0(1)_MDI1+ GBE0(1)_MDI1-	Bi-Dir	GBE_MDI	Bi-directional transmit/receive pair 1 to magnetics (Media Dependent Interface)
GBE0(1)_MDI2+ GBE0(1)_MDI2-	Bi-Dir	GBE_MDI	Bi-directional transmit/receive pair 2 to magnetics (Media Dependent Interface)
GBE0(1)_MDI3+ GBE0(1)_MDI3-	Bi-Dir	GBE_MDI	Bi-directional transmit/receive pair 3 to magnetics (Media Dependent Interface)
GBE0(1)_100#	Output OD	CMOS 3.3V	Link Speed Indication LED for 100Mbps Could be able to sink 24mA or more Carrier LED current
GBE0(1)_1000#	Output OD	CMOS 3.3V	Link Speed Indication LED for 1000Mbps Could be able to sink 24mA or more Carrier LED current
GBE0(1)_LINK_ACK#	Output OD	CMOS 3.3V	Link / Activity Indication LED Driven low on Link (10, 100 or 1000 mbps) Blinks on Activity Could be able to sink 24mA or more Carrier LED current
GBE0(1)_CTREF	Output	Reference Voltage	Center-Tap reference voltage for GBE0 Carrier board Ethernet magnetic (not required by the Module GBE PHY)

2.1.9.4. Suggested Magnetics

Listed below are suggested magnetics.

For normal temperature (0°C ~70°C) products.

Vendor	P/N	Package	Cores	Temp	Configuration
Halo	HFJ11-1G02E	Integrated RJ45	8	0°C~70°C	HP Auto-MDIX
UDE	RB1-BA6BT9WA	Integrated RJ45	8	-40°C~85°C	HP Auto-MDIX
Halo	TG1G-S002NZRL	24-pin SOIC-W	8	0°C~70°C	HP Auto-MDIX

For industrial temperature (-40°C ~85°C) products.

Vendor	P/N	Package	Cores	Temp	Configuration
UDE	RB1-BA6BT9WA	Integrated RJ45	8	-40°C~85°C	HP Auto-MDIX
Halo	TG1G-E012NZRL	24-pin SOIC-W	8	-40°C~85°C	HP Auto-MDIX

2.1.10 MIPI-DSI Interface

The *SMARC-iMX8MP* implements one 4-lane *MIPI-DSI* output streams that are shared with *LVDS1* interface and are defined in SMARC 2.0 edge connector from *i.MX8M Plus MIPI_DSI1* interface.

The *MIPI-DSI LCD* signals found on the *SMARC-i.MX8M Plus* offers one 4-lane channels, with resolutions up to 1,920 × 1,200 @60 fps at 24 bpp. They are generated from *MIPI_DSI1* signals from the *NXP® i.MX8M Plus* processor.

The following figure shows the *MIPI DSI LCD* block diagram.

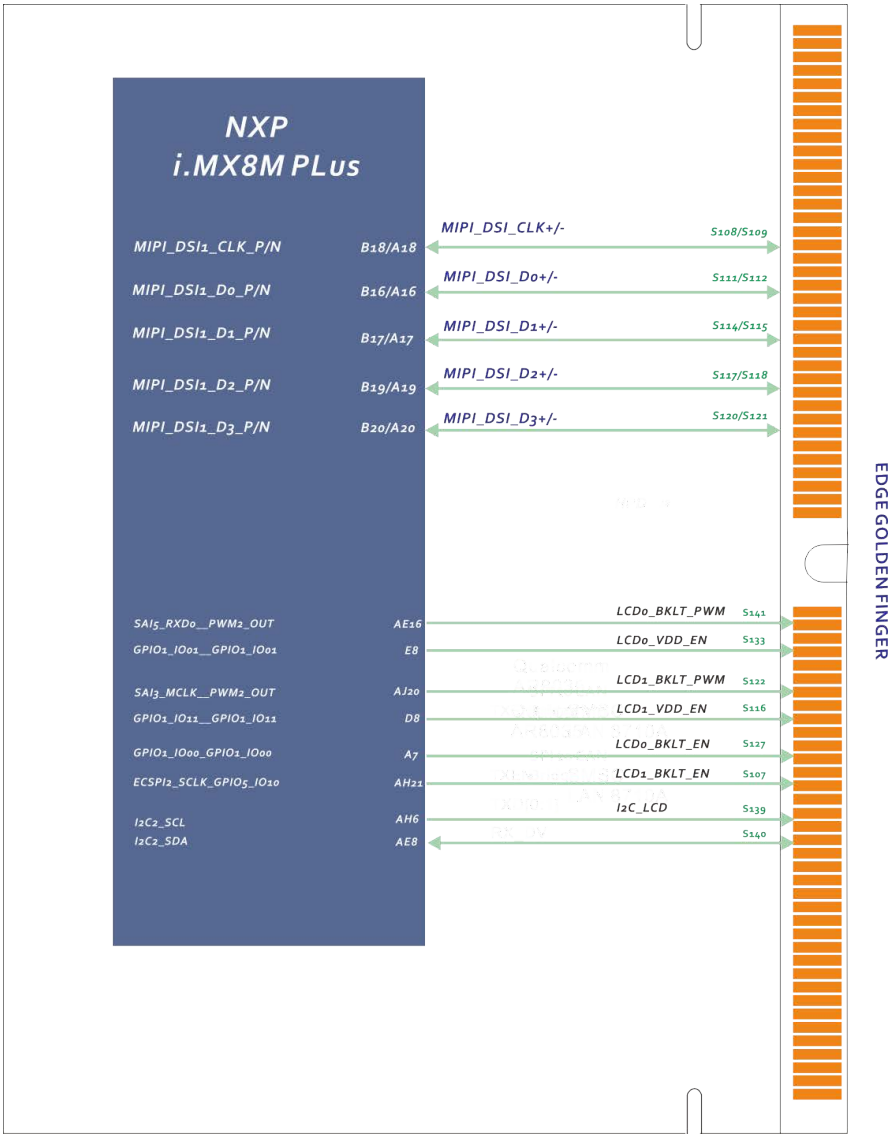


Figure 7 SMARC-iMX8MP MIPI-DSI LCD Diagram

2.1.10.1 MIPI-DSI Signals Data Flow

The *MIPI-DSI* signals from *i.MX8M Plus* processor to the golden fingle connector is shown in the following table:

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Note
Ball	**Mode**	**Pin Name**	**Pin#**	**Net Name**	
MIPI DSI					
A16	*N/A*	*MIPI_DSI1_D0_P*	*S111*	*LVDS1_0+ / eDP1_TX0+ / DSI1_D0+*	*MIPI DSI LCD data channel differential pairs 1*
B16	*N/A*	*MIPI_DSI1_D0_N*	*S112*	*LVDS1_0- / eDP1_TX0- / DSI1_D0-*	
A17	*N/A*	*MIPI_DSI1_D1_P*	*S114*	*LVDS1_1+ / eDP1_TX1+ / DSI1_D1+*	*MIPI DSI LCD data channel differential pairs 2*
B17	*N/A*	*MIPI_DSI1_D1_N*	*S115*	*LVDS1_1- / eDP1_TX1- / DSI1_D1-*	
A19	*N/A*	*MIPI_DSI1_D2_P*	*S117*	*LVDS1_2+ / eDP1_TX2+ / DSI1_D2+*	*MIPI DSI LCD data channel differential pairs 3*
B19	*N/A*	*MIPI_DSI1_D2_N*	*S118*	*LVDS1_2- / eDP1_TX2- / DSI1_D2-*	
A20	*N/A*	*MIPI_DSI1_D3_P*	*S120*	*LVDS1_3+ / eDP1_TX3+ / DSI1_D3+*	*MIPI DSI LCD data channel differential pairs 4*
B20	*N/A*	*MIPI_DSI1_D3_N*	*S121*	*LVDS1_3- / eDP1_TX3- / DSI1_D3-*	
A18	*N/A*	*MIPI_DSI1_CLK_P*	*S108*	*LVDS1_CK+ / eDP1_AUX+ / DSI1_CLK+*	*MIPI DSI LCD differential clock pairs*
B18	*N/A*	*MIPI_DSI1_CLK_N*	*S109*	*LVDS1_CK- / eDP1_AUX- / DSI1_CLK-*	

2.1.10.2 MIPI DSI Signals

The table below shows the *MIPI DSI* related signals.

Edge Golden Finder Signal Name	Direction	Coupling Tolerance	Description
DSI1_LANE[0:3]+	Output	AC Coupled off module	DSI Data Pair [0:3] positive
DSI1_LANE[0:3]-	Output	AC Coupled off module	DSI Data Pair [0:3] negative

2.1.10.3 Other LCD Control Signals

The signals in the table below support the *MIPI LCD* interfaces (as these are created from the same *i.MX8M Plus* source).

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
LCD0_VDD_EN	Output	CMOS 1.8V	High enables DSI1 panel VDD
LCD0_BKLT_EN	Output	CMOS 1.8V	High enables DSI1 panel backlight
LCD0_BKLT_PWM	Output	CMOS 1.8V	eDP0 display backlight PWM control
LCD1_VDD_EN	Output	CMOS 1.8V	High enables DSI1 panel VDD
LCD1_BKLT_EN	Output	CMOS 1.8V	High enables DSI1 panel backlight
LCD1_BKLT_PWM	Output	CMOS 1.8V	DSI1 display backlight PWM control

Below list *LCD* control signals that mapping to *CPU* iomux and *SMARC* edge connector.

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
A7	ALT0	GPIO1_IO00__ GPIO1_IO00	S127	LCD0_BKLT_EN	LCD0_ BKLT_EN	High enables lvds0 panel backlight
E8	ALT0	GPIO1_IO01__ GPIO1_IO01	S133	LCD0_VDD_EN	LCD0_ VDD_EN	High enables lvds0 panel VDD
AE16	ALT2	SAI5_RXD0__ PWM2_OUT	S141	LCD0_BKLT_PWM	LCD0_ BKLT_ PWM	Lvds0 display backlight PWM control
AH21	ALT5	ECSPI2_SCLK__ GPIO5_IO10	S107	LCD1_BKLT_EN	LCD1_ BKLT_EN	High enables lvds1 panel backlight
D8	ALT0	GPIO1_IO11__ GPIO1_IO11	S116	LCD1_VDD_EN	LCD1_ VDD_EN	High enables lvds1 panel VDD
AJ20	ALT2	SAI3_MCLK__ PWM4_OUT	S141	LCD1_BKLT_PWM	LCD1_ BKLT_ PWM	Lvds1 display backlight PWM control
AH6	ALT0	I2C2_SCL__ I2C2_SCL	S139	I2C_LCD_CK	I2C_ LCD_CK	I2C data – to read LCD display EDID EEPROMs
AE8	ALT0	I2C2_SDA__ I2C2_SDA	S140	I2C_LCK_DAT	I2C_ LCD_DAT	I2C data – to read LCD display EDID EEPROMs

2.1.11. PCIe Interfaces

The *SMARC-iMX8MP* offers one *PCI* Express Gen 3.0 single lane interface. The *PCIe* signals are routed from the *NXP® i.MX8M Plus* processor to the *PCI* Express port *A* of the *SMARC-iMX8MP* edge finger. These signals support *PCI* Express Gen. 3.0 interfaces at 6 Gb/s and are backward compatible to Gen. 2.0 and 1.1 interfaces at 5Gb/s and 2.5 Gb/s. Only single lane *PCI Express* link configuration is possible. *Pericom PI6CFGL201BZDIE* clock generator is used on *PCIe* port to make *PCIe* reference clock *HCSL* signals.

The following figure shows the *PCIE* port *A* block diagram.

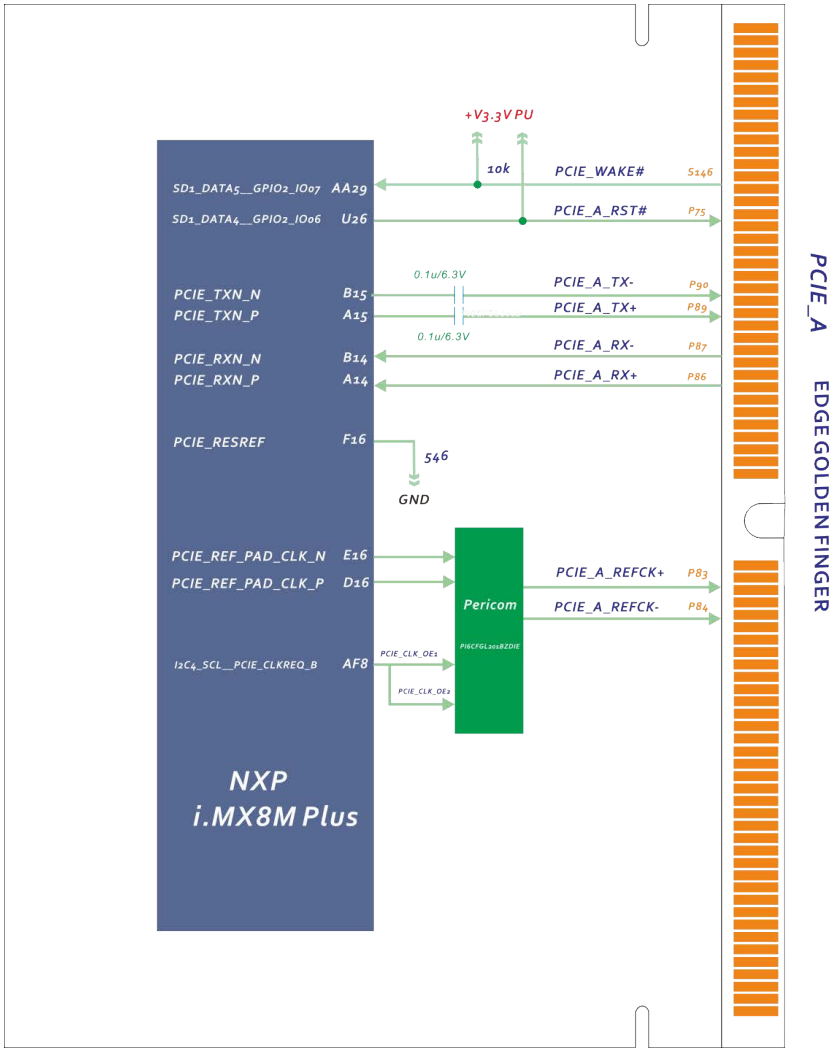


Figure 8 PCI Express Block Diagram

PCI Express interface signals are exposed on the *SMARC-iMX8MP* edge connector as shown below:

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
PCI Express Port A						
U26	ALT5	SD1_DATA4__ GPIO2_IO06	P75	PCIE_A_RST#	PCIE_A_RST#	Reset Signal for external devices.
N/A		From Clock Generator	P83	PCIE_A_REFCK+	PCIE_A_REFCK+	Differential PCI Express Reference Clock Signals for Lanes A
N/A		From Clock Generator	P84	PCIE_A_REFCK-	PCIE_A_REFCK-	
A14		PCIE_RXN_P	P86	PCIE_A_RX+	PCIE_A_RX+	Differential PCIe Link A receive data pair 0
B14		PCIE_RXN_N	P87	PCIE_A_RX-	PCIE_A_RX-	
A15		PCIE_TXN_P	P89	PCIE_A_TX+	PCIE_A_TX+	Differential PCIe Link A transmit data pair 0
B15		PCIE_TXN_N	P90	PCIE_A_TX-	PCIE_A_TX-	
AA29	ALT5	SD1_DATA5__ GPIO2_IO07	S146	PCIE_WAKE#	PCIE_WAKE#	PCIe wake up interrupt to host

2.1.11.1. PCIe_Link Signals

The table below shows the *PCIe_Link A* related signals.

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
PCI Express Port A			
PCIE_A_TX+ *PCIE_A_TX-*	*Output*	*HCSL PCIe*	*Differential PCIe Link A transmit data pair 0* *Series coupling caps is on the Module* *Caps is 0402 package 0.1uF*
PCIE_A_RX+ *PCIE_A_RX-*	*Input*	*HCSL PCIe*	*Differential PCIe Link A receive data pair 0* *No coupling caps on Module*
PCIE_A_REFCK+ *PCIE_A_REFCK-*	*Output*	*HCSL PCIe*	*Differential PCIe Link A reference clock output* *DC coupled*
PCIE_A_RST#	*Output*	*CMOS* *3.3V*	*PCIe Port A reset output*

2.1.11.2. PCIe Wake Signals

The table below shows the *PCIe Wake* signal.

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
PCIE_WAKE#	Input	CMOS 3.3V	PCIe wake up interrupt to host – common to PCIe links A, B, C – pulled up or terminated on Module

2.1.12. MIPI/CMOS Serial Camera Interface (MIPI_CSI)

The *NXP® i.MX8M Plus* provides connectivity to cameras via the *MIPI/CSI-2* transmitter and maintains image manipulation and processing with adequate synchronization and control. The Camera Serial Interface (*CSI*) controls the camera port and provides interface to an image sensor or a related device. The role of the camera ports is to receive input from video sources and to provide support for time-sensitive signals to the camera. Non-time-sensitive controls such as configuration, reset are performed by the ARM platform through I2C interface or GPIO signals.

The camera interface on *SMARC-iMX8MP* is designed as serial interfaces on *CSI0* pin groups that can support 4 lanes and *CSI1* pin groups that can support 2 lanes providing an interface between the system and the *MIPI* D-PHY, allowing communication with an *MIPI CSI-2* compliant camera sensor.

The *MIPI-CSI2* in *SMARC-iMX8MP* supports high speed mode (80Mbps - 1.5Gbps) per lane, providing 4K@45fps capability for the 4 lanes.

The following figure shows the serial camera interface block diagram.

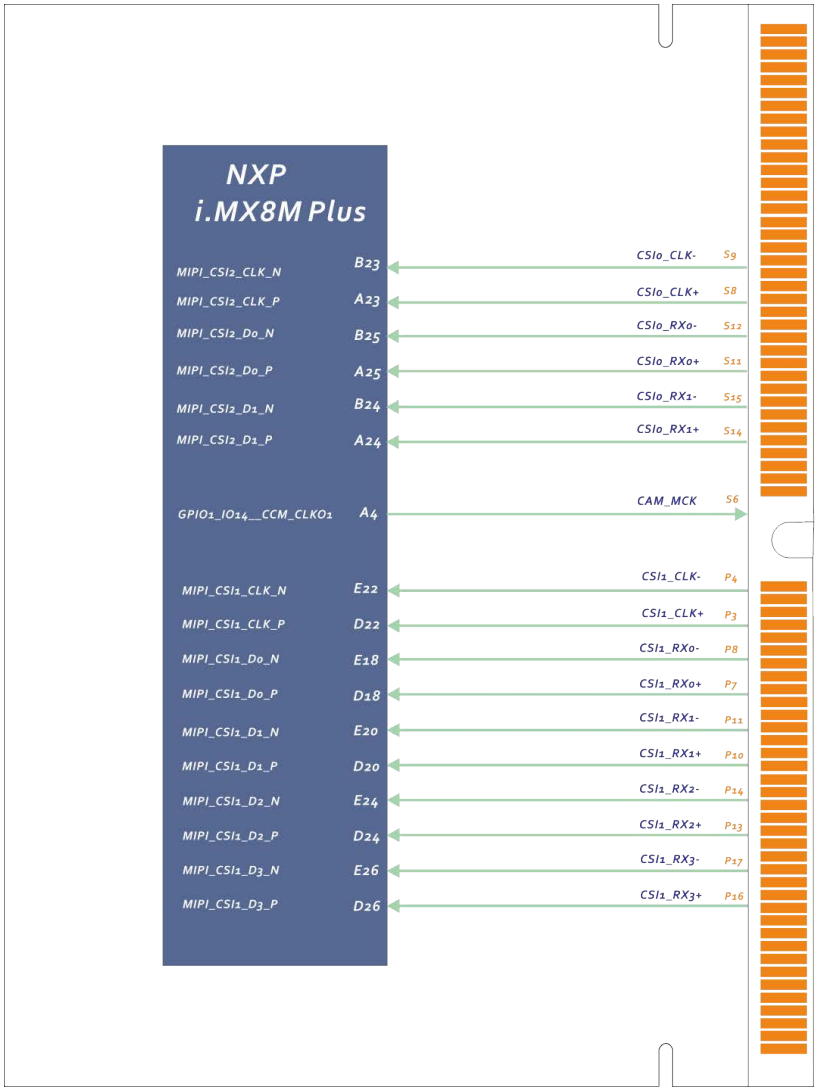


Figure 9 MIPI/Serial Camera Interface Block Diagram

MIPI/Serial Camera interface signals are exposed on the *SMARC-iMX8MP* edge connector as shown below:

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
MIPI/Serial Camera Interface 1 (CSI0)						
A4	ALT6	GPIO1_IO14__ CCM_CLKO1	S6	CAM_MCK	CAM_MCK	Master clock output for CSI camera support
B23	ALT0	MIPI_CSI2_CLK_N	S9	CSI0_CK-	CSI0_CK-	CSI0 differential clock inputs
A23	ALT0	MIPI_CSI2_CLK_P	S8	CSI0_CK+	CSI0_CK+	
B25	ALT0	MIPI_CSI2_D0_N	S12	CSI0_RX0-	CSI0_D0-	CSI0 differential data inputs
A25	ALT0	MIPI_CSI2_D0_P	S11	CSI0_RX0+	CSI0_D0+	
B24	ALT0	MIPI_CSI2_D1_N	S15	CSI0_RX1-	CSI0_D1-	
A24	ALT0	MIPI_CSI2_D1_P	S14	CSI0_RX1+	CSI0_D1+	
MIPI/Serial Camera Interface 2 (CSI1)						
E22	ALT0	MIPI_CSI1_CLK_N	P4	CSI1_CK-	CSI1_CK-	CSI1 differential clock inputs
D22	ALT0	MIPI_CSI1_CLK_P	P3	CSI1_CK+	CSI1_CK+	
E18	ALT0	MIPI_CSI1_D0_N	P8	CSI1_RX0-	CSI1_D0-	CSI1 differential data inputs
D18	ALT0	MIPI_CSI1_D0_P	P7	CSI1_RX0+	CSI1_D0+	
E20	ALT0	MIPI_CSI1_D1_N	P11	CSI1_RX1-	CSI1_D1-	
D20	ALT0	MIPI_CSI1_D1_P	P10	CSI1_RX1+	CSI1_D1+	
E24	ALT0	MIPI_CSI1_D2_N	P14	CSI1_RX2-	CSI1_D2-	
D24	ALT0	MIPI_CSI1_D2_P	P13	CSI1_RX2+	CSI1_D2+	
E26	ALT0	MIPI_CSI1_D3_N	P17	CSI1_RX3-	CSI1_D3-	
D26	ALT0	MIPI_CSI1_D3_P	P16	CSI1_RX3+	CSI1_D3+	

2.1.12.1. Camera I2C Support

The I2C_CAM0/1 port is intended to support serial and parallel cameras. Most contemporary cameras with I2C support allow a choice of two I2C address ranges.

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
I2C_CAM0						
AH20	ALT2	ECSPI2_MISO__ I2C4_SCL	S5	CSI0_TX+/ I2C_CAM0_CK	I2C_CAM0_CK	
AJ22	ALT2	ECSPI2_SS0__ I2C4_SDA	S7	CSI0_TX-/ I2C_CAM0_DAT	I2C_CAM0_DAT	
I2C_CAM1						
AC14	ALT3	SAI5_RXFS__ I2C6_SCL	S1	I2C_CAM1_CK	I2C_CAM1_CK	
AD14	ALT3	SAI5_RXC__ I2C6_SDA	S2	I2C_CAM1_DAT	I2C_CAM1_DAT	
CSI Clock Output						
A4	ALT6	GPIO1_IO14__ CCM_CLKO1	S6	CAM_MCK	CAM_MCK	

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
I2C_CAM0			
I2C_CAM0_DAT	Bi-Dir OD	CMOS 1.8V	Serial camera support link - I2C data
I2C_CAM0_CK	Bi-Dir OD	CMOS 1.8V	Serial camera support link - I2C clock
I2C_CAM1			
I2C_CAM1_DAT	Bi-Dir OD	CMOS 1.8V	Serial camera support link - I2C data
I2C_CAM1_CK	Bi-Dir OD	CMOS 1.8V	Serial camera support link - I2C clock

Note:

1. Embedian BSP and development board (EVK-STD-CARRIER-S20) supports Coral OV5645 camera module (P/N: G840-00180-01, https://coral.ai/products/camera/)

2.1.12.2. MIPI Serial Camera In – MIPI CSI0/1

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
MIPI_CSI0_D[0:1]+ MIPI_CSI0_D[0:1]-	Input	LVDS D-PHY	CSI1 differential data inputs
MIPI_CSI0_CK+ MIPI_CSI0_CK-	Input	LVDS D-PHY	CSI1 differential clock inputs
MIPI_CSI1_D[0:3]+ MIPI_CSI1_D[0:3]-	Input	LVDS D-PHY	CSI1 differential data inputs
MIPI_CSI1_CK+ MIPI_CSI1_CK-	Input	LVDS D-PHY	CSI1 differential clock inputs
CAM_MCK	Output	CMOS 1.8V	Master clock output for CSI1 camera support

2.1.13 SD/SDMMC Interface

SMARC-iMX8MP is configured to support two *MMC* controllers. One is used for on-module 8-bit *eMMC* support, and the other one is used for external *SDHC/SDIO* interface. The *SMARC-iMX8MP* module supports one 4-bit *SDIO* interface, per the *SMARC* 2.0 specification. The *SDIO* interface uses 3.3V signaling, per the *SMARC* spec and for compatibility with commonly available *SDIO* cards.

The following figure shows the *SDIO* block diagram.

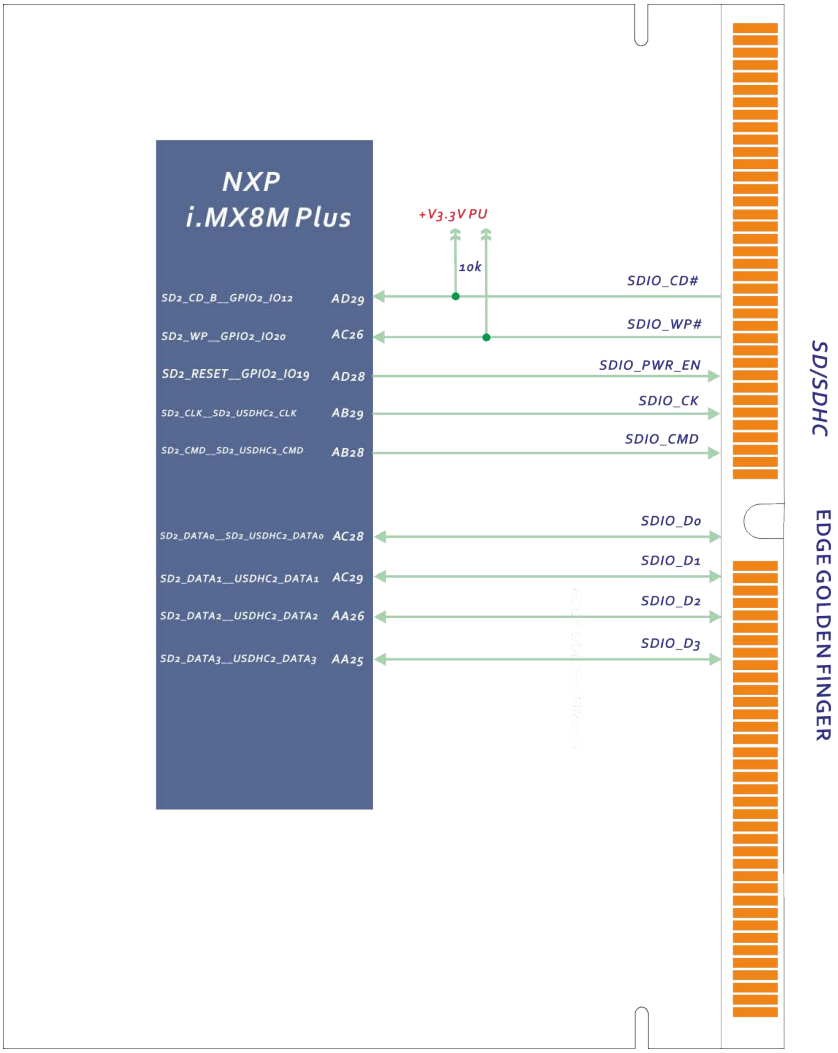


Figure 10 SD/SDIO/eMMC Interface Block Diagram

SDIO interface signals are exposed on the *SMARC* golden finger edge connector as shown below:

NXP i.MX8M CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
SD/SDIO						
AC28	ALT0	SD2_DATA0__ USDHC2_DATA0	P39	SDIO_D0	SDIO_D0	SDIO Data 0
AC29	ALT0	SD2_DATA1__ USDHC2_DATA1	P40	SDIO_D1	SDIO_D1	SDIO Data 1
AA26	ALT0	SD2_DATA2__ USDHC2_DATA2	P41	SDIO_D2	SDIO_D2	SDIO Data 2
AA25	ALT0	SD2_DATA3__ USDHC2_DATA3	P42	SDIO_D3	SDIO_D3	SDIO Data 3
AC26	ALT5	SD2_WP__ GPIO2_IO20	P33	SDIO_WP	SDIO_WP	SDIO write protect signal
AB28	ALT0	SD2_CMD__ USDHC2_CMD	P34	SDIO_CMD	SDIO_CMD	SDIO Command signal
AD29	ALT5	SD2_CD__ GPIO2_IO12	P35	SDIO_CD#	SDIO_CD#	SDIO card detect
AB29	ALT0	SD2_CLK__ USDHC2_CLK	P36	SDIO_CK	SDIO_CK	SDIO Clock Signal
AD28	ALT5	SD2_RESET_B__ GPIO2_IO19	P37	SDIO_PWR_EN	SDIO_PWREN	SD card power enable

Note:

1. The *SDIO* card power should be switched on the Carrier board and the *SDIO* lines should be *ESD* protected. The *SMARC* Evaluation Carrier schematic is useful as an implementation reference.
2. If *SD* boot up function is required, the pull-up resistor to 3.3V of *SDIO_PWR_EN* # should be *4.7k* or less.
3. *SDIO_WP* and *SDIO_CD#* are 10k pull up to 3.3V on module.

2.1.13.1. SDIO Card (4 bit) Interface

The Carrier SDIO Card can be selected as the Boot Device (See section 4.3).

Edge Golden Finder Signal Name	*Direction*	*Type Tolerance*	*Description*
SDIO_D[0:3]	*Bi-Dir*	*CMOS 3.3V*	*4 bit data path*
SDIO_CMD	*Bi-Dir*	*CMOS 3.3V*	*Command Line*
SDIO_CK	*Output*	*CMOS 3.3V*	*Clock*
SDIO_WP	*Input*	*CMOS 3.3V*	*Write Protect*
SDIO_CD#	*Input*	*CMOS 3.3V*	*Card Detect*
SDIO_PWR_EN	*Output*	*CMOS 3.3V*	*SD Card Power Enable*

Note:

SD Cards are not typically available with a 1.8V *I/O* voltage. The Module SD Card *I/O* level is specified as 3.3V and **not** *CMOS 1.8V*.

2.1.14 SPI/SPI1 Interface

The *SMARC-iMX8MP* module supports two *NXP i.MX8M Plus SPI* interfaces (*ECSPI*) that are available off-Module for general purpose use. Each *SPI* channel has two chip-selects that can connect two SPI slave devices on each channel. *SPI* devices will share the "*SPI0_DIN*", "*SPI0_DO*" and "*SPI0_CK*" pins, but each device will have its own chip select pin. The chip select signal is a low active signal.

The *SPI* interface is diagramed below.

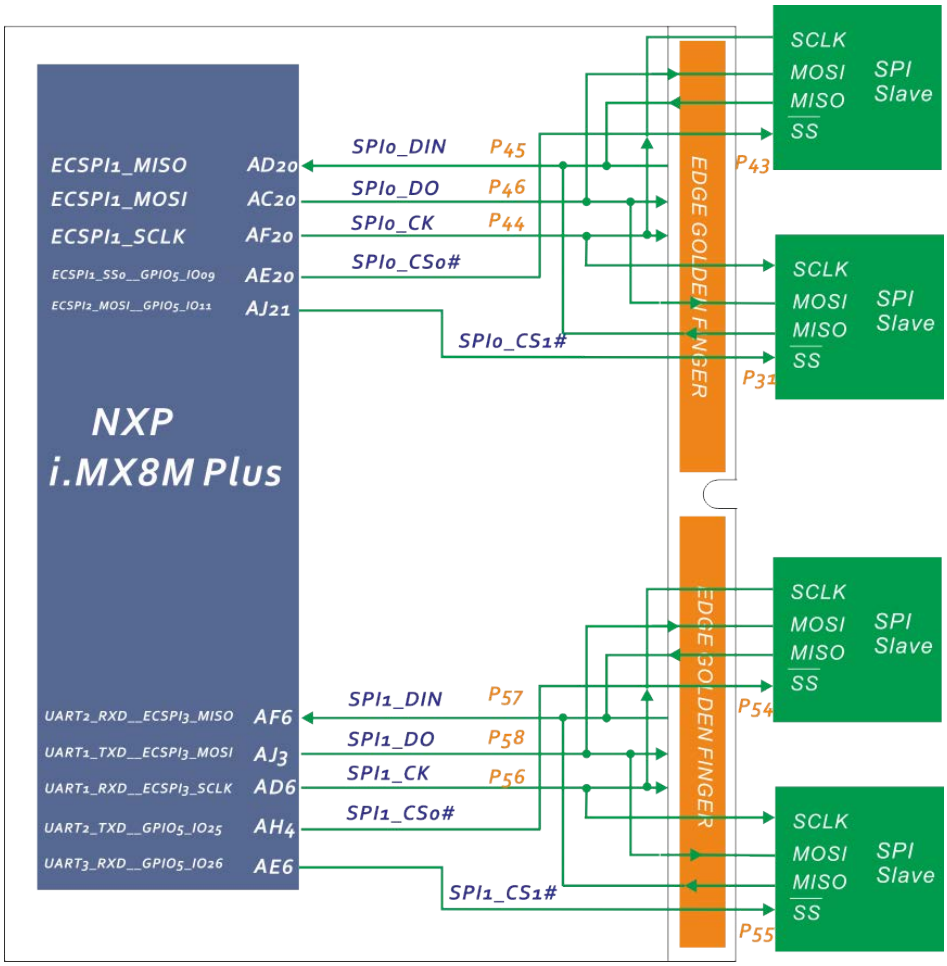


Figure 11 SPI Interface Block Diagram

SPI interface signals are exposed on the SMARC golden finger edge connector as shown below:

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
SPI0 Port						
AE20	ALT5	ECSPI1_SS0__ GPIO5_IO09	P43	SPI0_CS0#	SPI0_CS0#	SPI0 Master Chip Select 0 output
AJ21	ALT5	ECSPI2_MOSI__ GPIO5_IO11	P31	SPI0_CS1#	SPI0_CS1#	SPI0 Master Chip Select 1 output
AF20	ALT0	ECSPI1_SCLK__ ECSPI1_SCLK	P44	SPI0_CK	SPI0_SCLK	SPI0 Master Clock output
AD20	ALT0	ECSPI1_MISO__ ECSPI1_MISO	P45	SPI0_DIN	SPI0_DIN	SPI0 Master Data input (input to CPU, output from SPI device)
AC20	ALT0	ECSPI1_MOSI__ ECSPI1_MOSI	P46	SPI0_DO	SPI0_DO	SPI0 Master Data output (output from CPU, input to SPI device)

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
eSPI/SPI1 Port						
AH4	ALT5	UART2_TXD__ GPIO5_IO25	P54	ESPI_CS0#	ESPI_CS0#/ SPI1_CS0#	ESPI Master Chip Select 0 output
AE6	ALT5	UART3_RXD__ GPIO5_IO26	P55	ESPI_CS1#	ESPI_CS1#/ SPI1_CS1#	ESPI Master Chip Select 1 output
AD6	ALT1	UART1_RXD__ ECSPI3_SCLK	P56	ESPI_CK	ESPI_CK/ SPI1_CK	ESPI Master Clock output
AJ3	ALT1	UART2_RXD__ ECSPI3_MOSI	P58	ESPI_IO_0	ESPI_IO_0/ SPI1_DO	ESPI Master Data input (input to CPU, output from SPI device)
AF6	ALT1	UART1_TXD__ ECSPI3_MISO	P57	ESPI_IO_1	ESPI_IO_1/ SPI1_DI	ESPI Master Data output (output from CPU, input to SPI device)
			S56	ESPI_IO_2	ESPI_IO_2	Not Connected
			S57	ESPI_IO_3	ESPI_IO_3	Not Connected
			S58	ESPI_RESET#	ESPI_RESET#	Not Connected

2.1.14.1. SPI0 Signals

SMARC-iMX8MP does not support *SPI0* device boot up. The Carrier *SPI0* device cannot be selected as the Boot Device – see Section 4.3 Boot Select.

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
SPI0_CS0#	*Output*	*CMOS 1.8V*	*SPI0 Master Chip Select 0 output*
SPI0_CS1#	*Output*	*CMOS 1.8V*	*SPI0 Master Chip Select 1 output*
SPI0_CK	*Output*	*CMOS 1.8V*	*SPI0 Master Clock output*
SPI0_DIN	*Input*	*CMOS 1.8V*	*SPI0 Master Data input (input to CPU, output from SPI device)*
SPI0_DO	*Output*	*CMOS 1.8V*	*SPI0 Master Data output (output from CPU, input to SPI device)*

2.1.14.2. ESPI/SPI1 Signals

SMARC-iMX8MP does not support *ESPI* device boot up either. The Carrier *ESPI* device cannot be selected as the Boot Device – see Section 4.3 Boot Select.

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
ESPI_CS0#/ SPI1_CS0#	*Output*	*CMOS 1.8V*	*ESPI Master Chip Select 0 output*
ESPI_CS1#/ SPI1_CS1#	*Output*	*CMOS 1.8V*	*ESPI Master Chip Select 1 output*
ESPI_CK/ SPI1_CK	*Output*	*CMOS 1.8V*	*ESPI Master Clock output*
ESPI_IO_[0:1]/ SPI1_[DO:DIN]	*Bi-Dir*	*CMOS 1.8V*	*ESPI Master Data input/output*
ESPI_RESET#	*Output*	*CMOS 1.8V*	*Not Supported*
ESPI_ALERT[0:1]#	*Input*	*CMOC 1.8V*	*Not Supported*

2.1.15. I2S Interface

The *SMARC-iMX8MP* module uses *I2S* format for Audio signals. These signals are derived from the Synchronous Audio Interface (*SAI*) of the *NXP® i.MX8M Plus* processor. The Serial Audio Interface (SAI) implements a synchronous serial bus interface for connecting digital audio devices. It is by far the most common mechanism used to transfer two channels of audio data between devices within a system.

SMARC-iMX8MP supports two I2S instances (I2S0 and I2S2). I2S interface signals are exposed on the *SMARC-iMX8MP* golden finger edge connector as shown below:

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
AJ15	ALT0	SAI2_MCLK__ AUDIOMIX_SAI2 _MCLK	S38	AUDIO_MCK	AUD_MCLK	Master clock output to Audio codecs
I2S0 interface						
AJ17	ALT0	SAI2_TXFS__ AUDIOMIX_SAI2 _TX_SYNC	S39	I2S0_LRCK	I2S0_LRCK	Left& Right audio synchronization clock
AH16	ALT0	SAI2_TXD0__ AUDIOMIX_SAI2 _TX_DATA00	S40	I2S0_SDOUT	I2S0_SDOUT	Digital audio Output
AJ14	ALT0	SAI2_RXD0__ AUDIOMIX_SAI2 _RX_DATA00	S41	I2S0_SDIN	I2S0_SDIN	Digital audio Input
AH15	ALT0	SAI2_TXC__ AUDIOMIX_SAI2 _TX_BCLK	S42	I2S0_CK	I2S0_CK	Digital audio clock
I2S2 interface						
AC16	ALT0	SAI3_TXFS__ AUDIOMIX_SAI3 _TX_SYNC	S50	HDA_SYNC/ I2S2_LRCK	I2S2_LRCK	Left& Right audio synchronization clock
L26	ALT2	NAND_CE0_B__ AUDIOMIX_SAI3 _TX_DATA00	S51	HDA_SDO/ I2S2_SDOUT	I2S2_SDOUT	Digital audio Output
AF18	ALT0	SAI3_RXD__ AUDIOMIX_SAI3 _RX_DATA0	S52	HDA_SDI/ I2S2_SDIN	I2S2_SDIN	Digital audio Input
N25	ALT2	NAND_ALE__ AUDIOMIX_SAI3 _TX_BCLK	S53	HAD_CK/ I2S2_CK	I2S2_CK	Digital audio clock

Note:

SGTL5000 I2S audio codec is used in *EVK-STD-CARRIER-S20* evaluation carrier board.

2.1.15.1 I2S Signals

Edge Golden Finder Signal Name	*Direction*	*Type Tolerance*	*Description*
AUDIO_MCK	*Output*	*CMOS 1.8V*	*Master clock output to Audio codecs*
I2S0 Signals			
I2S0_LRCK	*Bi-Dir*	*CMOS 1.8V*	*Left& Right audio synchronization clock*
I2S0_SDOUT	*Output*	*CMOS 1.8V*	*Digital audio Output*
I2S0_SDIN	*Input*	*CMOS 1.8V*	*Digital audio Input*
I2S0_CK	*Bi-Dir*	*CMOS 1.8V*	*Digital audio clock*
I2S2 Signals			
I2S2_LRCK	*Bi-Dir*	*CMOS 1.8V*	*Left& Right audio synchronization clock*
I2S2_SDOUT	*Output*	*CMOS 1.8V*	*Digital audio Output*
I2S2_SDIN	*Input*	*CMOS 1.8V*	*Digital audio Input*
I2S2_CK	*Bi-Dir*	*CMOS 1.8V*	*Digital audio clock*

2.1.16. Asynchronous Serial Port (UARTs)

The *SMARC-iMX8MP* module supports four UARTs (*SER0:3*). UART *SER0* and *SER2* support flow control signals (*RTS#, CTS#*). UART *SER1* and *SER3* do not support flow control (*TX, RX* only). When working with software, *SER3* is used for *SMARC-iMX8MP* debugging console port.

The module asynchronous serial port signals have a *VDDIO* (*1.8V*) level signal swing. If the asynchronous ports are to interface with RS232 level devices, then a Carrier RS-232 transceiver is required. The logic side of the transceiver must be able to run at 1.8V levels. The selection of 1.8V compatible transceivers is a bit limited, although more are appearing with time. Two such devices are the Texas Instruments TRS3253E, and the Maxim MAX13235E, illustrated in the figures below. The TI part is more cost effective, but has a top speed of 1 Mbps. The MAX 13235E can operate at maximum speeds over 3 Mbps. The transceivers invert the polarity of the incoming and outgoing data and handshake lines.

The other alternative is to use a level-shift IC from 1.8V to 3.3V when designing carrier board and almost all transceivers available accept a 3.3V signal level: example includes the Texas Instruments MAX3243. Note that RS232 transceivers invert the signal; a logic ‘1’ is a negative voltage (-3.0V to -15V) and a logic ‘0’ a positive voltage (3.0V to 15V) on the RS232 line.

Asynchronous serial ports interface signals are exposed on the *SMARC* golden finger edge connector as shown below:

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	**Mode**	**Pin Name**	**Pin#**	**Pin Name**		
SER0 Port						
W28	ALT4	SD1_CLK__ UART1_DCE_TX	P129	SER0_TX	SER0_TX	Asynchronous serial port data out
W29	ALT4	SD1_CMD__ UART1_DCE_RX	P130	SER0_RX	SER0_RX	Asynchronous serial port data in
Y28	ALT4	SD1_DATA1__ UART1_DCE_CTS	P131	SER0_RTS#	SER0_RTS#	Request to Send handshake line for SER0
Y29	ALT4	SD1_DATA0__ UART1_DCE_RTS	P132	SER0_CTS#	SER0_CTS#	Clear to Send handshake line for SER0
SER1 Port						
AH5	ALT0	UART4_TXD__ UART4_DCE_TX	P134	SER1_TX	SER1_TX	Asynchronous serial port data out
AJ5	ALT0	UART4_RXD__ UART4_DCE_RX	P135	SER1_RX	SER1_RX	Asynchronous serial port data in

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
SER2 Port						
AA28	ALT4	SD1_DATA6__ UART3_DCE_TX	P136	SER2_TX	SER2_TX	Asynchronous serial port data out
U25	ALT4	SD1_DATA7__ UART3_DCE_RX	P137	SER2_RX	SER2_RX	Asynchronous serial port data in
W26	ALT4	SD1_STROBE__ UART3_DCE_CTS	P138	SER2_RTS#	SER2_RTS#	Request to Send handshake line for SER2
W25	ALT4	SD1_RESET_B__ UART3_DCE_RTS	P139	SER2_CTS#	SER2_CTS#	Clear to Send handshake line for SER2
SER3 Port (Debugging Port)						
V29	ALT4	SD1_DATA2__ UART2_DCE_TX	P140	SER3_TX	SER3_TX	Asynchronous serial port data out
V28	ALT4	SD1_DATA3__ UART2_DCE_RX	P141	SER3_RX	SER3_RX	Asynchronous serial port data in

2.1.16.1. UART Signals

Module pins for up to four asynchronous serial ports are defined. The ports are designated *SER0 – SER3*. Ports *SER0* and *SER2* are 4 wire ports (2 data lines and 2 handshake lines). Ports *SER1* and *SER3* are 2 wire ports (data only).

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
SER[0:3]_TX	*Output*	*CMOS 1.8V*	*Asynchronous serial port data out*
SER[0:3]_RX	*Input*	*CMOS 1.8V*	*Asynchronous serial port data in*
SER[0]_RTS#	*Output*	*CMOS 1.8V*	*Request to Send handshake line for SER0*
SER[0]_CTS#	*Input*	*CMOS 1.8V*	*Clear to Send handshake line for SER0*
SER[2]_RTS#	*Output*	*CMOS 1.8V*	*Request to Send handshake line for SER2*
SER[2]_CTS#	*Input*	*CMOS 1.8V*	*Clear to Send handshake line for SER2*

2.1.17. I2C Interface

There is a minimum configuration of I2C ports up to a maximum of 6 ports defined in the *SMARC* specification: *PM* (Power Management), *LCD* (Liquid Crystal Display), *GP* (General Purpose), *CAM0* (Camera 0), and CAM1 (Camera 1) and *HDMI. SMARC-iMX8MP* supports these six *I2C* in fast mode (400 KHz operation).
All *I2C* interfaces are implemented directly from *NXP i.MX8M Plus* processor interfaces.

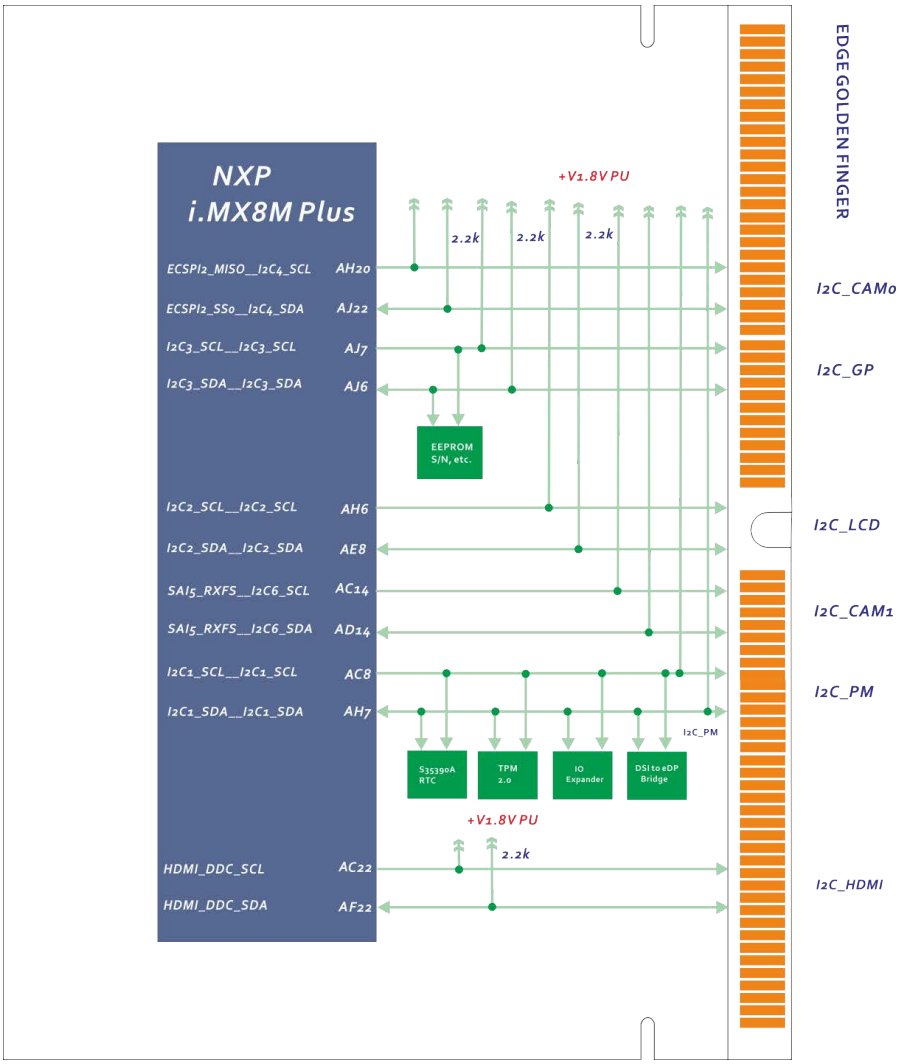


Figure 12 I2C Interface Block Diagram

This will be summarized below.

I2C Port		Primary Purpose	Alternative Use	I/O Voltage Level
Golden Finger Connector	i.MX8M Plus CPU			
I2C_PM	I2C1	Power Management support	System configuration management	CMOS 1.8V
I2C_GP	I2C3	General purpose use		CMOS 1.8V
I2C_LCD	I2C2	LCD display support, to read LCD display EDID EEPROMs (for parallel and LVDS LCD,)	General Purpose	CMOS 1.8V
I2C_CAM0	I2C4	Serial camera 0	General Purpose	CMOS 1.8V
I2C_CAM1	I2C6	Serial camera 1	General Purpose	CMOS 1.8V
HDMI_CTRL	HDMI_DDC	HDMI Control		CMOS 1.8V

Note:

1. The *2.2k* pull-up resistors for *I2C_SCL* and *I2C_SDA* signals are on module.

The *I2C* interface signals that are exposed on the *SMARC* golden finger edge connector as shown below:

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Note
Ball	**Mode**	**Pin Name**	**Pin#**	**Net Names**	
I2C_PM					
AC8	ALT0	I2C1_SCL__ I2C1_SCL	P121	I2C_PM_CK	Power management I2C bus clock
AH7	ALT0	I2C1_SDA__ I2C1_SDA	P122	I2C_PM_DAT	Power management I2C bus data
I2C_GP					
AJ7	ALT0	I2C3_SCL__ I2C3_SCL	S48	I2C_GP_CK	General purpose I2C bus clock
AJ6	ALT0	I2C3_SDA__ I2C3_SDA	S49	I2C_GP_DAT	General purpose I2C bus data
I2C_LCD					
AH6	ALT0	I2C2_SCL__ I2C2_SCL	S139	I2C_LCD_CK	LCD display I2C bus clock
AE8	ATL0	I2C2_SDA__ I2C2_SDA	S140	I2C_LCD_DAT	LCD display I2C bus data
I2C_CAM0					
AH20	ALT2	ECSPI2_MISO__ I2C4_SCL	S5	I2C_CAM0_CK	Camera 0 I2C bus clock
AJ22	ALT2	ECSPI2_SS0__ I2C4_SDA	S7	I2C_CAM0_DAT	Camera 0 I2C bus data
I2C_CAM1					
AC14	ALT3	SAI5_RXFS__ I2C6_SCL	S1	I2C_CAM1_CK	Camera 1 I2C bus clock
AD14	ALT3	SAI5_RXC__ I2C6_SDA	S2	I2C_CAM1_DAT	Camera 1 I2C bus data

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Note
Ball	Mode	Pin Name	Pin#	Net Names	
HDMI_CTRL					
AC22	N/A	HDMI_DDC_SCL	P105	HDMI_CTRL_CK	HDMI I2C bus clock
AF22	N/A	HDMI_DDC_SDA	P106	HDMI_CTRL_DAT	HDMI I2C bus data

Note:

All *I2C* bus defined in SMARC 2.0 specification are operated at 1.8V. The slave devices and their address details are listed in the following table:

#	Device	Description	Address (7-bit)	Address (8-bit)		Notes
				Read	Write	
I2C_GP						
1	*On Semiconductor CAT24C32*	*EEPROM*	*0x50*	*0xA1*	*0xA0*	*General purpose parameter EEPROM, Serial number, etc in PICMG EEEP format*
I2C_PM						
1	*NXP PCA6450CHN*	*PMIC*	*0x25*	*0x4B*	*0x4A*	*Power Management IC*
2.	*Seiko S-35390A*	*RTC*	*0x30*	*0xA1*	*0xA0*	*Real-Time Clock*
3	*ST ST33HTPH2E32AHC2*	*TPM*	*0x2E*	*0x5D*	*0x5C*	*TPM 2.0*
4	*TI TCA6408ARSV*	*IO Expander*	*0x20*	*0x41*	*0x40*	*I2C IO Expander*

Note:

On-module *EEPROM* has been moved from *I2C_PM* to *I2C_GP* at *SMARC 2.0 specification*.

2.1.18. CAN Bus Interface

The *FlexCAN* module in *i.MX8M Plus* processor is a communication controller implementing the CAN protocol according to the *ISO 11898-1* standard and CAN 2.0 B protocol specifications (*CAN-FD*). The Flexible Controller Area Network (*FlexCAN*) is a serial communications protocol which efficiently supports distributed real-time control with a high level of security. The *SMARC-iMX8MP* module supports two *CAN-FD* bus interfaces. *CAN-FD* interface signals are exposed on the *SMARC* golden finger edge connector as shown below:

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Note
Ball	Mode	Pin Name	Pin#	Net Names	
CAN0 BUS					
AD16	ALT6	SAI5_RXD1__ CAN1_TX	P143	CAN0_TX	CAN0 Transmit output
AF16	ALT6	SAI5_RXD2__ CAN1_RX	P144	CAN0_RX	CAN0 Receive input
CAN1 BUS					
AE14	ALT6	SAI5_RXD3__ CAN2_TX	P145	CAN1_TX	CAN1 Transmit output
AF14	ALT6	SAI5_MCLK__ CAN2_RX	P146	CAN1_RX	CAN1 Receive input

A CAN transceiver on carrier is necessary to adapt the signals from *SMARC* golden finger edge connector, which is TTL levels, to the physical layer used. Because the CAN bus system is typically used to connect multiple systems and is often run over very long distances, both power supply and signal path must be electrically isolated to meet a certain isolation level. Users can refer the "***SMARC Carrier Board Hardware Design Guide***" or CAN transceiver application note such as TI SLLA270 for more details.

2.1.18.1. CAN0 BUS Signals

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
CAN0_TX	Output	CMOS VDDIO	CAN0 Transmit output
CAN0_RX	Input	CMOS VDDIO	CAN0 Receive input

2.1.19.2. CAN1 BUS Signals

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
CAN1_TX	Output	CMOS VDDIO	CAN1 Transmit output
CAN1_RX	Input	CMOS VDDIO	CAN1 Receive input

Note:

Only industrial temperature grade *i.MX8M Plus* processors support *CAN-FD*. For commercial temperature grade *I.MX8M Plus* processors, it supports traditional *CAN* bus.

2.1.19. GPIOs

The *SMARC-iMX8MP* module supports *12* GPIOs, as defined by the *SMARC* specification. Specific alternate functions are assigned to some *GPIOs* such as *PWM / Tachometer* capability, Camera support, and HD Audio reset. All pins are capable of bi-directional operation. A default direction of operation is assigned, with half of them (*GPIO0 – GPIO5*) for use as outputs and the remainder (*GPIO6 – GPIO11*) as inputs by *SMARC* hardware specification.

GPIO signals are exposed on the *SMARC* golden finger edge connector as shown below:

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
GPIOs						
R25	ALT5	NAND_DATA00__ GPIO3_IO06	P108	GPIO0/CAM0_PWR#	GPIO0	Camera 0 Power Enable, active low output
L25	ALT5	NAND_DATA01__ GPIO3_IO07	P109	GPIO1/CAM1_PWR#	GPIO1	Camera 1 Power Enable, active low output
L24	ALT5	NAND_DATA02__ GPIO3_IO08	P110	GPIO2/CAM0_RST#	GPIO2	Camera 0 Reset, active low output
N24	ALT5	NAND_DATA03__ GPIO3_IO09	P111	GPIO3/CAM1_RST#	GPIO3	Camera 1 Reset, active low output
B5	ALT0	GPIO1_IO15 __ GPIO1_IO15	P112	GPIO4/HDA_RST#	GPIO4	HD Audio Reset, active low output
AD8	ALT5 ALT1	I2C4_SDA__ GPIO5_IO21/ PWM1_OUT	P113	GPIO5/PWM_OUT	GPIO5	PWM output
AH19	ALT5	SAI3_TXC__ GPIO5_IO00	P114	GPIO6/TACHIN	GPIO6	Tachometer input (used with the GPIO5 PWM)
AH18	ALT5	SAI3_TXD__ GPIO5_IO01	P115	GPIO7	GPIO7	
AJ18	ALT5	SAI3_RXC__ GPIO4_IO29	P116	GPIO8	GPIO8	
AE18	ALT5	SPDIF_TX__ GPIO5_IO03	P117	GPIO9	GPIO9	
AD18	ALT5	SPDIF_RX__ GPIO5_IO04	P118	GPIO10	GPIO10	
AC18	ALT5	SPDIF_EXT_CLK__ GPIO5_IO05	P118	GPIO11	GPIO11	

2.1.19.1. GPIO Signals

Twelve Module pins are allocated for *GPIO* (general purpose input / output) use. All pins are capable of bi-directional operation. By *SMARC* specification, *GPIO0 – GPIO5* are recommended for use as outputs and the remainder (*GPIO6 – GPIO11*) as inputs.
At Module power-up, the state of the GPIO pins may not be defined, and may briefly be configured in the "wrong" state, before boot loader code corrects them. Carrier designers should be aware of this and plan accordingly. All GPIO pins are capable of generating interrupts. The interrupt characteristics (edge or level sensitivity, polarity) are generally configurable in the *i.MX8M Plus* register set.

Edge Golden Finder Signal Name	Preferred Direction	Type Tolerance	Description
GPIO0/CAM0_PWR#	Output	CMOS 1.8V	Camera 0 Power Enable, active low output
GPIO1/CAM1_PWR#	Output	CMOS 1.8V	Camera 1 Power Enable, active low output
GPIO2/CAM0_RST#	Output	CMOS 1.8V	Camera 0 Reset, active low output
GPIO3/CAM1_RST#	Output	CMOS 1.8V	Camera 1 Reset, active low output
GPIO4/HDA_RST#	Output	CMOS 1.8V	HD Audio Reset, active low output
GPIO5/PWM_OUT	Output	CMOS 1.8V	PWM output
GPIO6/TACHIN	Input	CMOS 1.8V	Tachometer input (used with the GPIO5 PWM)
GPIO7/PCAM_FLD	Input	CMOS 1.8V	
GPIO8/CAN0_ERR#	Input	CMOS 1.8V	
GPIO9/CAN1_ERR#	Input	CMOS 1.8V	
GPIO10	Input	CMOS 1.8V	
GPIO11	Input	CMOS 1.8V	

2.1.20 Watchdog Timer Interface

i.MX8M Plus features an internal *WDT*. Embedian's Linux kernel enables the internal *i.MX8M Plus WDT* and makes this functionality available to users through the standard Linux Watchdog API.
A description of the API is available following the link below:
http://www.kernel.org/doc/Documentation/watchdog/watchdog-api.txt

WDT signals are exposed on the *SMARC* golden finger edge connector as shown below:

NXP i.MX8M Plus CPU			SMARC-iMX8MP Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
Watchdog Timer						
AC10	ALT5	SAI1_RXD0__ GPIO4_IO02	S145	WDT_TIME_ OUT#	WDT_TIME_OUT#	Watchdog-Timer Output

2.1.21 JTAG

The following figure shows the *SMARC-iMX8MP JTAG* connectors location and pin out.

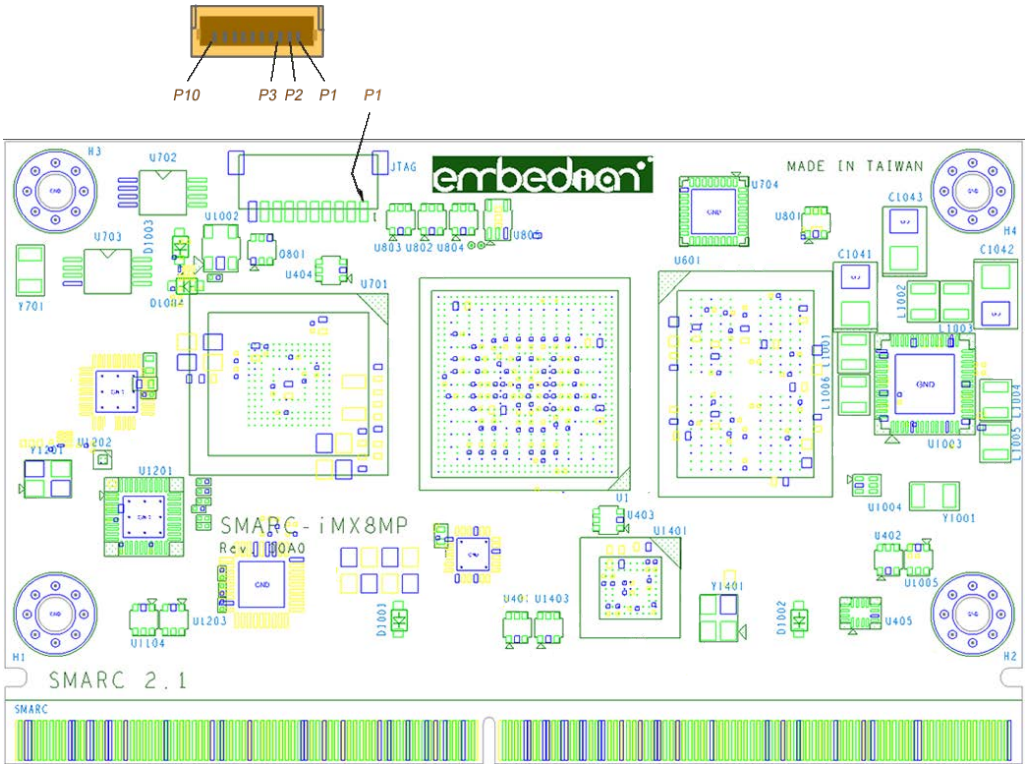


Figure 13 JTAG Connector Location and Pinout

JTAG functions for CPU debug and test are implemented on separate small form factor connector (CN3: *JST SM10B-SRSS-TB*, 1mm pitch R/A SMD Header). The *JTAG* pins are used to allow test equipment and circuit emulators to have access to the Module CPU. The pin-outs shown below are used:

NXP i.MX8M Plus CPU			JTAG(Connector: JST SM10B-SRSS-TB, 1mm pitch R/A SMD Header)		Type	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
JTAG						
			1	VDD_33A	Power	JTAG I/O Voltage (sourced by Module)
			2	nTRST	I	JTAG Reset, active low
G14	ALT0	JTAG_TMS	3	TMS	I	JTAG mode select
F14	ALT0	JTAG_TDO	4	TDO	O	JTAG data out
G16	ALT0	JTAG_TDI	5	TDI	I	JTAG data in
G18	ALT0	JTAG_TCK	6	TCK	I	JTAG clock
			7	RTCK	I	JTAG return clock
			8	GND	Ground	Ground
			9	MFG_Mode#	I	Pulled low to allow in-circuit SPI ROM update
			10	GND	Ground	Ground

2.1.22 Boot ID EEPROM

The *SMARC-iMX8MP* module includes an I2C serial *EEPROM* available on the *I2C_GP* bus. An On Semiconductor *24C32* or equivalent *EEPROM* is used in the module. The device operates at 1.8V. The Module serial *EEPROM* is placed at I2C slave addresses *A2 A1 A0* set to 0 (I2C slave address 50 hex, 7 bit address format or *A0 / A1* hex, 8 bit format) (for I2C *EEPROMs*, address bits *A6 A5 A4 A3* are set to binary 0101 convention).

The module serial *EEPROM* is intended to retain module parameter information, including serial number. The module serial *EEPROM* data structure conforms to the *PICMG® EEEP* Embedded *EEPROM* Specification.

Note:

The *EEPROM ID* memory layout is now follow the mainline and as follows.

Name	Size (Bytes)	Contents
Header	4	MSB 0xEE3355AA LSB
Board Name	8	Name for Board in ASCII "SM8MPQ6G" = Embedian SMARC-iMX8MP Computer on Module with Quad Core and 6GB LPDDR4 Configuration "SM8MPQ4G" = Embedian SMARC-iMX8MP Computer on Module with Quad Core and 4GB LPDDR4 Configuration "SM8MPD4G" = Embedian SMARC-iMX8MP Computer on Module with Dual Core and 4GB LPDDR4 Configuration
Version	4	Hardware version code for version in ASCII "00A0" = rev. A0
Serial Number	12	Serial number of the board. This is a 12 character string which is: WWYYM8QMnnnn Where: WW = 2 digit week of the year of production YY = 2 digit year of production M8 = Module iMX8 QM/QP= QuadMax or QuadPlus Core nnnn = incrementing board number

Name	Size (Bytes)	Contents
Configuration Option	32	Codes to show the configuration setup on this board. These 32 bytes are reserved by default.
MAC Address	6	Ethernet MAC Address (10:0D:32:XX:XX:XX)
MAC Address	6	Ethernet MAC Address for 2nd LAN (10:0D:32:XX:XX:XX)
Available	32720	Available space for other non-volatile codes/data

2.2 SMARC-iMX8MP Debug

2.2.1. Serial Port Debug

SMARC module has 4 serial output ports, *SER0, SER1, SER2 and SER3*. Out of these 4 serial ports, *SER3* is set as the serial debug port use for *i.MX8M Plus* from Embedian. Users can change to any port they want to from *u-boot* defconfig file. *SER3* is exposed (along with all other serial ports available on the module) in the *SMARC-iMX8MP* Evaluation Carrier. The default baud rate setting is *115,200 8N1*.
SER3 pin out of the *SMARC-iMX8MP* is shown below:

NXP i.MX8M Plus CPU		SMARC-iMX8MP Edge Golden Finger		Net Names	Notes
mode	Pin Name	Pin#	Pin Name		
SER3 (Debugging Port)					
ALT4	SD1_DATA2__ UART2_DCE_TX	P140	SER3_TX	SER3_TX	Asynchronous serial port data out
ALT4	SD1_DATA3__ UART2_DCE_RX	P141	SER3_RX	SER3_RX	Asynchronous serial port data in

2.3 Mechanical Specifications

2.3.1. Module Dimensions

The *SMARC-iMX8MP* complies with *SMARC* Hardware Specification in an 82mm x 50 mm form factor.

2.3.2. Height on Top

2.9mm maximum (without PCB) complied with *SMARC* specification defines as 3mm as the maximum.

2.3.3. Height on Bottom

0.9mm maximum (without PCB) complied with *SMARC* specification defines as 1.3mm as the maximum.

2.3.4. Mechanical Drawings

The mechanical information is shown in Figure 14: *SMARC-iMX8MP* Mechanical Drawings (Top View) and Figure 15: *SMARC-iMX8MP* Mechanical Drawings (Bottom View))

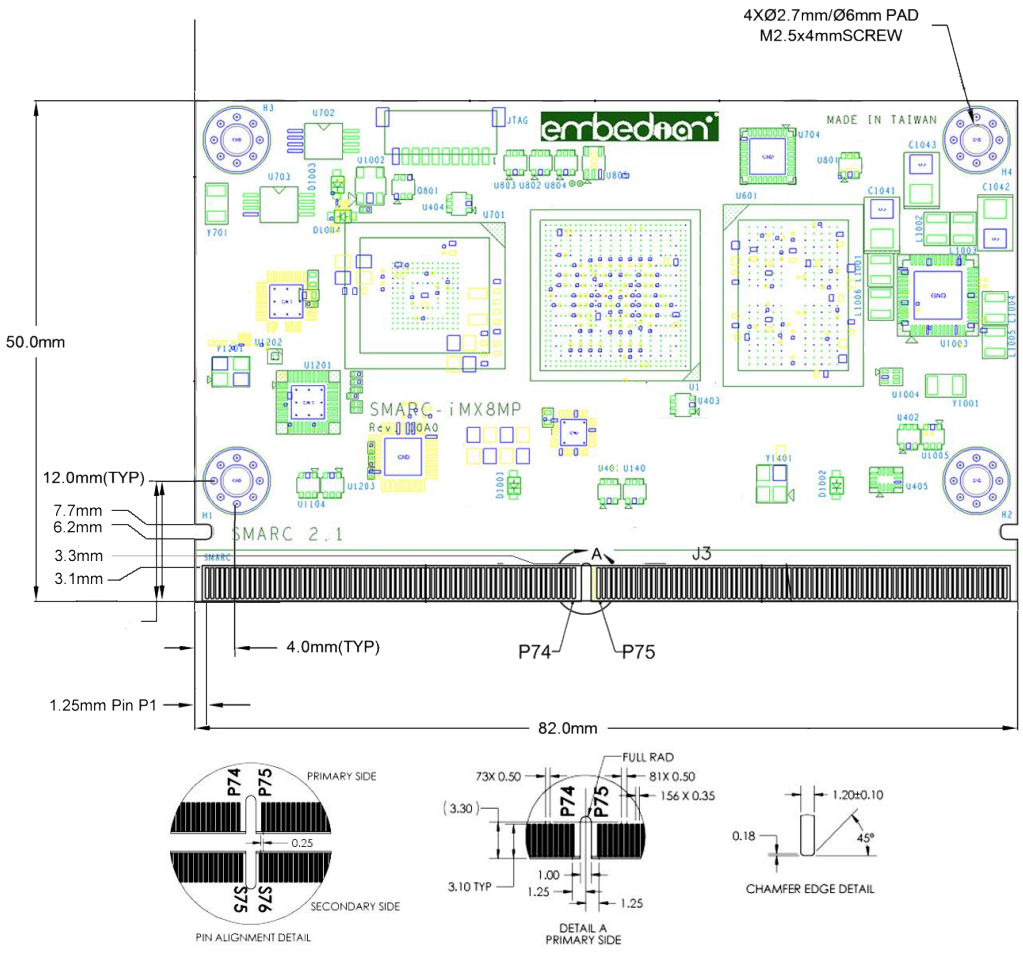


Figure 14 SMARC-iMX8MP Mechanical Drawings (Top View)

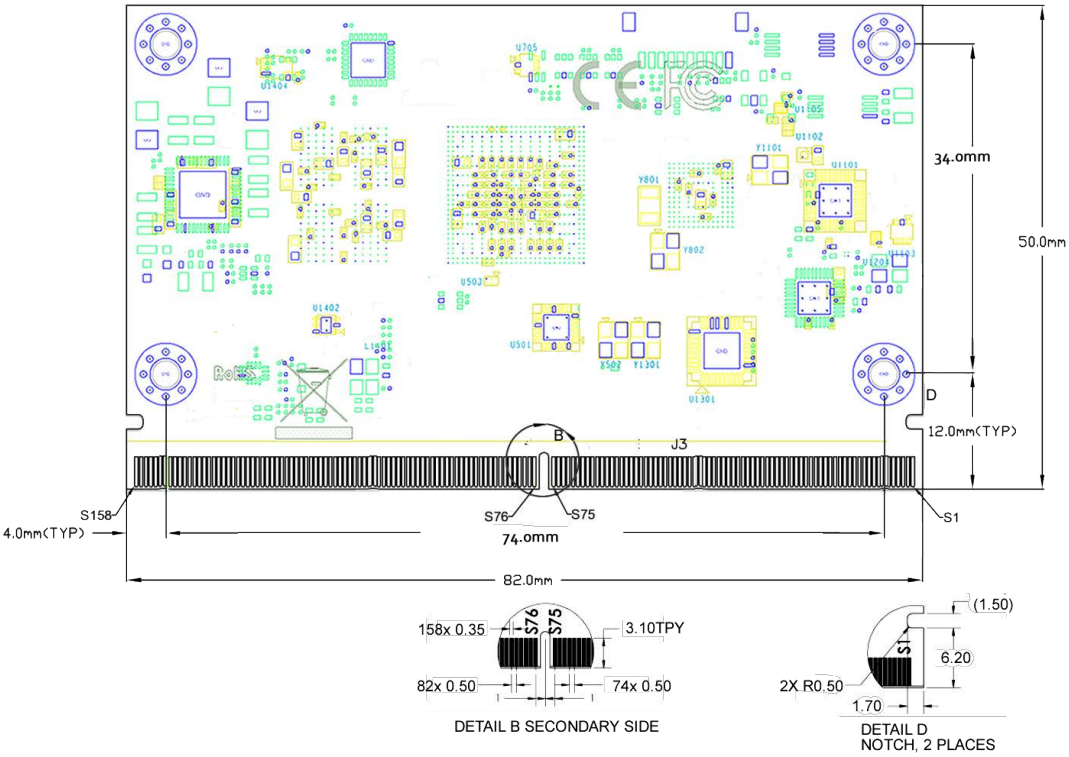


Figure 15 SMARC-iMX8MP Mechanical Drawings (Bottom View)

The figure on the following page details the 82mm x 50mm Module mechanical attributes, including the pin numbering and edge finger pattern.

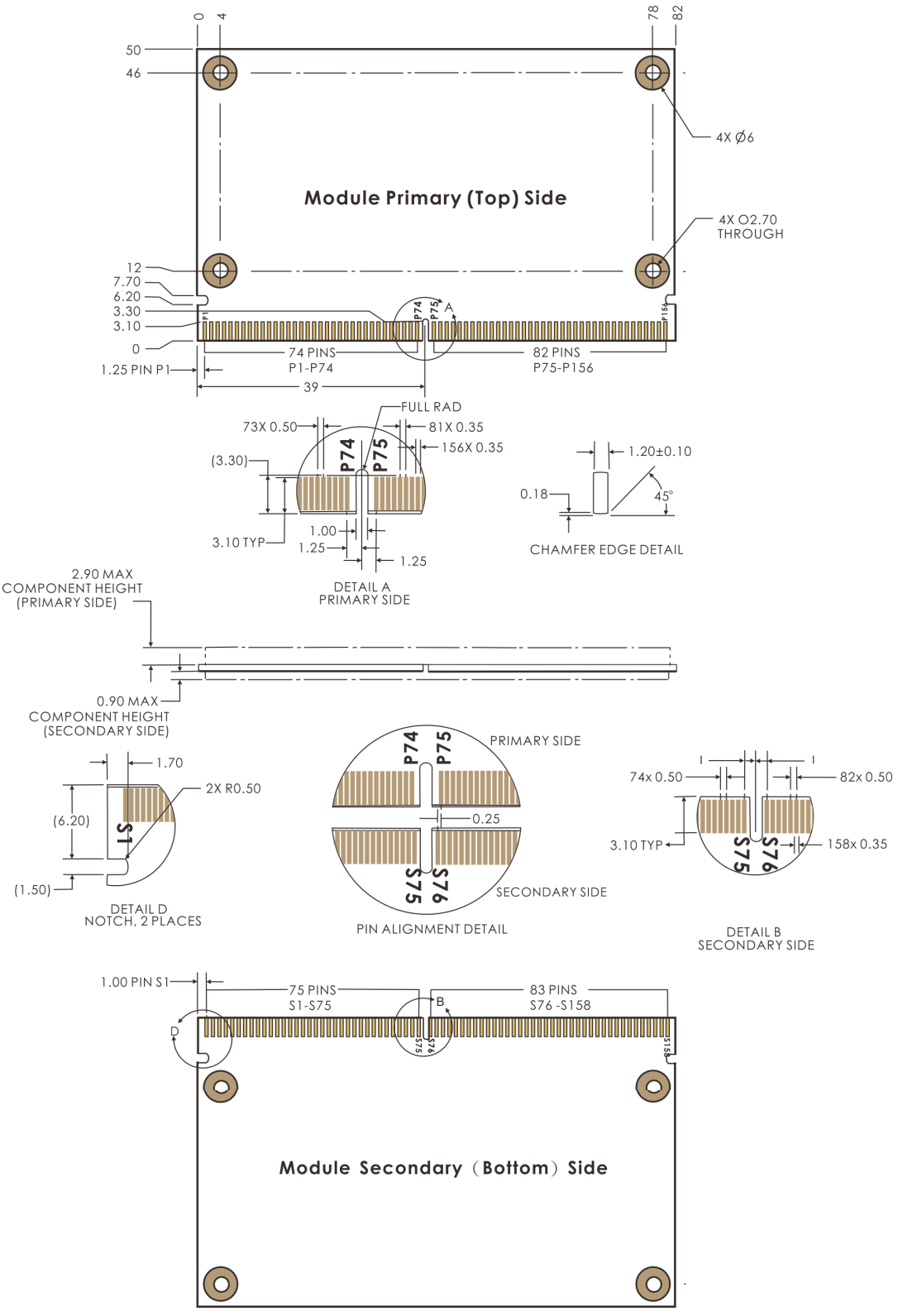


Figure 16 SMARC-iMX8MP Module Mechanical Outline

Top side major component (IC and Connector) information is shown in Figure 17: *SMARC-iMX8MP* Top side components.

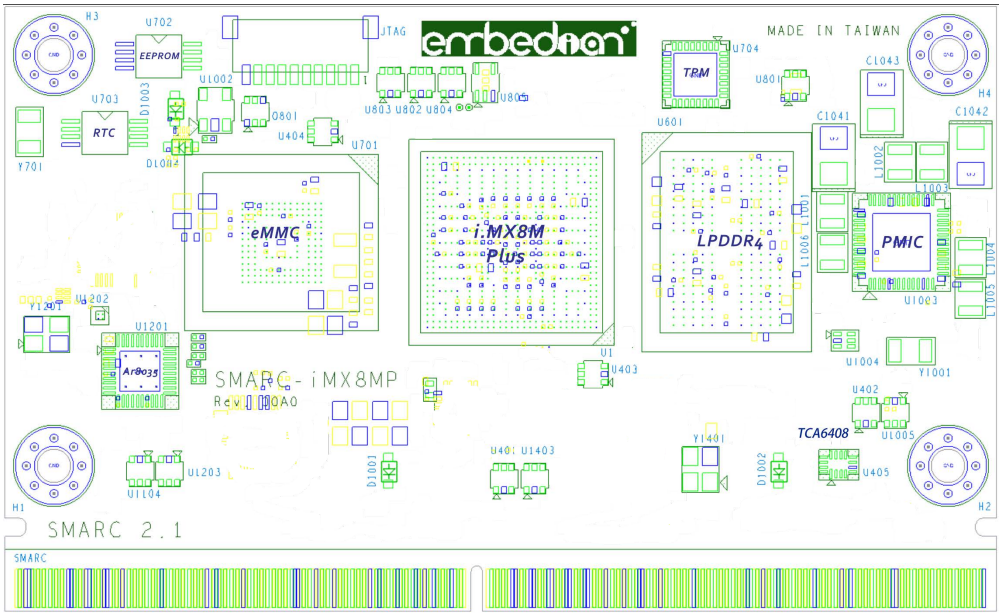


Figure 17 SMARC-iMX8MP Top Side Components

Bottom side major component (IC and Connector) information is shown in Figure 18: *SMARC-iMX8MP* Bottom side components.

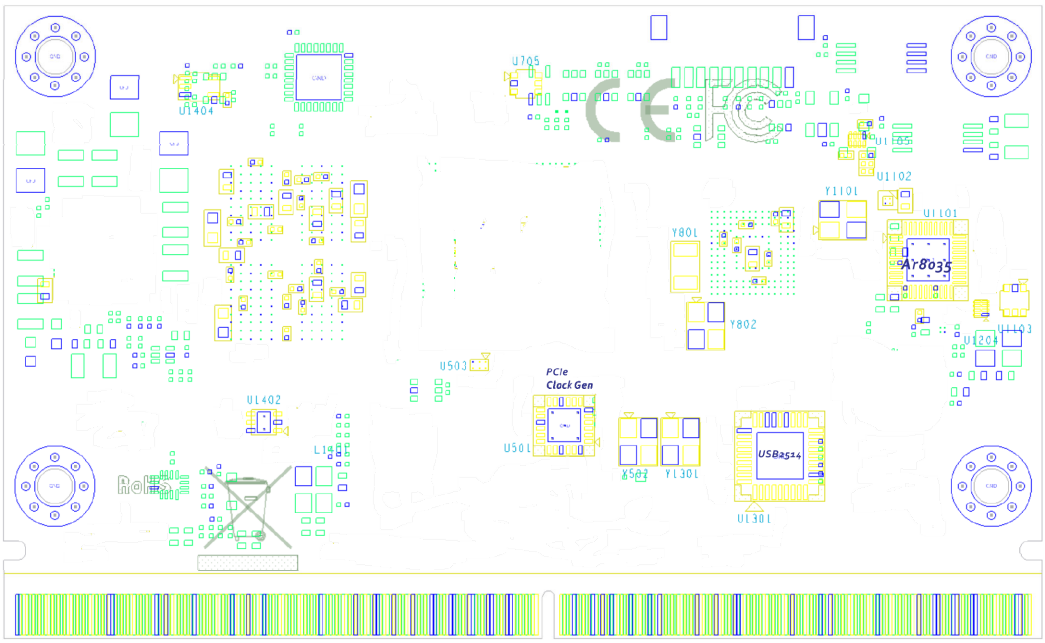


Figure 18 SMARC-iMX8MP Bottom Side Components

SMARC-iMX8MP height information from Carrier board Top side to tallest Module component is shown in Figure 19: *SMARC-iMX8MP* Minimum "Z" Height:

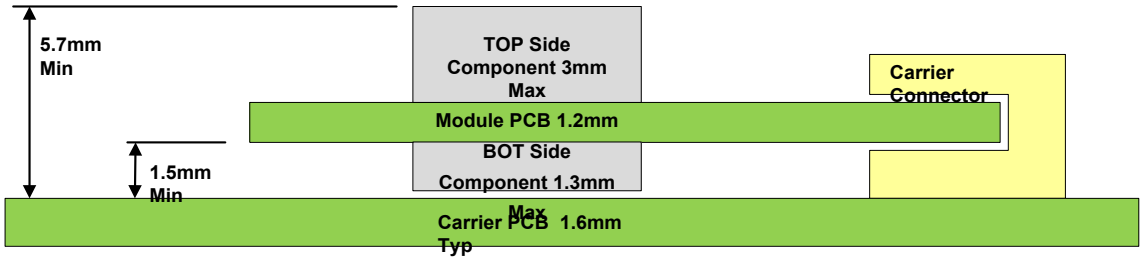


Figure 19 SMARC-iMX8MP Minimum "Z" Height

The *SMARC* connector board-to-board stack heights that are available may result in the use of non-standard spacer lengths. The board-to-board stack heights available include 1.5mm, 2.7mm and 5mm. Of these three, only the spacer for the 5mm stack would likely be a standard length.

When a 1.5mm stack height Carrier board connector is used, there shall not be components on the Carrier board Top side in the Module region. Additionally, when 1.5mm stack height connectors are used, there should not be PCB traces on the Carrier top side in the Module shadow. This is to prevent possible problems with metallic Module heat sink attachment hardware that may protrude through the Module.

If Carrier board components are required in this region, then the Carrier components must be on the Carrier Bottom side, or a taller Module-to-Carrier connector may be used. Stack heights of 2.7mm, 3mm, 5mm and up are available.

2.3.5. Carrier Board Connector PCB Footprint

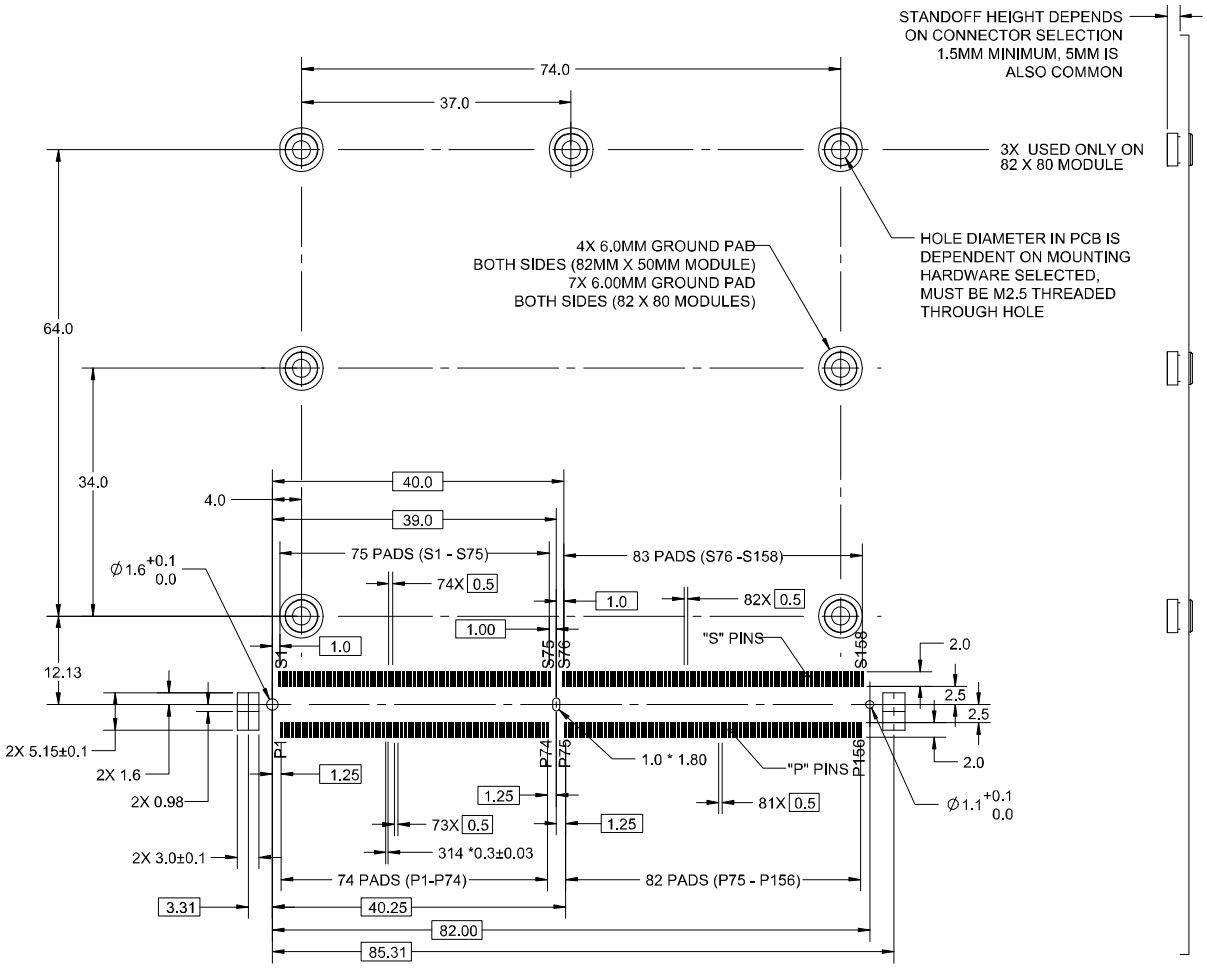


Figure 20 Carrier Board Connector PCB Footprint

Note:

The hole diameter for the 4 holes (82mm x 50mm Module) or 7 holes (82mm x 80mm Module) depends on the spacer hardware selection. See the section below for more information on this.

2.3.6. Module Assembly Hardware

The *SMARC-iMX8MP* module is attached to the carrier with four M2.5 screws. A 4mm length screw is usually used. The attachment holes are located on the corners of the module. Attachment holes have a 6mm diameter pad, 2.7 mm dia drill hole as shown Figure 14: *SMARC-iMX8MP* Mechanical Drawings (Top View)

2.3.7. Carrier Board Standoffs

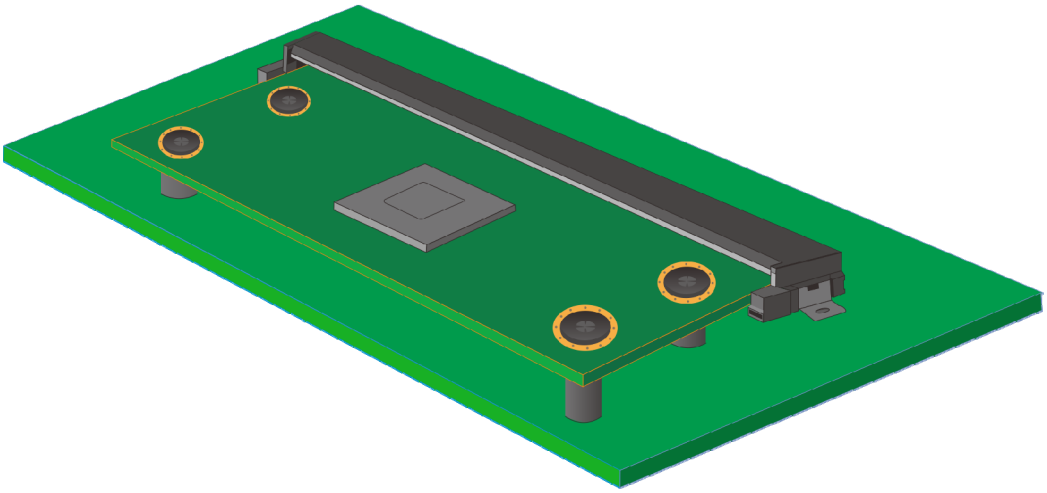


Figure 21 Screw Fixation

Standoffs secured to the Carrier board are expected. The standoffs are to be used with M2.5 hardware. Most implementations will use Carrier board standoffs that have M2.5 threads (as opposed to clearance holes). A short M2.5 screw and washer, inserted from the Module top side, secures the Module to the Carrier board threaded standoff.

The *SMARC* connector board-to-board stack heights that are available may result in the use of non-standard spacer lengths. The board-to-board stack heights available include 1.5mm, 2.7mm and 5mm. Of these three, only the

spacer for the 5mm stack would likely be a standard length.

Penn Engineering and Manufacturing (PEM) (www.pemnet.com) makes surface mount spacers with M2.5 internal threads. The product line is called SMTSO (“surface mount technology stand offs”). The shortest standard length offered is 2mm. A custom part with 1.5mm standoff length, M2.5 internal thread, and 5.56mm standoff OD is available from PEM. The Carrier PCB requires a 4.22mm hole and 6.2mm pad to accept these parts.

Other vendors such as RAF Electronic Hardware (www.rafhdwe.com) offer M2.5 compatible swaged standoffs. Swaged standoffs require the use of a press and anvil at the CM. Their use is common in the industry. The standoff OD and Carrier PCB hole size requirements are different from the PEM SMTSO standoffs described above.

2.3.8. Carrier Connector

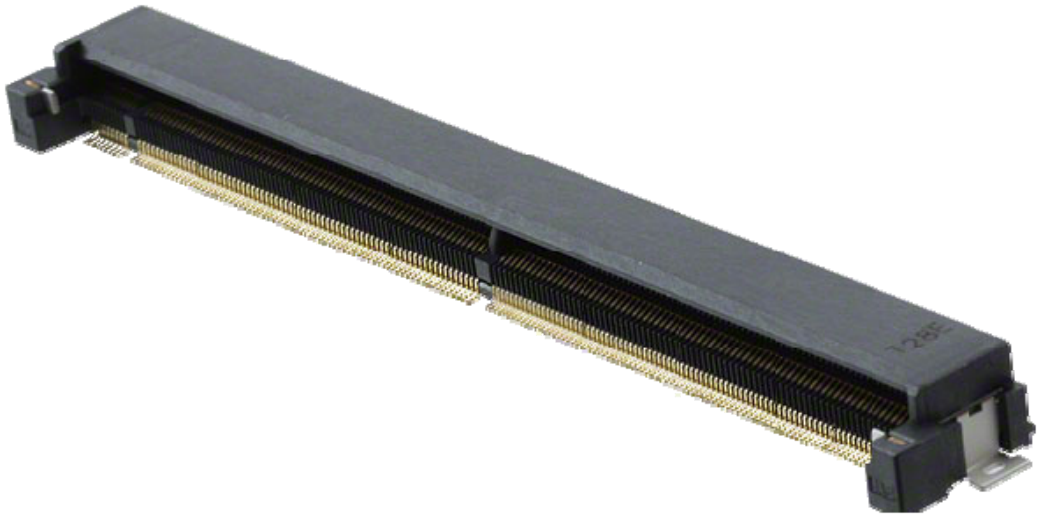


Figure 22 MXM3 Carrier Connector

The Carrier board connector is a 314 pin 0.5mm pitch right angle part designed for use with 1.2mm thick mating PCBs with the appropriate edge finger pattern. The connector is commonly used for MXM3 graphics cards. The *SMARC* Module uses the connector in a way quite different from the *MXM3* usage.

Vender	Vendor P/N	Stack Height	Body Height	Contact Plating	Pin Style	Body Color
Foxconn	AS0B821-S43B - *H	1.5mm	4.3mm	Flash	Std	Black
Foxconn	AS0B821-S43N - *H	1.5mm	4.3mm	Flash	Std	Ivory
Foxconn	AS0B826-S43B - *H	1.5mm	4.3mm	10 u-in	Std	Black
Foxconn	AS0B826-S43N - *H	1.5mm	4.3mm	10 u-in	Std	Ivory
Lotes	AAA-MXM-008-P04_A	1.5mm	4.3mm	Flash	Std	Tan
Lotes	AAA-MXM-008-P03	1.5mm	4.3mm	15 u-in	Std	Tan
Speedtech	B35P101-02111-H	1.56mm	4.0mm	Flash	Std	Black
Speedtech	B35P101-02011-H	1.56mm	4.0mm	Flash	Std	Tan
Speedtech	B35P101-02112-H	1.56mm	4.0mm	10 u-in	Std	Black
Speedtech	B35P101-02012-H	1.56mm	4.0mm	10 u-in	Std	Tan
Speedtech	B35P101-02113-H	1.56mm	4.0mm	15 u-in	Std	Black
Speedtech	B35P101-02013-H	1.56mm	4.0mm	15 u-in	Std	Tan
Aces	91781-314 2 8-001	2.7mm	5.2mm	3 u-in	Std	Black

Vender	Vendor P/N	Stack Height	Body Height	Contact Plating	Pin Style	Body Color
Foxconn	AS0B821-S55B - *H	2.7mm	5.5mm	Flash	Std	Black
Foxconn	AS0B821-S55N - *H	2.7mm	5.5mm	Flash	Std	Ivory
Foxconn	AS0B826-S55B - *H	2.7mm	5.5mm	10 u-in	Std	Black
Foxconn	AS0B826-S55N - *H	2.7mm	5.5mm	10 u-in	Std	Ivory
Speedtech	B35P101-02121-H	2.76mm	5.2mm	Flash	Std	Black
Speedtech	B35P101-02021-H	2.76mm	5.2mm	Flash	Std	Tan
Speedtech	B35P101-02122-H	2.76mm	5.2mm	10 u-in	Std	Black
Speedtech	B35P101-02022-H	2.76mm	5.2mm	10 u-in	Std	Tan
Speedtech	B35P101-02123-H	2.76mm	5.2mm	15 u-in	Std	Black
Speedtech	B35P101-02023-H	2.76mm	5.2mm	15 u-in	Std	Tan
Foxconn	AS0B821-S78B - *H	5.0mm	7.8	Flash	Std	Black
Foxconn	AS0B821-S78N - *H	5.0mm	7.8	Flash	Std	Ivory
Foxconn	AS0B826-S78B - *H	5.0mm	7.8	10 u-in	Std	Black
Foxconn	AS0B826-S78N - *H	5.0mm	7.8	10 u-in	Std	Ivory
Yamaichi [1]	CN113-314-2001	5.0mm	7.8	0.3 u-meter	Std	Black

Other, taller stack heights may be available from these and other vendors. Stack heights as tall as 11mm are shown on the Aces web site.

Note:

1. *Yamaichi CN113-314-2001* is automotive grade.
2. The vendor drawings for the connectors listed above show a PCB footprint pattern for use with an MXM3 graphics card. This footprint, and the associated pin numbering, is not suitable for *SMARC* use. The MXM3 standard gangs large groups of pins together to provide ~80W capable power paths needed for X86 graphics cards. The *SMARC* module “ungangs” these pins to allow more signal pins. Footprint and pin numbering information for application of this 314 pin connector to *SMARC* is given in the sections below.

2.3.9. Module Cooling Solution—Heat Spreader

A standard heat-spreader plate for use with the *SMARC* 82mm x 50mm form factor is described below. A standard heat spreader plate definition allows the customer to use a Module from multiple vendors.

The heat spreader plate is sized at 82mm x 42mm x 3mm, and sits 3mm above the *SMARC* Module. The heat spreader plate ‘Y’ dimension is deliberately set at 42mm and not 50mm, to allow the plate to clear the *SMARC MXM3* connector. The plate is shown in the figures below.

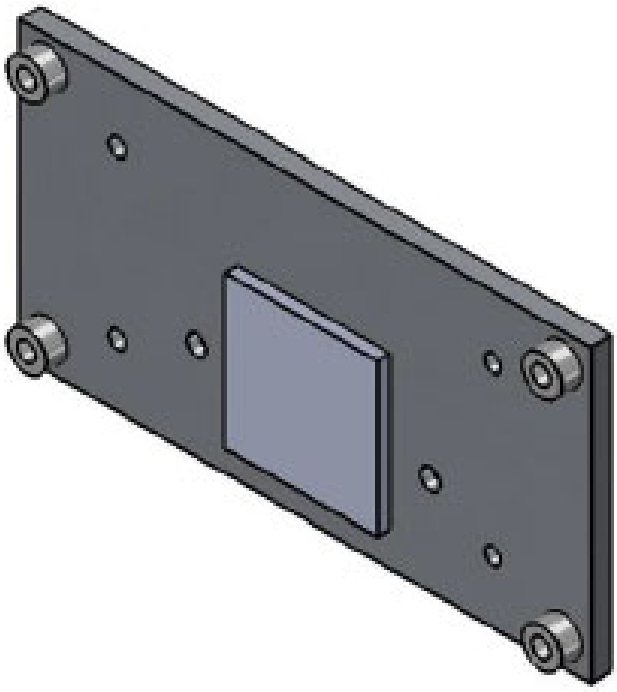


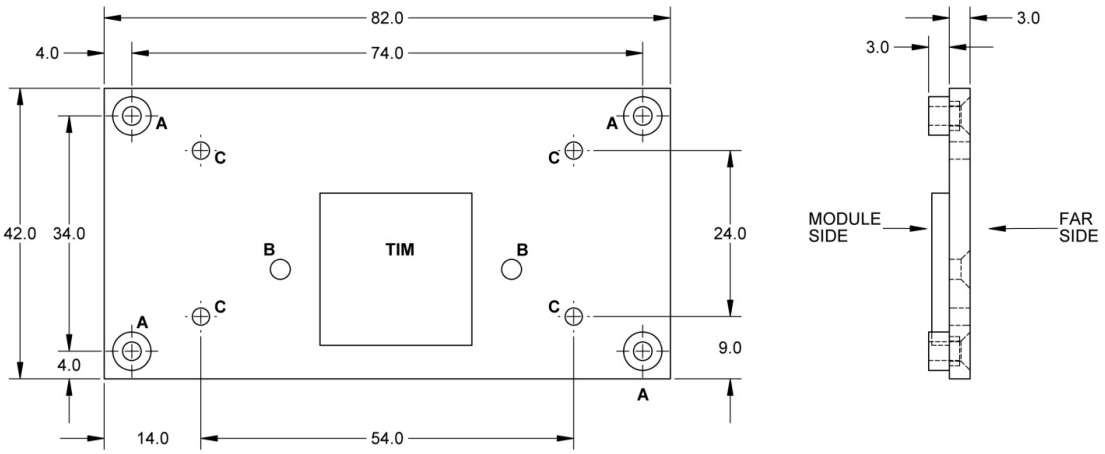
Figure 23 Heat Spreader

The internal square in the figure above is a thermally conductive and mechanically compliant Thermal Interface Material (or “TIM”). The exact X-Y position and Z thickness details of the TIM vary from design to design.

The two holes immediately adjacent to the TIM serve to secure the PCB in the SOC area and compress the TIM.

The four interior holes that are further from the center allow a heat sink to be attached to the heat spreader plate, or they can be used to secure the heat spreader plate to a chassis wall that serves as a heat sink.

Dimensions and further details may be found in the following figure.



Dimensions in the figure above are in millimeters. “TIM” stands for “Thermal Interface Material”. The TIM takes up the small gap between the SOC top and the Module - facing side of the heat spreader.

Hole Reference	Description	Size
A	*SMARC Module corner mounting holes Spacing determined by SMARC specification for 82mm x 50mm Modules.* *Typically these holes have 3mm length press fit or swaged clearance standoffs on the Module side.* *These holes are typically countersunk on the far side of the plate, to allow the heat spreader plate to be flush with a secondary heat sink.*	*Hole size depends on standoffs used. Standoff diameter must be compatible with SMARC Module mounting hole pad and hole size (6.0mm pads, 2.7mm holes on the Module). The holes and standoffs are for use with M2.5 screw hardware.* *The far side of these holes are counter-sunk to allow the attachment screw to be flush with the far side heat spreader surface.*
B	*Not Defined*	
C	*Fixed location holes to allow the attachment of a heat sink to the heat spreader, or to allow the heat spreader to be secured to a chassis wall that can serve as a heat sink.*	*M3 threaded holes*

2.4 Electrical Specifications

2.4.1. Supply Voltage

The *SMARC-iMX8MP* module operates over an input voltage range of 3.0V to 5.25V. Power is provided from the carrier through 10 power pins as defined by the *SMARC* specification.

Caution! A single 5V or 3.3V DC input is recommended.

2.4.2. RTC/Backup Voltage

3.0V RTC backup power is provided through the VDD_RTC pin from the carrier board. This connection provides back up power to the module PMIC. The RTC is powered via the primary system 3.3V supply during normal operation and via the VBAT power input, if it is present, during power-off.

2.4.3. No Separate Standby Voltage

The *SMARC-iMX8MP* does not have a standby power rail. Standby operation is powered through the main supply voltage rail, as defined in the *SMARC* specification.

2.4.4. Module I/O Voltage

The *SMARC-iMX8MP* module supports 1.8V (*SMARC* v2.0 compliant) level I/O voltage depending on the part number that users selected.

2.4.5. MTBF

The *SMARC-iMX8MP* System *MTBF* (hours) : >100,000 hours

The above *MTBF* (Mean Time Between Failure) values were calculated using a combination of manufacturer's test data, if the data was available, and a Bellcore calculation for the remaining parts. The Bellcore calculation used is "Method 1 Case 1". In that particular method the components are assumed to be operating at a 50 % stress level in a 40° C ambient environment and the system is assumed to have not been burned in. Manufacturer's data has been used wherever possible. The manufacturer's data, when used, is specified at 50°C, so in that sense the following results are slightly conservative. The MTBF values shown below are for a 40°C in an office or telecommunications environment. Higher temperatures and other environmental stresses (extreme altitude, vibration, salt water exposure, etc.) lower MTBF values.

2.4.6. Power Consumption

The power consumption values listed in this document were measured under a controlled environment. The hardware used for testing includes an *SMARC-iMX8MP* module, carrier board is *EVK-STD-CARRIER-S20* with *7-inch* LVDS display, SD card and USB keyboard. The carrier board was powered externally by a power supply unit so that it does not influence the power consumption value that is measured for the module. The USB keyboard was detached once the module was configured within the OS. All recorded values were averaged over a 30 second time period. The modules were cooled by the heatspreader specific to the module variants.

Each module was measured while running Yocto Sumo. To measure the worst case power consumption, the cooling solution was removed and the CPU core temperature was allowed to run between 95° and 100°C at 100% workload. The peak current value was then recorded. This value should be taken into consideration when designing the system's power supply to ensure that the power supply is sufficient during worst case scenarios.

Power consumption values were recorded during the following stages:

Yocto Gatesgarth

- Desktop Idle
- 100% CPU workload
- 100% CPU workload at approximately 100°C peak power consumption

Note: With the linux stress tool, we stressed the CPU to maximum frequency.
The table below provides additional information about the different variants offered by the *SMARC-iMX8MP*.

SMARC Part Number	*Desktop Idle*	*100% workload*	*Max. power consumption (Amp/Watts)*
SMARC-iMX8MP-Q-6G	*TBD*	*TBD*	*TBD*
SMARC-iMX8MP-Q-4G	*TBD*	*TBD*	*TBD*

2.5 Environmental Specifications

2.5.1. Operating Temperature

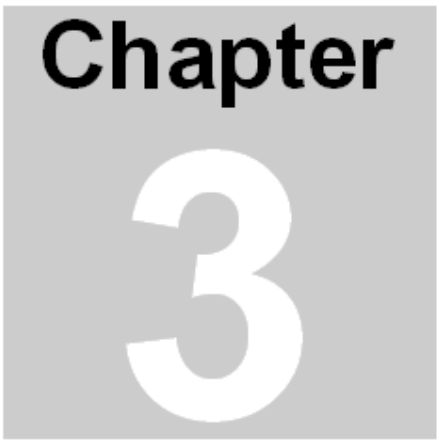
The *SMARC-iMX8MP* module operates from -40°C to 85°C air temperature, with a passive heat sink arrangement.

2.5.2. Humidity

Operating: 10% to 90% RH (non-condensing).
Non-operating: 5% to 95% RH (non-condensing).

2.5.3. ROHS/REACH Compliance

The *SMARC-iMX8MP* module is compliant to the *2002/95/EC RoHS* directive and *REACH* directive.



Connector PinOut

This Chapter gives detail pinout of *SMARC-iMX8MP* golden finger edge connector.

Section include：

- *SMARC-iMX8MP* Connector Pin Mapping

Chapter 3 Connector Pinout

The Module pins are designated as *P1 – P156* on the Module Primary (Top) side, and *S1 – S158* on the Module Secondary (Bottom) side. There are total of 314 pins on the Module. The connector is sometimes identified as a 321 pin connector, but 7 pins are lost to the key (4 on the primary side and 3 on secondary side).

The Secondary (Bottom) side faces the Carrier board when a normal or standard Carrier connector is used.

The *SMARC-iMX8MP* module pins are deliberately numbered as *P1 – P156* and *S1 – S158* for clarity and to differentiate the *SMARC* Module from MXM3 graphics modules, which use the same connector but use the pins for very different functions. MXM3 cards and MXM3 baseboard connectors use different pin numbering scheme.

3.1 SMARC-iMX8MP Connector Pin Mapping

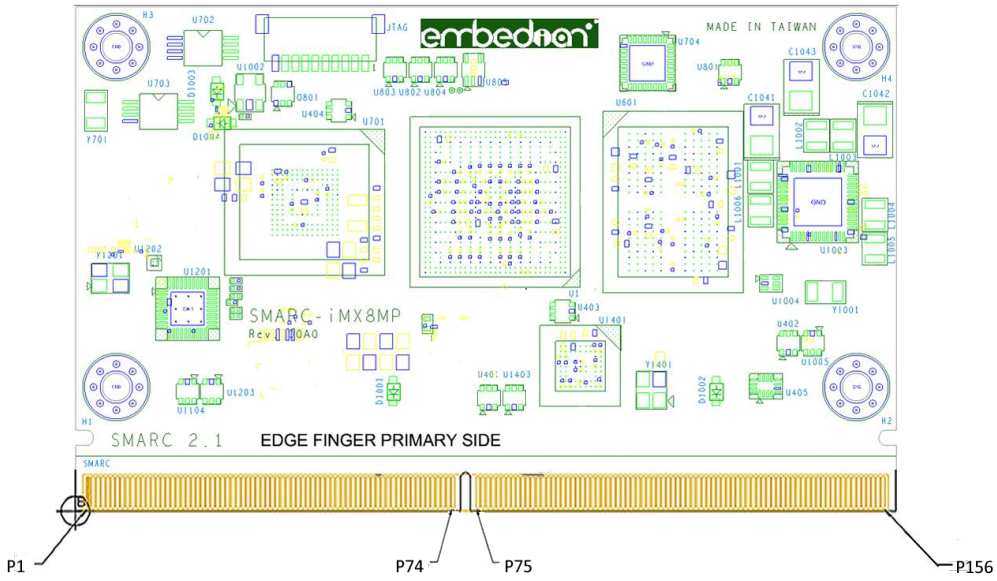


Figure 24 SMARC-iMX8MP edge finger primary pins

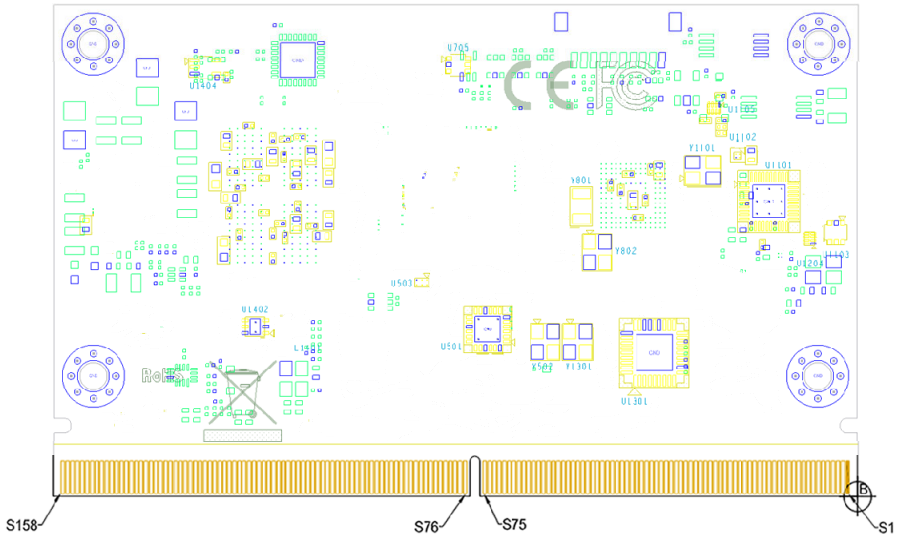


Figure 25 SMARC-iMX8MP edge finger secondary pins

The next tables describe each pin, its properties, and its use on the module and development board.

The "*SMARC Edge Finger*" column shows the connection of the signals defined in the *SMARC* specification. The "*NXP i.MX8M Plus CPU*" column shows the connection of the CPU signals on the module. The format of this column is "*Ball/Mode/Signal Name*" where "*Signal Name*" is the chip where the signals are connected, and "*Ball*" is the name of the pad where the signals are connected as they are defined in the *i.MX8M Plus* processor datasheet.

Pinout Legend

I	*Input*
O	*Output*
I/O	*Input or output*
P	*Power*
AI	*Analogue input*
AO	*Analogue output*
AIO	*Analogue Input or analogue output*
OD	*Open Drain Signal*
#	*Low level active signal*

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P1	SMB_ALERT_1V8#					Not used
P2	GND				P	Ground
P3	CSI1_CK+	D22		MIPI_CSI1_CLK_P	I	CSI1 differential clock inputs
P4	CSI1_CK-	E22		MIPI_CSI1_CLK_N	I	CSI1 differential clock inputs
P5	GBE1_SDP	AH8	ALT5	SAI1_RXC__ GPIO4_IO01		IEEE 1588 Trigger Signal.
P6	GBE0_SDP	B8	ALT0	GPIO1_IO09__ GPIO1_IO09		IEEE 1588 Trigger Signal.
P7	CSI1_RX0+	D18		MIPI_CSI1_D0_P	I	CSI1 differential data inputs 0 (positive)
P8	CSI1_RX0-	E18		MIPI_CSI1_D0_N	I	CSI1 differential data input 0 (negative)
P9	GND				P	Ground
P10	CSI1_RX1+	D20		MIPI_CSI1_D1_P	I	CSI1 differential data input 1 (positive)
P11	CSI1_RX1-	E20		MIPI_CSI1_D1_N	I	CSI1 differential data inputs 1 (negative)
P12	GND				P	Ground
P13	CSI1_RX2+	D24		MIPI_CSI1_D2_P		CSI1 differential data inputs 2 (positive)
P14	CSI1_RX2-	E24		MIPI_CSI1_D2_N		CSI1 differential data inputs 2 (negative)
P15	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P16	CSI1_RX3+	D26		MIPI_CSI1_D3_P		CSI1 differential data inputs 3 (positive)
P17	CSI1_RX3-	E26		MIPI_CSI1_D3_N		CSI1 differential data inputs 3 (negative)
P18	GND				P	Ground
P19	GbE0_MDI3-				AIO	Qualcomm AR8035 Differential Transmit/Receive Negative Channel 3
P20	GbE0_MDI3+				AIO	Qualcomm AR8035 Differential Transmit/Receive Positive Channel 3
P21	GbE0_LINK100#				O OD	Link Speed Indication LED for 100Mbps Could be able to sink 24mA or more Carrier LED current
P22	GbE0_LINK1000#				O OD	Link Speed Indication LED for 1000Mbps Could be able to sink 24mA or more Carrier LED current
P23	GbE0_MDI2-				AIO	Qualcomm AR8035 Differential Transmit/Receive Negative Channel 2
P24	GbE0_MDI2+				AIO	Qualcomm AR8035 Differential Transmit/Receive Positive Channel 2

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P25	GbE0_LINK_ACT#				O OD	Link / Activity Indication LED Driven low on Link (10, 100 or 1000 mbps) Blinks on Activity Could be able to sink 24mA or more Carrier LED current
P26	GbE0_MDI1-				AIO	Qualcomm AR8035 Differential Transmit/Receive Negative Channel 1
P27	GbE0_MDI1+				AIO	Qualcomm AR8035 Differential Transmit/Receive Positive Channel 1
P28	GbE0_CTREF				O	Qualcomm AR8035 Center tap reference voltage for GBE Carrier board Ethernet magnetic
P29	GbE0_MDI0-				AIO	Qualcomm AR8035 Differential Transmit/Receive Negative Channel 0
P30	GbE0_MDI0+				AIO	Qualcomm AR8035: Differential Transmit/Receive Positive Channel 0

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P31	SPI0_CS1#	AJ21	ALT5	ECSPI2_MOSI__ GPIO5_IO11	O	SPI0 Master Chip Select 1 output.
P32	GND				P	Ground
P33	SDIO_WP	AC26	ALT5	SD2_WP__ GPIO2_IO20	I	Write Protect
P34	SDIO_CMD	AB28	ALT0	SD2_CMD__ USDHC2_CMD	IO	Command Line
P35	SDIO_CD#	AD29	ALT5	SD2_CD_B__ GPIO2_IO12	I	Card Detect
P36	SDIO_CK	AB29	ALT0	SD2_CLK__ USDHC2_CLK	O	Clock
P37	SDIO_PWR_EN	AD28	ALT5	USDHC1_RESET_B__ GPIO2_IO19	O	SD card power enable
P38	GND				P	Ground
P39	SDIO_D0	AC28	ALT0	SD2_DATA0__ USDHC2_DATA0	IO	Data path
P40	SDIO_D1	AC29	ALT0	SD2_DATA1__ USDHC2_DATA1	IO	Data path
P41	SDIO_D2	AA26	ALT0	SD2_DATA2__ USDHC2_DATA2	IO	Data path
P42	SDIO_D3	AA25	ALT0	SD2_DATA3__ USDHC2_DATA3	IO	Data path

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P43	SPI0_CS0#	AE20	ALT5	ECSPI1_SS0__ GPIO5_IO09	O	SPI0 Master Chip Select 0 output,
P44	SPI0_CK	AF20	ALT0	ECSPI1_SCLK__ ECSPI1_SCLK	O	SPI0 Master Clock output
P45	SPI0_DIN	AD20	ALT0	ECSPI1_MISO__ ECSPI1_MISO	I	SPI0 Master Data input (input to CPU, output from SPI device)
P46	SPI0_DO	AC20	ALT0	ECSPI1_MOSI__ ECSPI1_MOSI	O	SPI0 Master Data output (output from CPU, input to SPI device)
P47	GND				P	Ground
P48	SATA_TX+					Not used
P49	SATA_TX-					Not used
P50	GND				P	Ground
P51	SATA_RX+					Not used
P52	SATA_RX-					Not used
P53	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P54	ESPI_CS0#/ SPI1_CS0#	AH4	ALT5	UART2_TXD__ GPIO5_IO25	O	SPI1 Master Chip Select 0 output
P55	ESPI_CS1#/ SPI1_CS1#	AF6	ALT5	UART3_RXD__ GPIO5_IO26	O	SPI1 Master Chip Select 1 output
P56	ESPI_CK/ SPI1_CK	AD6	ALT1	UART1_RXD__ ECSPI3_SCLK	O	SPI1 Master Clock output
P57	ESPI_IO_1/ SPI1_DIN	AF6	ALT1	UART2_RXD__ ECSPI3_MISO	I	SPI1 Master Data input (input to CPU, output from SPI device)
P58	ESPI_IO_0/ SPI1_DO	AJ3	ALT1	UART1_TXD__ ECSPI3_MOSI	O	SPI1 Master Data output (output from CPU, input to SPI device)
P59	GND				P	Ground
P60	USB0+	D10		USB1_D_P	AIO	Differential USB0 data
P61	USB0-	E10		USB1_D_N	AIO	Differential USB0 data
P62	USB0_EN_OC#	A5/ A6	ALT0	GPIO1_IO11__ GPIO1_IO11 (EN)/ GPIO1_IO12__ GPIO1_IO12 (OC)	IO OD	Pulled low by Module OD driver to disable USB0 power. Pulled low by Carrier OD driver to indicate over-current situation If this signal is used, a pull-up is required on the Carrier

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P63	USB0_VBUS_DET	A11		USB1_VBUS	I	USB host power detection, when this port is used as a device
P64	USB0_OTG_ID	B11		USB1_ID	I	USB OTG ID input, active high
P65	USB1+				IO	Differential USB1 data pair (from USB2514 port 3)
P66	USB1-				IO	Differential USB1 data pair (from USB2514 port 3)
P67	USB1_EN_OC#	From USB2514			IO OD	Pulled low by Module OD driver to disable USB0 power Pulled low by Carrier OD driver to indicate over-current situation If this signal is used, a pull-up is required on the Carrier
P68	GND				P	Ground
P69	USB2+				IO	Differential USB2 data pair (from USB2514 port2)
P70	USB2-				IO	Differential USB2 data pair (from USB2514 port2)

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P71	USB2_EN_OC#	From USB2514			IO OD	Pulled low by Module OD driver to disable USB0 power Pulled low by Carrier OD driver to indicate over-current situation If this signal is used, a pull-up is required on the Carrier
P72	RSVD					Not used
P73	RSVD					Not used
P74	USB3_EN_OC#	From USB2514			IO OD	Pulled low by Module OD driver to disable USB0 power Pulled low by Carrier OD driver to indicate over-current situation If this signal is used, a pull-up is required on the Carrier

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P75	PCIE_A_RST#	U26	ALT5	SD1_DATA4__ GPIO2_IO06	O	Reset Signal for external devices.
P76	USB4_EN_OC#	From USB2514				Pulled low by Module OD driver to disable USB0 power Pulled low by Carrier OD driver to indicate over-current situation If this signal is used, a pull-up is required on the Carrier
P77	PCIE_B_CKREQ#					Not used
P78	PCIE_A_CKREQ#	AF8	ALT2	I2C4_SCL__ PCIE_CLKREQ_B		PCIe A Clock Request
P79	GND				P	Ground
P80	PCIE_C_REFCK+					Not used
P81	PCIE_C_REFCK-					Not used
P82	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P83	PCIE_A_REFCK+	from PI6CFGL201BZDIE			O	Differential PCI Express Reference Clock Signals for Lanes A
P84	PCIE_A_REFCK-	from PI6CFGL201BZDIE			O	Differential PCI Express Reference Clock Signals for Lanes A
P85	GND				P	
P86	PCIE_A_RX+	A14		PCIE_RXN_P	I	Differential PCIe Link A receive data pair 0
P87	PCIE_A_RX-	B14		PCIE_RXN_N	I	Differential PCIe Link A receive data pair 0
P88	GND				P	Ground
P89	PCIE_A_TX+	A15		PCIE_TXN_P	O	Differential PCIe Link A transmit data pair 0
P90	PCIE_A_TX-	B15		PCIE_TXN_N	O	Differential PCIe Link A transmit data pair 0
P91	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P92	HDMI_D2+ / DP1_LANE0+	AH27	N/A	HDMI_TX2_P	O	TMDS / HDMI data differential pair 2 / DP Data Pair 0+
P93	HDMI_D2- / DP1_LANE0-	AJ27	N/A	HDMI_TX2_N	O	TMDS / HDMI data differential pair 2/ DP Data Pair 0-
P94	GND				P	Ground
P95	HDMI_D1+/ DP1_LANE1+	AH26	N/A	HDMI_TX1_P	O	TMDS / HDMI data differential pair 1
P96	HDMI_D1-/ DP1_LANE1-	AJ26	N/A	HDMI_TX1_N	O	TMDS / HDMI data differential pair 1
P97	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P98	HDMI_D0+/ DP1_LANE2+	AH25	N/A	HDMI_TX0_P	O	TMDS / HDMI data differential pair 0
P99	HDMI_D0-/ DP1_LANE2-	AJ25	N/A	HDMI_TX0_N	O	TMDS / HDMI data differential pair 0
P100	GND				P	Ground
P101	HDMI_CK+/ DP1_LANE3+	AH24	N/A	HDMI_TXC_P	O	HDMI differential clock output pair
P102	HDMI_CK-/ DP1_LANE3-	AJ24	N/A	HDMI_TXC_N	O	HDMI differential clock output pair
P103	GND				P	Ground
P104	HDMI_HPD/ DP1_HDP	AE22	N/A	HDMI_HPD	I	HDMI Hot Plug Detect input
P105	HDMI_CTRL_CK/ DP1_AUX+	AC22	N/A	HDMI_DDC_SCL	IO OD	I2C Clock
P106	HDMI_CTRL_DAT/ DP1_AUX-	AF22	N/A	HDMI_DDC_SDA	IO OD	I2C Data
P107	DP1_AUX_SEL					Not used

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P108	GPIO0 / CAM0_PWR#	R25	ALT5	NAND_DATA00__ GPIO3_IO06	IO	Camera 0 Power Enable, active low output
P109	GPIO1 / CAM1_PWR#	L25	ALT5	NAND_DATA01__ GPIO3_IO07	IO	Camera 1 Power Enable, active low output
P110	GPIO2 / CAM0_RST#	L24	ALT5	NAND_DATA02__ GPIO3_IO08	IO	Camera 0 Reset, active low output
P111	GPIO3 / CAM1_RST#	N24	ALT5	NAND_DATA03__ GPIO3_IO09	IO	Camera 1 Reset, active low output
P112	GPIO4 / HDA_RST#	B5	ALT0	GPIO1_IO15 __ GPIO1_IO15	IO	HD Audio Reset, active low output
P113	GPIO5 / PWM_OUT	AD8	ALT5 ALT1	I2C4_SDA__ GPIO5_IO21/ PWM1_OUT	IO	PWM output
P114	GPIO6 / TACHIN	AH19	ALT5	SAI3_TXC__ GPIO5_IO00	IO	Tachometer input (used with the GPIO5 PWM)
P115	GPIO7	AH18	ALT5	SAI3_TXD__ GPIO5_IO01	IO	
P116	GPIO8	AJ18	ALT5	SAI3_RXC__ GPIO4_IO29	IO	
P117	GPIO9	AE18	ALT5	SPDIF_TX__ GPIO5_IO03	IO	
P118	GPIO10	AD18	ALT5	SPDIF_RX__ GPIO5_IO04	IO	
P119	GPIO11	AC18	ALT5	SPDIF_EXT_CLK__ GPIO5_IO05	IO	
P120	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P121	I2C_PM_CK	AC8	ALT0	I2C1_SCL__ I2C1_SCL	IO OD	Power management I2C bus clock
P122	I2C_PM_DAT	AH7	ALT0	I2C1_SDA__ I2C1_SDA	IO OD	Power management I2C bus data
P123	BOOT_SEL0#	B4	ALT0	GPIO1_IO05__ GPIO1_IO05	I	SYSBOOT and Line De-multiplexer Logic Pulled up on Module. Driven by OD part on Carrier.
P124	BOOT_SEL1#	A3	ALT0	GPIO1_IO06__ GPIO1_IO06	I	SYSBOOT and Line De-multiplexer Logic Pulled up on Module. Driven by OD part on Carrier.
P125	BOOT_SEL2#	F6	ALT0	GPIO1_IO07__ GPIO1_IO07	I	SYSBOOT and Line De-multiplexer Logic Pulled up on Module. Driven by OD part on Carrier.
P126	RESET_OUT#				O	General purpose reset output to Carrier board.

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P127	RESET_IN#				I	Reset input from Carrier board. Carrier drives low to force a Module reset, floats the line otherwise Pulled up on Module. Driven by OD part on Carrier.
P128	POWER_BTN#				I	Power-button input from carrier board. Carrier to float the line in in-active state. Active low, level sensitive. It is de-bounced on the Module Pulled up on Module. Driven by OD part on Carrier.
P129	SER0_TX	W28	ALT4	SD1_CLK__ UART1_DCE_TX	O	Asynchronous serial port data out
P130	SER0_RX	W29	ALT4	SD1_CMD__ UART4_DCE_RX	I	Asynchronous serial port data in
P131	SER0_RTS#	Y28	ALT4	SD1_DATA1__ UART1_DCE_CTS	O	Request to Send handshake line for SER0
P132	SER0_CTS#	Y29	ALT4	SD1_DATA0__ UART1_DCE_RTS	I	Clear to Send handshake line for SER0
P133	GND				P	Ground
P134	SER1_TX	AH5	ALT0	UART4_TXD__ UART4_DCE_TX	O	Asynchronous serial port data out
P135	SER1_RX	AJ5	ALT0	UART4_RXD__ UART4_DCE_RX	I	Asynchronous serial port data in

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P136	SER2_TX	AA28	ALT4	SD1_DATA6__ UART3_DCE_TX		Asynchronous serial port data out
P137	SER2_RX	U25	ALT4	SD1_DATA7__ UART3_DCE_RX		Asynchronous serial port data in
P138	SER2_RTS#	W26	ALT4	SD1_STROBE__ UART3_DCE_CTS		Request to Send handshake line for SER2
P139	SER2_CTS#	W25	ALT4	SD1_RESET_B__ UART3_DCE_RTS		Clear to Send handshake line for SER2
P140	SER3_TX	V29	ALT4	SD1_DATA2__ UART2_DCE_TX	O	Asynchronous serial port data out
P141	SER3_RX	V28	ALT4	SD1_DATA3__ UART2_DCE_RX	I	Asynchronous serial port data in
P142	GND				P	Ground
P143	CAN0_TX	AD16	ALT6	SAI5_RXD1__ CAN1_TX	O	CAN0 Transmit output
P144	CAN0_RX	AF16	ALT6	SAI5_RXD2__ CAN1_RX	I	CAN0 Receive input
P145	CAN1_TX	AE14	ATL6	SAI5_RXD3__ CAN2_TX	O	CAN1 Transmit output
P146	CAN1_RX	AF14	ALT6	SAI5_MCLK__ CAN2_RX	I	CAN1 Receive input

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P147	VDD_IN				P	Power in
P148	VDD_IN				P	Power in
P149	VDD_IN				P	Power in
P150	VDD_IN				P	Power in
P151	VDD_IN				P	Power in
P152	VDD_IN				P	Power in
P153	VDD_IN				P	Power in
P154	VDD_IN				P	Power in
P155	VDD_IN				P	Power in
P156	VDD_IN				P	Power in

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S1	CSI1_TX+/ I2C_CAM1_CK	AC14	ALT3	SAI5_RXFS__ I2C6_SCL	IO OD	Camera1 I2C bus clock
S2	CSI1_TX-/ I2C_CAM1_DAT	AD14	ALT3	SAI5_RXC__ I2C6_SDA	IO OD	Camera1 I2C bus data
S3	GND				P	Ground
S4	RSVD					Not used
S5	CSI0_TX+ / I2C_CAM0_CK	AH20	ALT2	ECSPI2_MISO__ I2C4_SCL	IO OD	Camera0 I2C bus clock
S6	CAM_MCK	A4	ALT6	GPIO1_IO14__ CCM_CLKO1	O	Master clock output for CSI camera support
S7	CSI0_TX- / I2C_CAM0_DAT	AJ22	ALT2	ECSPI2_SS0__ I2C4_SDA	IO OD	Camera0 I2C bus data
S8	CSI0_CK+	A23	ALT0	MIPI_CSI2_CLK_P	I	CSI0 differential clock inputs
S9	CSI0_CK-	B23	ALT0	MIPI_CSI2_CLK_N	I	CSI0 differential clock inputs
S10	GND				P	Ground
S11	CSI0_RX0+	A25	ALT0	MIPI_CSI2_D0_P	I	CS0 differential data inputs 0+
S12	CSI0_RX0-	B25	ALT0	MIPI_CSI2_D0_N	I	CSI0 differential data input 0-
S13	GND				P	Ground
S14	CSI0_RX1+	A24	ALT0	MIPI_CSI2_D1_P	I	CSI0 differential data input 1+
S15	CSI0_RX1-	B24	ALT0	MIPI_CSI2_D1_N	I	CSI0 differential data inputs 1-
S16	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S17	GbE1_MDI0+				AIO	Qualcomm AR8035 Differential Transmit/Receive Positive Channel 0
S18	GbE1_MDI0-				AIO	Qualcomm AR8035: Differential Transmit/Receive Negative Channel 0
S19	GbE1_LINK100#				O OD	Link Speed Indication LED for 100Mbps Could be able to sink 24mA or more Carrier LED current
S20	GbE1_MDI1+				AIO	Qualcomm AR8035 Differential Transmit/Receive Positive Channel 1
S21	GbE1_MDI1-				AIO	Qualcomm AR8035 Differential Transmit/Receive Negative Channel 1
S22	GbE1_LINK1000#				O OD	Link Speed Indication LED for 1000Mbps Could be able to sink 24mA or more Carrier LED current

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S23	GbE1_MDI2+				AIO	Qualcomm AR8035 Differential Transmit/Receive Positive Channel 2
S24	GbE1_MDI2-				AIO	Qualcomm AR8035 Differential Transmit/Receive Negative Channel 2
S25	GND				P	Ground
S26	GbE1_MDI3+				AIO	Qualcomm AR8035 Differential Transmit/Receive Positive Channel 3
S27	GbE1_MDI3-				AIO	Qualcomm AR8035 Differential Transmit/Receive Negative Channel 3
S28	GbE1_CTREF				O	Qualcomm AR8035 Center tap reference voltage for GBE Carrier board Ethernet magnetic
S29	PCIE_D_TX+					Not used
S30	PCIE_D_TX-					Not used
S31	GBE1_LINK_ACK#				O OD	Link / Activity Indication LED Driven low on Link (10, 100 or 1000 mbps) Blinks on Activity Could be able to sink 24mA or more Carrier LED current
S32	PCIE_D_RX+					Not used
S33	PCIE_D_RX-					Not used
S34	GND					Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S35	USB4+					Differential USB4 data pair (from USB2514 port 4)
S36	USB4-					Differential USB4 data pair (from USB2514 port 4)
S37	USB3_VBUS_DET					Not used
S38	AUDIO_MCK	AJ15	ALT0	SAI2_MCLK__ AUDIOMIX_SAI2 _MCLK	O	Master clock output to Audio codecs
S39	I2S0_LRCK	AJ17	ALT0	SAI2_TXFS__ AUDIOMIX_SAI2 _TX_SYNC	IO	Left& Right audio synchronization clock
S40	I2S0_SDOUT	AH16	ALT0	SAI2_TXD0__ AUDIOMIX_SAI2 _TX_DATA00	O	Digital audio Output
S41	I2S0_SDIN	AJ14	ALT0	SAI2_RXD0__ AUDIOMIX_SAI2 _RX_DATA00	I	Digital audio Input
S42	I2S0_CK	AH15	ALT0	SAI2_TXC__ AUDIOMIX_SAI2 _TX_BCLK	IO	Digital audio clock
S43	ESPI_ALERT0#					Not used
S44	ESPI_ALERT1#					Not used
S45	RSVD					Not used
S46	RSVD					Not used
S47	GND				G	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S48	I2C_GP_CK	AJ7	ALT0	I2C3_SCL__ I2C3_SCL	IO OD	General purpose I2C bus clock
S49	I2C_GP_DAT	AJ6	ALT0	I2C3_SDA__ I2C3_SDA	IO OD	General purpose I2C bus clock
S50	HDA_SYNC/ I2S2_LRCK	AC16	ALT0	SAI3_TXFS__ AUDIOMIX_SAI3_TX_ SYNC	IO	Left& Right audio synchronization clock
S51	HDA_SDO/ I2S2_SDOUT	L26	ALT2	NAND_CE0_B__AUDI OMIX_SAI3_TX_DATA 00	O	Digital audio Output
S52	HDA_SDI/ I2S2_SDIN	AF18	ALT0	SAI3_RXD__ AUDIOMIX_SAI3_RX_ DATA0	I	Digital audio Input
S53	HDA_CK/ I2S2_CK	N25	ALT2	NAND_ALE__ AUDIOMIX_SAI3_TX_ BCLK	IO	Digital audio clock
S54	SATA_ACT#					Not used
S55	USB5_EN_OC#					Not used
S56	ESPI_IO_2					Not used
S57	ESPI_IO_3					Not used
S58	ESPI_RESET#					Not used
S59	USB5+					Not used
S60	USB5-					Not used
S61	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S62	USB3_SSTX+	A13		USB2_TX_P	AO	USB3 data transmit signal differential pairs positive
S63	USB3_SSTX-	B13		USB2_TX_N	AO	USB3 data transmit signal differential pairs negative
S64	GND				P	Ground
S65	USB3_SSRX+	A12		USB2_RX_P	AI	USB3 data receive signal differential pairs positive
S66	USB3_SSRX-	B12		USB2_RX_N	AI	USB3 receive signal differential pairs negative
S67	GND				P	Ground
S68	USB3+					Differential USB3 data pair (from USB2514 port 1)
S69	USB3-					Differential USB3 data pair (from USB2514 port 1)
S70	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S71	USB2_SSTX+	A10		USB1_TX_P	AO	USB2 data transmit signal differential pairs positive
S72	USB2_SSTX--	B10		USB1_TX_N	AO	USB2 data transmit signal differential pairs negative
S73	GND				P	Ground
S74	USB2_SSRX+	A9		USB1_RX_P	AI	USB2 data receive signal differential pairs positive
S75	USB2_SSRX-	B9		USB1_RX_N	AI	USB2 receive signal differential pairs negative
S76	PCIE_B_RST#					Not used
S77	PCIE_C_RST#					Not used
S78	PCIE_C_RX+					Not used
S79	PCIE_C_RX-					Not used
S80	GND				P	Ground
S81	PCIE_C_TX+					Not used
S82	PCIE_C_TX-					Not used
S83	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S84	PCIE_B_REFCK+					Not used
S85	PCIE_B_REFCK-					Not used
S86	GND				P	Ground
S87	PCIE_B_RX+					Not used
S88	PCIE_B_RX-					Not used
S89	GND				P	Ground
S90	PCIE_B_TX+					Not used
S91	PCIE_B_TX-					Not used
S92	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S93	DP0_LANE0+				AIO	eDP0 data pair 0+
S94	DP0_LANE0-				AIO	eDP0 data pair 0-
S95	DP0_AUX_SEL					Not used
S96	DP0_LANE1+				AIO	eDP0 data pair 1+
S97	DP0_LANE1-				AIO	eDP0 data pair 1-
S98	DP0_HPD				I	eDP 0 Hot Plug Detect pins
S99	DP0_LANE2+				AIO	eDP0 data pair 2+
S100	DP0_LAN2-				AIO	eDP0 data pair 2-
S101	GND				P	Ground
S102	DP0_LANE3+				AIO	eDP0 data pair 3+
S103	DP0_LANE3-				AIO	eDP0 data pair 3-

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S104	USB3_OTG_ ID					Not used
S105	DP0_AUX+					eDP0 auxiliary channel pair +
S106	DP0_AUX-					eDP0 auxiliary channel pair -
S107	LCD1_BKLT_EN	AH21	ALT5	ECSPI2_SCLK__ GPIO5_IO10	O	High enables lvds1 panel backlight
S108	LVDS1_CK+ / eDP1_AUX+ / DSI1_CLK+	A28/ A18		LVDS1_CLK_P/ MIPI_DSI1_CLK_P	O	LVDS1/eDP1/DSI1 LCD differential clock pairs
S109	LVDS1_CK- / eDP1_AUX- / DSI1_CLK-	B28/ B18		LVDS1_CLK_N/ MIPI_DSI1_CLK_N	O	LVDS1/eDP1/DSI1 LCD differential clock pairs
S110	GND				P	Ground
S111	LVDS1_0+ / eDP1_TX0+ / DSI1_D0+	A26/ A16		LVDS1_D0_P/ MIPI_DSI1_D0_P	AIO	LVDS1/eDP1/DSI1 LCD data channel differential pairs 1
S112	LVDS1_0- / eDP1_TX0- / DSI1_D0-	B26/ B16		LVDS1_D0_N/ MIPI_DSI1_D0_N	AIO	LVDS1/eDP1/DSI1 LCD data channel differential pairs 1
S113	eDP1_HPD					Not used

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S114	LVDS1_1+ / eDP1_TX1+ / DSI1_D1+	A27/ A17		LVDS1_D1_P/ MIPI_DSI1_D1_P	AIO	LVDS1/eDP1/DSI1 LCD data channel differential pairs 2
S115	LVDS1_1- / eDP1_TX1- / DSI1_D1-	B27/ B17		LVDS1_D1_N/ MIPI_DSI1_D1_N	AIO	LVDS1/eDP1/DSI1 LCD data channel differential pairs 2
S116	LCD1_VDD_EN	D8	ALT0	GPIO1_GPIO11__ GPIO1_IO11	O	High enables lvds1 panel VDD
S117	LVDS1_2+ / eDP1_TX2+ / DSI1_D2+	B29/ A19		LVDS1_D2_P/ MIPI_DSI1_D2_P	AIO	LVDS1/eDP1/DSI1 LCD data channel differential pairs 3
S118	LVDS1_2- / eDP1_TX2- / DSI1_D2-	C28/ B19		LVDS1_D2_N/ MIPI_DSI1_D2_N	AIO	LVDS1/eDP1/DSI1 LCD data channel differential pairs 3
S119	GND				P	Ground
S120	LVDS1_3+ / eDP1_TX3+ / DSI1_D3+	C29/ A20		LVDS1_D3_P/ MIPI_DSI1_D3_P	AIO	LVDS1/eDP1/DSI1 LCD data channel differential pairs 4
S121	LVDS1_3- / eDP1_TX3- / DSI1_D3-	D28/ B20		LVDS1_D3_N/ MIPI_DSI1_D3_N	AIO	LVDS1/eDP1/DSI1 LCD data channel differential pairs 4
S122	LCD1_BKLT_ PWM	AJ20	ALT1	SAI3_MCLK__ PWM4_OUT	O	LCD1 display backlight PWM control
S123	GPIO13	AJ9	ALT5	SAI1_RXFS__ GPIO4_IO00		GPIO
S124	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S125	LVDS0_0+ / eDP0_TX0+ / DSI0_D0+	D25		LVDS0_D0_P	AIO	LVDS0 LCD data channel differential pairs 1
S126	LVDS0_0- / eDP0_TX0- / DSI0_D0-	E28		LVDS0_D0_N	AIO	LVDS0 LCD data channel differential pairs 1
S127	LCD_BKLT_EN	A7	ALT0	GPIO1_IO00__ GPIO1_IO00	O	High enables lvds0 panel backlight
S128	LVDS0_1+ / eDP0_TX1+ / DSI0_D1+	E29		LVDS0_D1_P	AIO	LVDS0 LCD data channel differential pairs 2
S129	LVDS0_1- / eDP0_TX1- / DSI0_D1-	F28		LVDS0_D1_N	AIO	LVDS0 LCD data channel differential pairs 2
S130	GND				P	Ground
S131	LVDS0_2+ / eDP0_TX2+ / DSI0_D2+	G29		LVDS0_D2_P	AIO	LVDS0 LCD data channel differential pairs 3
S132	LVDS0_2- / eDP0_TX2- / DSI0_D2-	H28		LVDS0_D2_N	AIO	LVDS0 LCD data channel differential pairs 3
S133	LCD_VDD_EN	E8	ALT0	GPIO1_IO01__ GPIO1_IO01	O	High enables lvds0 panel VDD

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S134	LVDS0_CK+ / eDP0_AUX+ / DSI0_CLK+	F29		LVDS0_CLK_P	O	LVDS0 LCD differential clock pairs
S135	LVDS0_CK- / eDP0_AUX- / DSI0_CLK-	G28		LVDS0_CLK_N	O	LVDS0 LCD differential clock pairs
S136	GND				P	Ground
S137	LVDS0_3+ / eDP0_TX3+ / DSI0_D3+	H29		LVDS0_D2_P	AIO	LVDS0 LCD data channel differential pairs 4
S138	LVDS0_3- / eDP0_TX3- / DSI0_D3-	J28		LVDS0_D2_N	AIO	LVDS0 LCD data channel differential pairs 4
S139	I2C_LCD_CK	AH6	ALT0	I2C2_SCL__ I2C2_SCL	IO OD	LCD display I2C bus clock
S140	I2C_LCD_DAT	AE8	ATL0	LVDS1_I2C0_SDA__ LVDS1_I2C0_SDA	IO OD	LCD display I2C bus clock
S141	LCD_BKLT_PWM	AE16	ALT2	SAI5_RXD0__ PWM2_OUT	O	LCD0 display backlight PWM control
S142	GPIO12	A8	ALT0	GPIO1_IO08__ GPIO1_IO08		GPIO
S143	GND				P	Ground
S144	eDP0_HPD					Not used
S145	WDT_TIME_OUT#	AC10	ALT5	SAI1_RXD0__ GPIO4_IO02	O	Watchdog-Timer Output
S146	PCIE_WAKE#	AA29	ALT5	SD1_DATA5__ GPIO2_IO07	I	PCI Express Wake Event: Sideband wake signal asserted by components requesting wakeup.

SMARC Edge Finger		NXP i.MX8M Plus CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S147	VDD_RTC				P	Low current RTC circuit backup power - 3.0V nominal It is sourced from a Carrier based Lithium cell or Super Cap
S148	LID#	From Port0 of TCA6408 IO Expander			I	Lid open/close indication to Module. Low indicates lid closure (which system may use to initiate a sleep state). Carrier to float the line in in-active state. Active low, level sensitive. Should be de-bounced on the Module Pulled up on Module. Driven by OD part on Carrier.
S149	SLEEP#	From Port1 of TCA6408 IO Expander			I	Sleep indicator from Carrier board. May be sourced from user Sleep button or Carrier logic. Carrier to float the line in in-active state. Active low, level sensitive. Should be de-bounced on the Module. Pulled up on Module. Driven by OD part on Carrier.

SMARC Edge Finger		NXP i.MX8M PlusMQ CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S150	VIN_PWR_BAD#				I	Power bad indication from Carrier board. Module and Carrier power supplies (other than Module and Carrier power supervisory circuits) shall not be enabled while this signal is held low by the Carrier. Pulled up on Module. Driven by OD part on Carrier.
S151	CHARGING#	From Port2 of TCA6408 IO Expander			I	Held low by Carrier if DC input for battery charger is present. Pulled up on Module. Driven by OD part on Carrier.
S152	CHARGER_PRSNT#	From Port4 of TCA6408 IO Expander			I	Held low by Carrier if DC input for battery charger is present.
S153	CARRIER_STBY#				O	The Module shall drive this signal low when the system is in a standby power state

SMARC Edge Finger		NXP i.MX8M PlusMQ CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S154	CARRIER_PWR_ON				O	Carrier board circuits (apart from power management and power path circuits) should not be powered up until the Module asserts the CARRIER_PWR_ON signal.
S155	FORCE_RECOV#				I	Pulled up on Module. Driven by OD part on Carrier.
S156	BATLOW#	From Port3 of TCA6408 IO Expander			I	Battery low indication to Module. Carrier to float the line in in-active state. Pulled up on Module. Driven by OD part on Carrier.
S157	TEST#				I	Held low by Carrier to invoke Module SD Boot UP. Pulled up on Module. Driven by OD part on Carrier.
S158	GND				P	Ground

Power Control Signals between SMARC Module and Carrier

This Chapter points out the handshaking rule between *SMARC* module and carrier.

Section include：

- *SMARC-iMX8MP* Module Power
- Power Signals
- Power Flow and Control Signals Block Diagram
- Power States
- Power Sequences
- Terminations
- Boot Select

Chapter 4 Power Control Signals between SMARC-iMX8MP Module and Carrier

SMARC modules are designed to be driven with a single +3V to +5.25V input power rail. Unlike Q7 module, there is no separate voltage rail for standby power, other than the very low current RTC voltage rail. All module operating and standby power comes from the single set of *VDD_IN* pins. This suits battery power sources well, and is also easy to use with non-battery sources.

SMARC module has specific handshaking rules to the carrier by *SMARC* hardware specification. To design the carrier board, users need to follow these rules or it might not boot up. Some pull-up and pull-down also need to be cared to make all functions work.

4.1 SMARC-iMX8MP Module Power

4.1.1. Input Voltage / Main Power Rail

The allowable Module DC input voltage range for *SMARC-iMX8MP* is from 3.0V to 5.25V. This voltage is brought in on the *VDD_IN* pins and returned through the numerous *GND* pins on the connector.

Ten pins are allocated to *VDD_IN*. The connector pin current rating is 0.5A per pin. This works out to 5A total for the 10 pins. At the lowest allowed Module input voltage, this would allow up to 16.75W of electrical power to be brought in (with no de-rating on the connector current capability). With a 40% connector current de-rating, up to 10W may be brought in at 3V.
SMARC-iMX8MP typically consumes 3~4W depending on dual or quad cores and is pretty safe in using the connector.

4.1.2. No Separate Standby Voltage

There is no separate voltage rail for standby power, other than the very low current RTC voltage rail. *SMARC-iMX8MP* operating and standby power comes from the single set of *VDD_IN* pins. This suits battery power sources well, and is also easy to use with non-battery sources.

4.1.3. RTC/Backup Voltage

RTC backup power is brought in on the *VDD_RTC* rail. The RTC consumption is typically 15 microA or less. The allowable *VDD_RTC* voltage range shall be 2.0V to 3.25V. The *VDD_RTC* rail is sourced from a Carrier based Lithium cell, or it may be left open if the RTC backup functions are not required. *SMARC-iMX8MP* module is able to boot without a *VDD_RTC* voltage source.

Lithium cells, if used on Carrier, shall be protected against charging by a Carrier Schottky diode. The diode is placed in series with the positive battery terminal. The diode anode is on the battery side, and the cathode on the Module *VDD_RTC* side.

Note that if a Super cap is used, current may flow out of the Module *VDD_RTC* rail to charge the Super Cap.

4.1.4. Power Sequencing

The Module signal *CARRIER_PWR_ON* exists to ensure that the Module is powered before the main body of Carrier circuits (those outside the power and power control path on the Carrier). The main body of Carrier board circuits should not be powered until the Module asserts the *CARRIER_PWR_ON* signal as a high. Module hardware will assert *CARRIER_PWR_ON* when all Module supplies necessary for Module booting are up.

The IO power of carrier board will be turn on at the stage of power on sequence. If the IO power of carrier board been turn on earlier than the *SMARC* module, the power on carrier board might feedback to *SMARC*

module through IO lines and disturbs the *SMARC* module power on sequence. More seriously, it might cause to the CPU won't boot up. It is always recommended that the power on module has to be earlier than that on carrier board.

The boot up of module depends on when you release the reset signal of your carrier board. The module will boot up when the reset signal on your carrier board is released. Before that, the module will not boot up. That's why designer needs to put the *RESET_IN#* in the last stage of power to serve as the "*power good*" signal of the carrier board.

The module will not boot up till the module power is ready because the carrier board hasn't released the reset signal yet.

The sequence is as follows:

Module Power Ready --> *CARRIER_POWER_ON* -->*RESET_IN#* -->Boot Up

4.1.5. RESET_IN#

The *SMARC* module does not know the IO power status from the carrier board, and put *RESET_IN#* in the last stage of power can serve as the "*power good*" signal of carrier board. This also assures that the power of carrier board is good when *SMARC* module booting up.

4.1.6. VDD_IO

The 3.3V *VDD_IO* is depreciated from *SMARC* 1.1 specification.

SMARC-iMX8MP supports 1.8V *VDD_IO* only.

4.1.7. Power Bad Indication (VIN_PWR_BAD#)

Power bad indication is from carrier board and is an input signal for Module. Module and Carrier power supplies (other than Module and Carrier power

supervisory circuits) will not be enabled while this signal is held low by the Carrier.
This signal has a 100K pull-up on module and is driven by *OD* part on Carrier.

4.1.8. System Power Domains

It is useful to describe an *SMARC* system as being divided into a hierarchy of three power domains:
1) Battery Charger power domain (can be neglected if the system is not battery powered only)
2) *SMARC* Module power domain
3) Carrier Circuits power domain
The Battery Charger domain includes circuits that are active whenever either charger input power and / or battery power are available. These circuits may include power supply supervisor(s), battery chargers, fuel gauges and, depending on the battery configuration, switching power section(s) to step down a high incoming battery voltage.
The *SMARC* Module domain includes the *SMARC* module.
The Carrier Circuits domain includes “everything else” (and does not include items from the Battery Charger and Module domain, even though they may be mounted on the Carrier).
This is illustrated in the figure below.

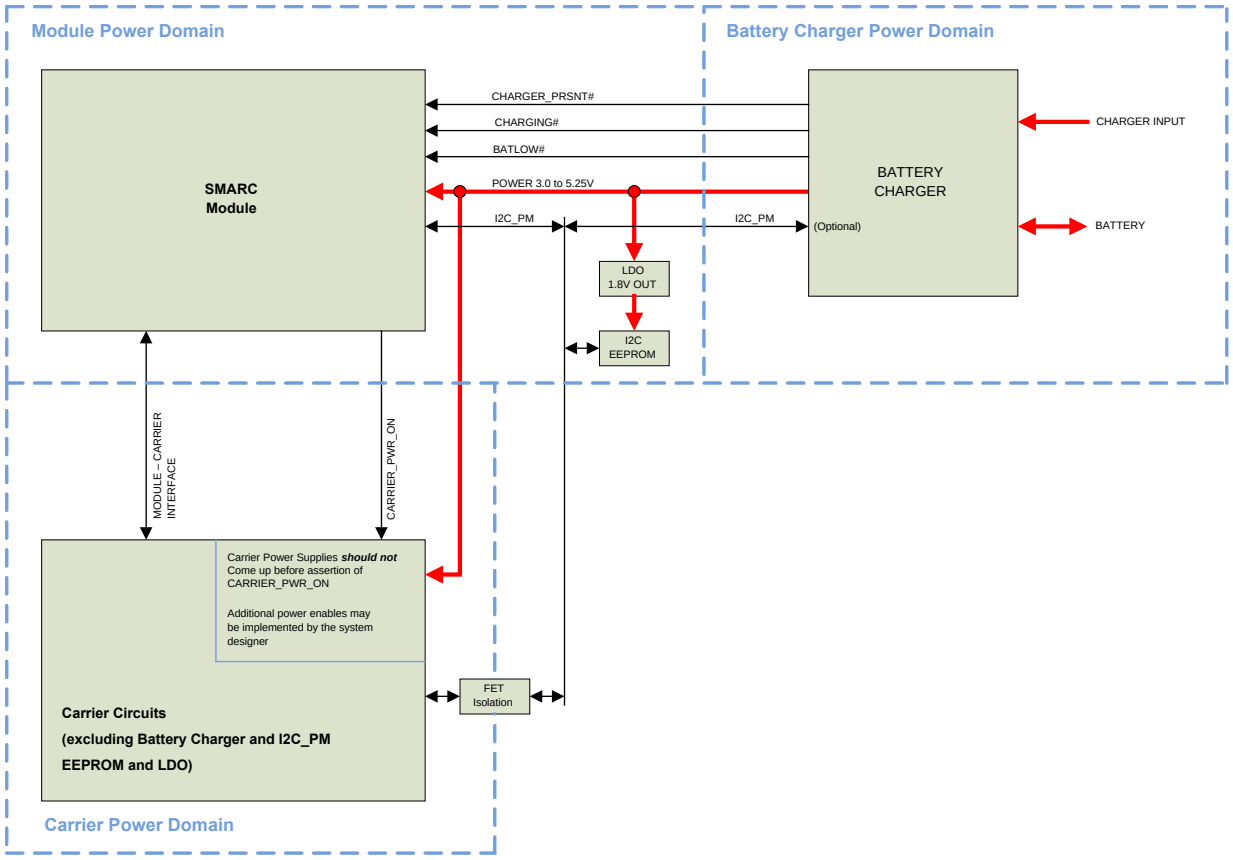


Figure 26 System Power Domains

4.2 Power Signals

4.2.1. Power Supply Signals

SMARC Edge Finger		I/O	Type	Power Rail	Description
Pin#	Pin Name				
P147, P148, P149, P150, P151,P152, P153, P154, P155, P156	VDD_IN	I	PWR	3.0V~5.25V[1]	Main power supply input for the module
P2, S3, P9, S10, P12, S13, P15, S16, P18, S25, P32, S34, P38, S47, P47, P50, P53, P59, S61, S64, S67, P68, S70, S73, P79, S80, P82, S83, P85, S86, P88, S89, P91, S92, P94, P97, P100, S101, P103, S110, S119, P120, S124, S130, P133, S136, P142, S143, S158	GND	I	PWR		Common signal and power ground
S147	VDD_ RTC	I	PWR	3.3V	RTC supply, can be left unconnected if internal RTC is not used

4.2.2. Power Control Signals

The input pins listed in the following table are all active low and are meant to be driven by *OD* (open drain) devices on the Carrier. The Carrier either floats the line or drives it to *GND*. No Carrier pull-ups are needed. The pull-up functions are performed on the Module. The voltage rail that these lines are pulled to on the Module varies, depending on the design, and may be 3.3V or *VDD_IN*.

SMARC Edge Finger		I/O	Type	Power Rail	Description
Pin#	Pin Name				
S150	VIN_PWR_BAD#	I	CMOS	VDD_IN	Power bad indication from Carrier board
S154	CARRIER_PWR_ON	O	CMOS	VDD_IO	Signal to inform Carrier board circuits being powered up
P126	RESET_OUT#	O	CMOS	VDD_IO	General purpose reset output to Carrier board.
P127	RESET_IN#	I	CMOS	VDD_IO	Reset input from Carrier board. Carrier drives low to force a Module reset, floats the line otherwise. Pulled up on Module. Driven by OD part on Carrier.
P128	POWER_BTN#	I	CMOS	VDD_IO	Power-button input from Carrier board. Carrier to float the line in in-active state. Active low, level sensitive. It is de-bounced on the Module Pulled up on Module. Driven by OD part on Carrier.

4.2.3. Power Management Signals

The pins listed in the following table are related to power management. They will be used in a battery-operated system.

SMARC Edge Finger		I/O	Type	Power Rail	Description
Pin#	Pin Name				
S156	BATLOW#	I	CMOS	VDD_IO	Battery low indication to Module. Carrier to float the line in in-active state. Pulled up on Module. Driven by OD part on Carrier.
S154	CARRIER_PWR_ON	O	CMOS	VDD_IO	Signal to inform Carrier board circuits being powered up
S153	CARRIER_STBY#	O	CMOS	VDD_IO	Module will drive this signal low when the system is in a standby power state
S152	CHARGER_PRSNT#	I	CMOS	VDD_IO	Held low by Carrier if DC input for battery charger is present. Pulled up on Module. Driven by OD part on Carrier.

SMARC Edge Finger		I/O	Type	Power Rail	Description
Pin#	Pin Name				
S151	CHARGING#	I	Strap	VDD_IO	Held low by Carrier during battery charging. Carrier to float the line when charge is complete. Pulled up on Module. Driven by OD part on Carrier.
S149	SLEEP#	I	CMOS	VDD_IO	Sleep indicator from Carrier board. May be sourced from user Sleep button or Carrier logic. Carrier to float the line in in-active state. Active low, level sensitive. Should be de-bounced on the Module. Pulled up on Module. Driven by OD part on Carrier.
S148	LID#	I	CMOS	VDD_IO	Lid open/close indication to Module. Low indicates lid closure (which system may use to initiate a sleep state). Carrier to float the line in in-active state. Active low, level sensitive. Should be de-bounced on the Module Pulled up on Module. Driven by OD part on Carrier.

4.2.4. Special Control Signals (TEST#)

SMARC-iMX8MP does not support to boot up from *SPI NOR* flash. *SMARC-iMX8MP* module boots up from the onboard *eMMC* Flash first. The firmware in the *eMMC* flash will read the *BOOT_SEL* configuration and decides where to load the u-boot.

In some situations like the firmware in *eMMC* flash needed to be upgrade/restore or at factory default where the firmware in *eMMC* flash is empty or at development stage that the firmware in *eMMC* needs to be modified, users will need an alternative way to boot up from SD card first. The *TEST#* pin serves as this purpose. The *TEST#* pin is pulled high on module. If carrier board leaves this pin floating or pulls high, the module will boot up from on-module *eMMC*. If carrier board pulls this pin to *GND*, the module will boot up from *SD* card first. The first stage bootloader in *i.MX8M Plus* CPU ROM codes will load the 2nd stage bootloader based on the setting of this *#TEST* pin (*S157*).

4.3 Power Flow and Control Signals Block Diagram

Following figures shows the power flow and control signals block diagram.

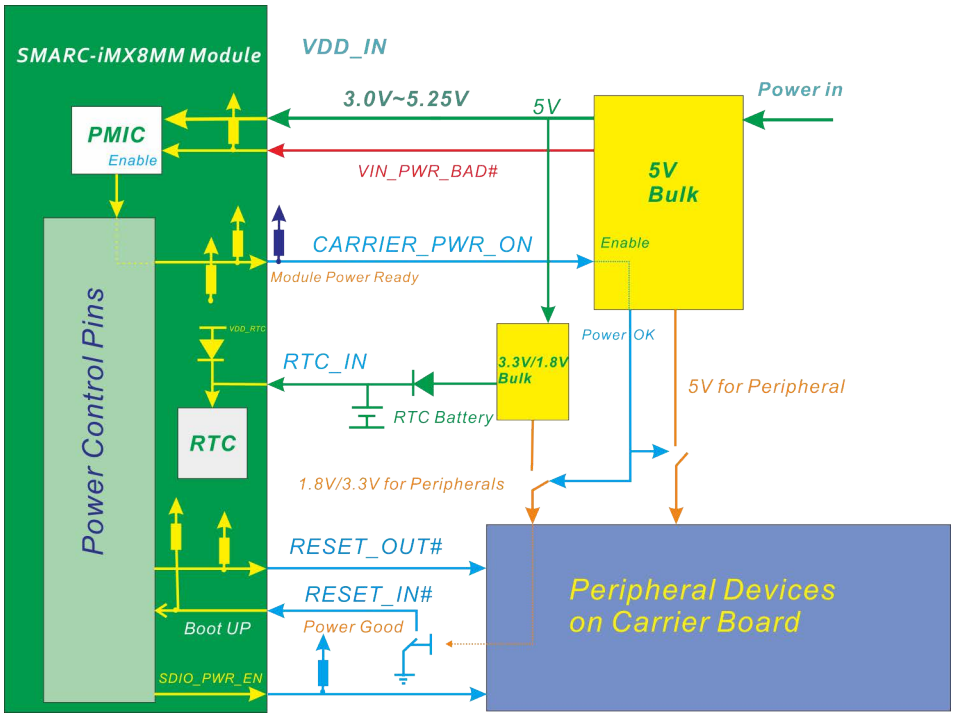


Figure 27 Power Block Diagram

When main power is supplied from the carrier, a voltage detector will assert *VIN_PWR_BAD#* signal to tell the module and carrier that the power is good. This signal will turn on the *PMIC* on module to power on the module.

Carrier power circuits in the carrier Power domain should not power up unless the module asserts *CARRIER_PWR_ON*. The module signal *CARRIER_PWR_ON* exists to ensure that the module is powered before the main body of carrier circuits (those outside the power and power control path on the carrier).

The main body of carrier board circuits will not be powered until the module asserts the *CARRIER_PWR_ON* signal being correct. Module hardware will assert *CARRIER_PWR_ON* when all power supplies necessary for module booting are ready. The module will continue to assert signal *RESET_OUT#* after the release of *CARRIER_PWR_ON*, for a period sufficient to allow carrier power circuits to come up. When Carrier power is ready, it will assert *RESET_IN#* to inform module booting up.

If users would like to have SD boot up, *SDIO_PWR_EN* signal have to be pull up to 3.3V on carrier.

Module and carrier power supplies will not be enabled if the *VIN_PWR_BAD#* is held low by carrier. It is a power bad indication signal from carrier and is 200k pull up to *VDD_IN* on module.

4.4 Power States

The *SMARC-iMX8MP* module supports different power states. The table below describes the behavior in the different states and which power rails and peripherals are active. Additional power states can be implemented if required using available GPIOs to control additional power domains and peripherals.

Abbr.	Name	Description	Module	Carrier Board
UPG	*Unplugged*	*No power is applied to the system, except the RTC battery might be available*	*No main VDD_IN applied from fixed DC supply, VDD_IN available if backup battery is implemented*	*No power supply input, RTC battery maybe inserted*
OFF	*off*	*System is off, but the carrier board input supply is available*	*The main VDD_IN is available, but the CPU and peripherals are not running. Only the PMIC is running*	*Carrier board provides power for module, the peripheral supplies are not available*
SUS	*Suspend*	*System is suspended and waits for wakeup sources to trigger*	*CPU is suspended, wakeup capable peripherals are running while others might be switched off*	*Power rails are available on carrier board, peripherals might be stopped by software*
RUN	*Running*	*System is running*	*All power rails are available, CPU and peripherals are running*	*All power rails are available, peripherals are running*
RST	*Reset*	*System is put in reset state by holding RESET_IN# is low*	*All power rails are available, CPU and peripherals are in reset state*	*All power rails are available, peripherals are in reset state*

The figure below shows a sequence diagram for the different power states. The module automatically enters into the running mode when the main power rail is applied to the module. In the running mode, the system can be set to

suspend by software. There might be different wake up sources available. Consult the datasheet for *SMARC-iMX8MP* module for more information about the available wakeup events.

In the running state, a shutdown request can be triggered by software. This turns off all power rails on the module and requests the carrier board to switch of the power rails for the peripherals. The module can be brought back to the running mode in two ways. The module main voltage rail (*VDD_IN*) can be removed and applied again. If needed, this could also be done with a button and a small circuit. *SMARC-iMX8MP* module supports being power cycled by asserting the *RESET_IN#* signal (e.g. by pressing the reset button or shunt and relief the reset jumper), please consult the associated module datasheet for more information about the support power cycle methods.

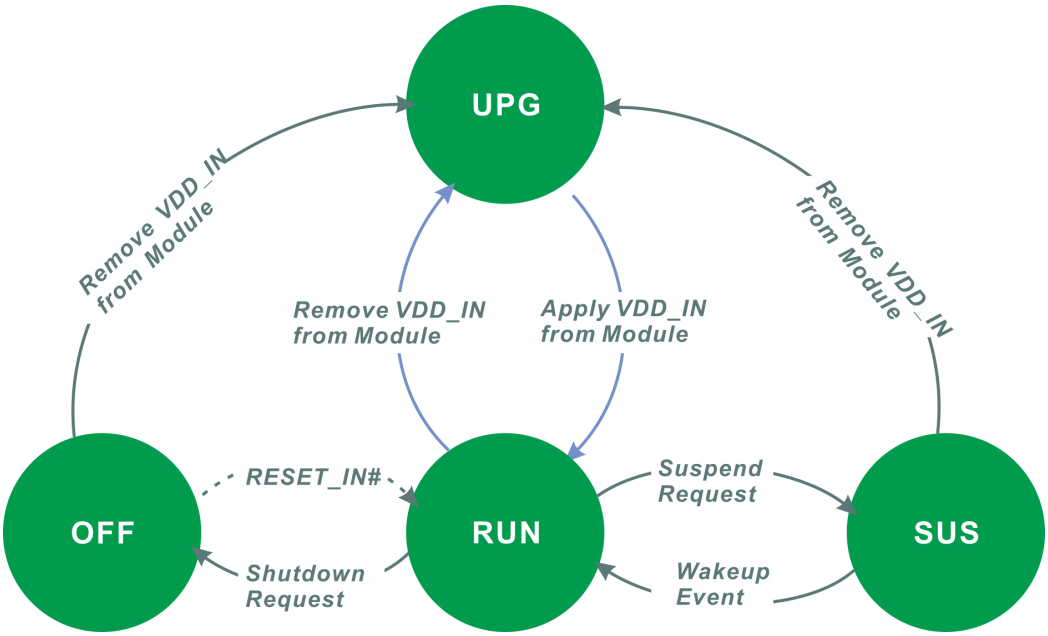


Figure 28 Power States and Transitions

4.5 Power Sequences

When main power is supplied from the carrier, a voltage detector will assert *VIN_PWR_BAD#* signal to tell the module and carrier that the power is good. This signal will enable the *PMIC* on module to power on the module. The module will not power up if the module receives a low-active *VIN_PWR_BAD#* signal.

The *SMARC-iMX8MP* module starts asserting *CARRIER_PWR_ON* as soon as the main voltage supply being applied to the module and all power supplies necessary for module booting are up. This is to ensure that the module is powered before the main body of carrier circuits (those outside the power and power control path on the carrier). The module will continue to assert signal *RESET_OUT#* after the release of *CARRIER_PWR_ON*, for a period sufficient time (at least 10ms) to allow carrier power circuits that the peripheral supplies need to ramp up.

The peripheral power rails on the carrier board need to ramp up in a correct sequence. The sequence starts normally with the highest voltage (e.g. 5V) followed by the lower voltages (e.g. 3.3V then 1.8V and so on). Peripherals normally require that a lower voltage rails is never present if a higher rail is missing. Check the datasheet of all peripheral components on the carrier board for a proper sequencing. The *SMARC-iMX8MP* modules guarantees to apply the reset output *RESET_OUT#* not earlier than 100ms after the *CARRIER_PWR_ON* goes high. This gives the carrier board a sufficient time for ramping up all power rails. *SDIO_PWR_EN* signal have to be pull up to 3.3V on carrier if users would like to have SD boot up functionality.

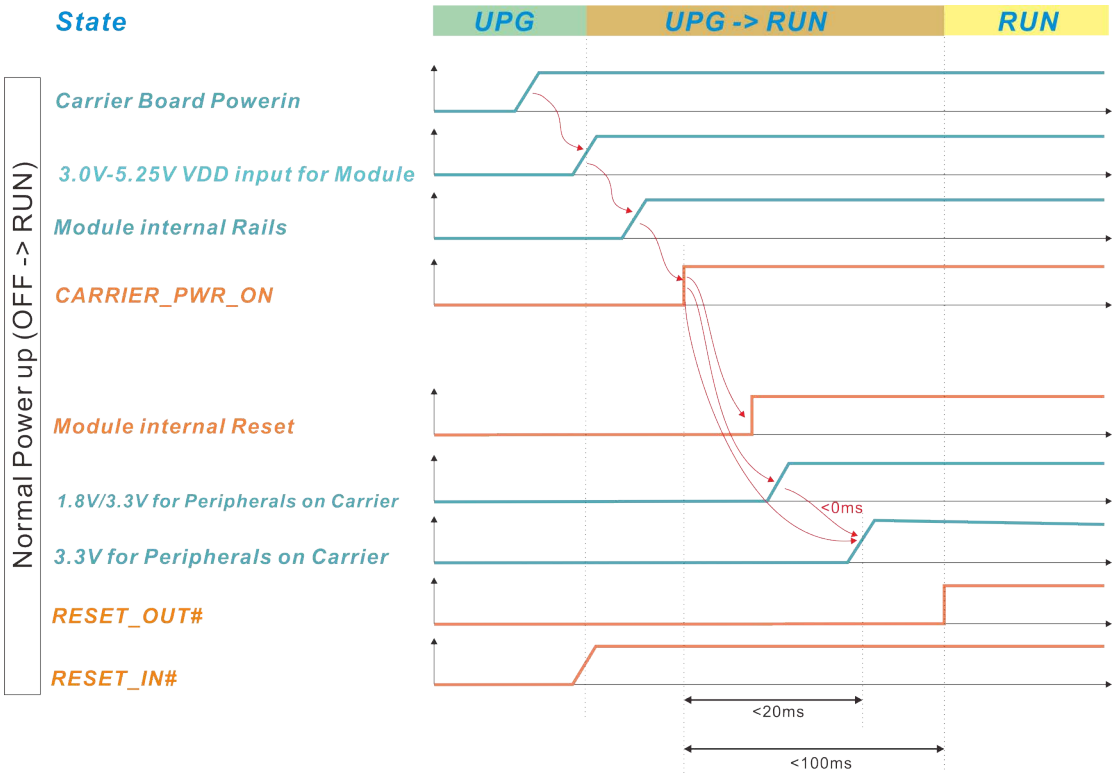


Figure 29 Power-Up Sequences

If the operating system supports it, a shutdown sequence can be initiated. Some systems may benefit from shutting down instead of just removing the main power supply as this allows the operating system to take care of any housekeeping (e.g. bringing mass storage devices to a controlled halt). Some operating system may not provide the shutdown function.

As it is not permitted that a lower voltage rail is present when a higher voltage rail has been switched off, the sequence of shutting down the peripheral voltages needs to be considered. The lower voltages (e.g. peripheral 3.3V) need to ramp down before the higher ones do (e.g. peripheral 5V).

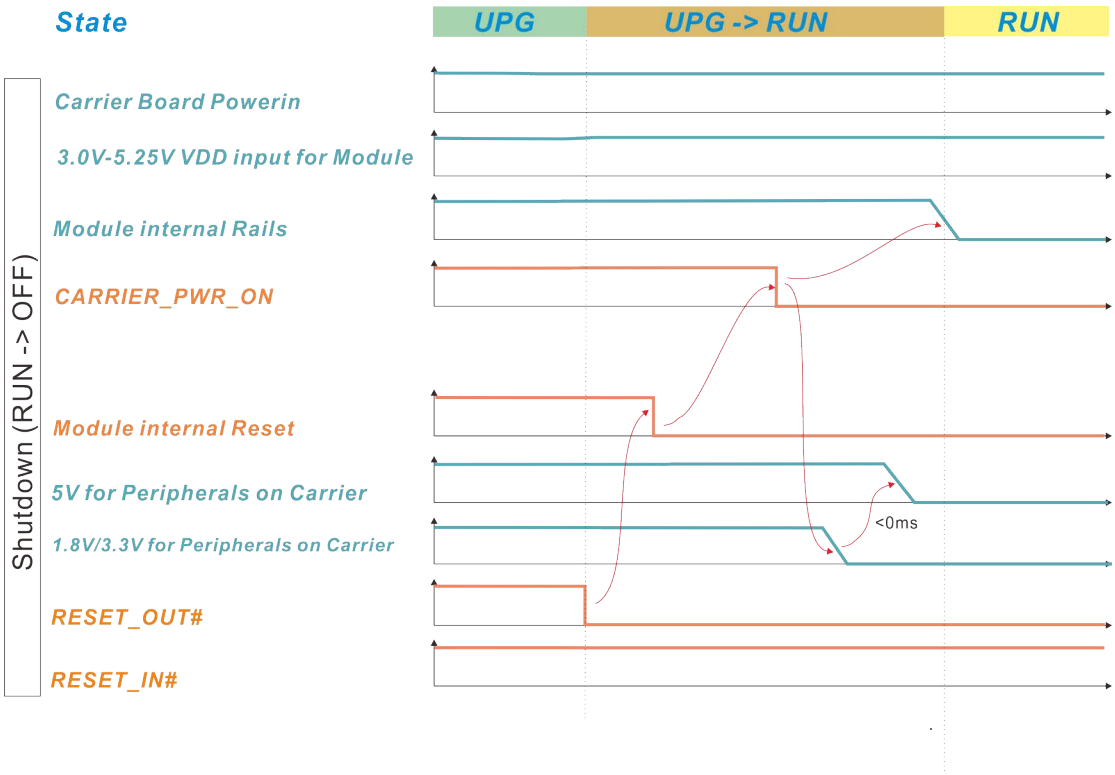


Figure 30 Shutdown Sequence

When the *RESET_IN#* is asserted, a reset cycle is initiated. The module internal reset and the external reset output *RESET_OUT#* are asserted as long as *RESET_IN#* is asserted. If the reset input *RESET_IN#* is de-asserted, the internal reset and the *RESET_OUT#* will remain low for at least 1ms until they are also de-asserted and the module starts booting again. This guarantees a minimum reset time of 1ms even if the reset input *RESET_IN#* is triggered for a short time.

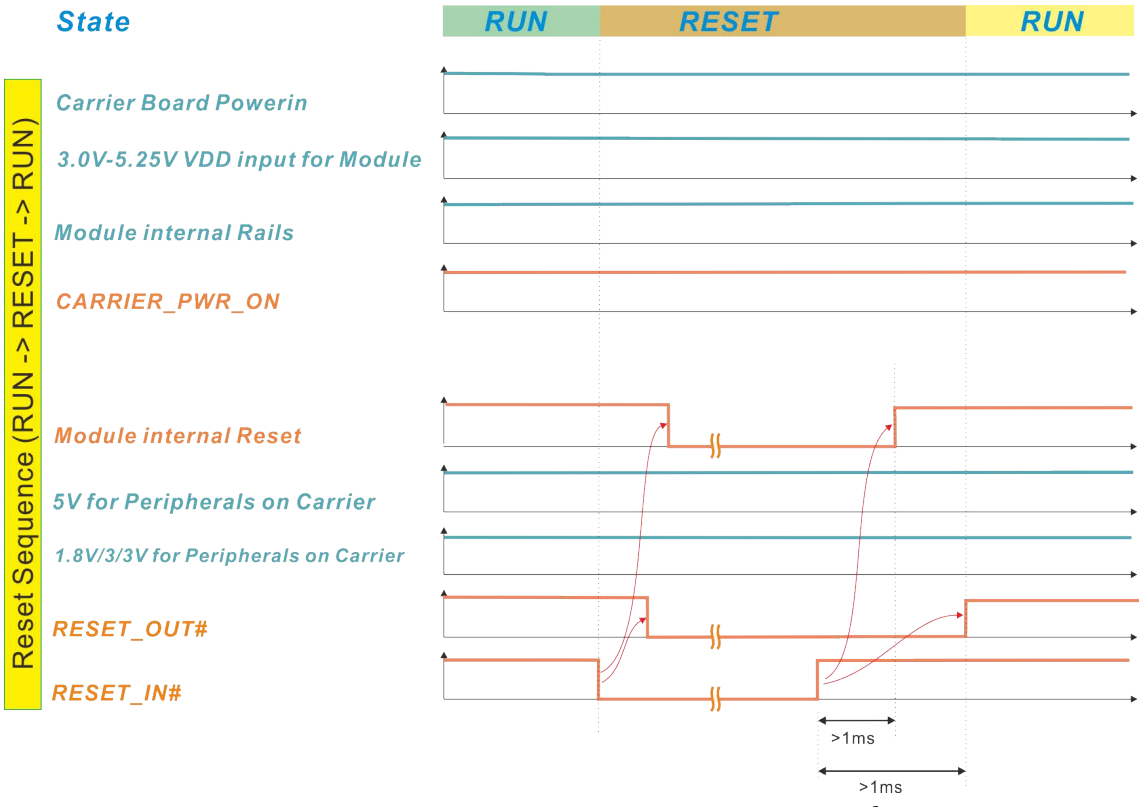


Figure 31 Reset Sequence

4.6 Terminations

4.6.1. Module Terminations

The Module signals listed below will be terminated on the Module. The terminations follow the guidance given in the table below.

Signal Name	Series Termination	Parallel Termination	Notes
HDMI_CTRL_DAT		1.5k pull-up to 1.8V	Carrier pull-up required
HDMI_CTRL_CK		1.5k pull-up to 1.8V	Carrier pull-up required
PCIE_A_TX+	0.2uF Capacitor		
PCIE_A_TX-	0.2uF Capacitor		
I2C_PM_DAT		2.2K pull-up to 1.8V	
I2C_PM_CK		2.2K pull-up to 1.8V	
I2C_LCD_DAT		2.2K pull-up to 1.8V	
I2C_LCD_CK		2.2K pull-up to 1.8V	
I2C_CAM[0:1]_DAT		2.2K pull-up to 1.8V	
I2C_CAM[0:1]_CK		2.2K pull-up to 1.8V	
I2C_GP_DAT		2.2K pull-up to 1.8V	
I2C_GP_CK		2.2K pull-up to 1.8V	
SDIO_CD#		10k pull-up to 3.3V	
SDIO_WP		10k pull-up to 3.3V	

Signal Name	Series Termination	Parallel Termination	Notes
USB[0:4]_EN_OC#		10K pull-up to 3.3V or a switched 3.3V on the Module	x is '0' or '1' Switched 3.3V: if a USB channel is not used, then the USBx_EN_OC# pull-up rail may be held at GND to prevent leakage currents.
VIN_PWR_BAD#		200k pull-up to VIN	
USB[2:3]_SSTX+	0.2uF Capacitor		
USB[2:3]_SSTX-	0.2uF Capacitor		

4.6.2. Carrier/Off-Module Terminations

The following Carrier terminations are required, if the relevant interface is used. If unused, the *SMARC* Module pins may be left un-connected.

Module Signal Group Name	Carrier Series Termination	Carrier Parallel Termination	Notes
GBE_MDI	*Magnetics module appropriate for 10/100/1000 GBE transceivers*	*Secondary side center tap terminations appropriate for Gigabit Ethernet implementations*	
GBE_LINK **(GBE status LED sinks)**		*If used, current limiting resistors and diodes to pulled to a positive supply rail*	*The open drain GBE status signals, GBE_LINK100#, GBE_LINK1000# and GBE_LINK_ACT#, if used, need Carrier based current limiting resistors and LEDs. The LED may be integrated into a Carrier RJ45 jack. A resistor of 68 ohms, and a LED with the anode tied to Carrier 3.3V, is typical.*
LVDS LCD		*100 ohm resistive termination across the differential pairs at the endpoint of the signal path, usually on the display assembly*	

Module Signal Group Name	Carrier Series Termination	Carrier Parallel Termination	Notes
HDMI_CTRL_DAT **HDMI_CTRL_CK**		*Pull-ups to VDD_IO on each of these lines is required on the Carrier.* *The pull-ups may be part of an integrated HDMI ESD protection and control-line level shift device, such as the Texas Instruments TPD12S016.* *If discrete Carrier pull-ups are used, they should be 10K.*	
PCIe_A_RX+ **PCIe_A_RX-**	*Series coupling caps near the TX pins of the Carrier board PCIe device (0.2uF)*		
PCIe_B_RX+ **PCIe_B_RX-**	*Series coupling caps near the TX pins of the Carrier board PCIe device (0.2uF)*		
USB[2:3]_SSRX+ **USB[2:3]_SSRX-**	*Series coupling caps near the TX pins of the Carrier board USB 3.0 device (0.2uF)*		

4.7 Boot Device Selection

SMARC hardware specification defines three pins (*BOOT_SEL[0:2]*) that allow the Carrier board user to select from eight possible boot devices. *SMARC-iMX8MP* does not support boot up from SPI flash. If *TEST#* is not shunt cross to GND, the first stage of bootloader on *SMARC-iMX8MP* will boot up from on-module *eMMC* first. The firmware on *eMMC* will read the boot device configuration and load the second stage bootloader from selected boot devices. The *BOOT_SELx#* pins are weakly pulled up on the Module and the pin states decoded by module logic. The Carrier shall either leave the Module pin Not Connected ("Float" in the table below) or shall pull the pin to GND, per the table below.

	Carrier Connection			Boot Source
	BOOT_SEL2#	BOOT_SEL1#	BOOT_SEL0#	
0	GND	GND	GND	Carrier SATA
1	GND	GND	Float	Carrier SD Card
2	GND	Float	GND	Carrier eSPI (CS0#)
3	GND	Float	Float	Carrier SPI
4	Float	GND	GND	Module Device (USB)
5	Float	GND	Float	Remote Boot (GBE)
6	Float	Float	GND	Module eMMC Flash
7	Float	Float	Float	Module SPI

If *TEST#* pin is shunt cross to GND, the first stage of bootloader on *SMARC-iMX8MP* will boot up from off-module *SD* card. This is a back door to restore/upgrade the firmware in on-module *eMMC*.