User's Manual

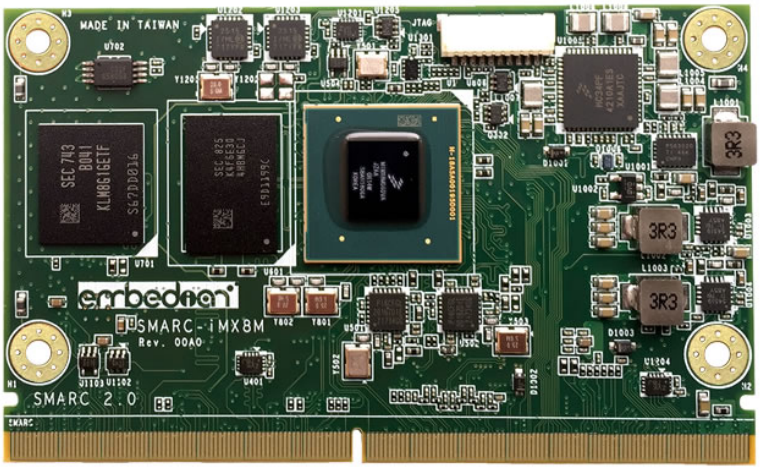
SMARC Computer on Module

NXP/Freescale *i.MX8M* Cortex A53 and Cortex M4
24bits dual-channel LVDS
HDMI 2.0a/DP
4 x COM Ports
1 x SDHC
4 x USB Host 2.0, 1 x USB 2.0 OTG, 2 x USB3.0
1 x 10/100/1000M Gigabit Ethernet
2 x CAN Bus, 2 x SPIs, 5 x I2Cs, 12-bit GPIOs
2 x PCIe x1 Gen. 2.1, 2 x MIPI_CSI

SMARC-iMX8M

Dual, Quad and Quad Lite Cores

(SMARC 2.0 Specification Compliant)


MADE IN TAIWAN
SMARC-iMX8M
Rev. 00A0
SMARC 2.0
3R3
3R3
3R3

Revision History

Revision	Date	Changes from Previous Revision
1.0	2018/09/03	Initial Release
1.2	2018/12/07	Make change for rev. 00B0.
1.4	2019/10/28	Make changes for rev. 00C0
		BOOT_SEL0: Change from GPIO1_IO04 to GPIO1_IO08
		BATLOW#: Change from GPIO1_IO08 to SAI2_RXFS
		CHARGER_PRSNT#: Change from GPIO1_IO12 to SAI2_RXC
		SLEEP#: Change from GPIO1_IO010 to GPIO1_IO12
		ADD USB0_OTG_ID pin from GPIO1_IO10
		ADD SD_VSELECT pin from GPIO1_IO04
		LVDS_EN: Change from SAI1_RXD0 to SAI3_RXFS
		LVDS_IRQ: Change from SAI1_RXD2 to SAI3_RXC
1.5	2019/11/15	Correct CPU Ball Mappings for some pins
2.0	2021/05/18	Make Changes for rev. 00E0
		SER0_CTS#: Change pinmux from ECSPI2_MISO__UART4_DCE_CTS to ECSPI2_SS0__UART4_DCE_RTS
		SER0_RTS#: Change pinmux from ECSPI2_SS0__UART4_DCE_RTS to ECSPI2_MISO__UART4_DCE_CTS
		SER0_TX: Change pinmux from UART4_TXD__UART4_DCE_TX to ECSPI2_MOSI__UART4_DCE_TX
		SER0_RX: Change pinmux from UART4_RTX__UART4_DCE_RX to ECSPI2_SCLK__UART4_DCE_RX
		SER2_CTS#: Change pinmux from ECSPI2_SCLK__GPIO5_IO10 to UART4_TXD__UART2_DCE_RTS
		SER2_RTS#: Change pinmux from ECSPI2_MOSI__GPIO5_IO11 to UART4_RXD__UART2_DCE_CTS
		Charging# Change pinmux from GPIO1_IO01__GPIO1_IO1 to NAND_DATA05__GPIO3_IO11
		ADD SW HDMI hot plug emulator "HDMI_HPD_HOG#" via GPIO1_DATA01

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
- NXP is a registered trademark of NXP
- 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-iMX8M* for NXP *i.MX8M* 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* processor reference manual and related documentation for additional information.

Introduction

This Chapter gives background information on the *SMARC-iMX8M*

Section include：

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

Chapter 1 Introduction

The *SMARC-iMX8M* offers high-performance processing for a low-power System-on-Module. It perfectly fits various embedded products, the growing market of connected and portable devices and segment for connected streaming audio/video devices, scanning/imaging devices and various devices requiring high-performance but low-power processors.

The product is based on the NXP *i.MX 8M* Dual/Quad Lite/Quad family of multi-purpose processors, featuring an ARM® Cortex™-A53 up to 1.5GHz with an additional 266MHz ARM Cortex-M4 core.

This heterogeneous multicore processing architecture enables the device to run an open operating system like Linux on the Cortex-A53 core and an RTOS like FreeRTOS™ on the Cortex-M4 core for time and security critical tasks.

The module connector has 314 edge fingers that mate with a low profile 314 pin 0.5mm pitch right angle connector (this connector is sometimes identified as an 321 pin connector, but 7 pins are lost to the key).

Featuring NXP's *i.MX8M* System-on-Chip, Embedian's *SMARC-iMX8M* offers single- or dual-channel 18-bit/24-bit LVDS LCD, Gigabit Ethernet, HDMI/DP, SDHC, USB 2.0, USB 3.0, UARTs, CAN bus, PCIe and many peripheral interfaces in a cost effective, low power, miniature package. Embedian's *SMARC-iMX8M* thin and robust design makes it an ideal building block for reliable system design with a wide range of products in target markets requiring high-performance processing with low power consumption, compact size and a cost-effective solution.

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

- Building Control - Fire and Security panel, Elevator Control, HVAC control
- Industrial Vehicle - Avionics cockpit display, in-flight infotainment, train and heavy equipment HMI
- Healthcare – patient monitor
- Personal UAVs
- Smart Cities
- Voice control and voice assistants
- General Control System
- And more

Complete and cost-efficient Embedian evaluation kits for Yocto build, Debian

9 , Ubuntu 16.04 and Android Oreo 8.1 allow immediate and professional embedded product development with dramatically reduced design risk and time-to-market.

1.1 Features and Functionality

The *SMARC-iMX8M* module is based on the *i.MX8M* processor with dual, quad lite, and quad core 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.
- Processor: *NXP i.MX8M* ARM Cortex-A53 and Cortex-M4 up to 1.5GHz
- Memory: Onboard 16GB eMMC Flash
- Onboard 2GB or 4GB LPDDR4
- Networking: 1 x 10/100/1000 Mbps Ethernet

  Display:
  - Single channel LVDS LCD 24-bit or dual channel LVDS
  - HDMI 2.0a/DP
- Expansion: 1 x SDHC/SDIO, 5x USB 2.0 (one OTG), 2 x USB 3.0, 2 x PCIe x1 Gen 2.1
- 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 (one eSPI)
  - 5 x I2C
  - 2 x I2S
  - 2 x CAN Bus
  - 2 x PWM
  - 2 x MIPI CSI (Camera Interface)
  - 12 x GPIOs
  - WDT
- SW Support: Linux, Yocto Build, Ubuntu 16.04, Debian 9, Android Oreo 8.1
- Power Consumption (Typcal)

- ~4W
- 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)

1.2 Module Variant

The *SMARC-iMX8M* module is available with various options based on processors in this family from NXP, LPDDR4 memory configuration, and operating temperature ranges.

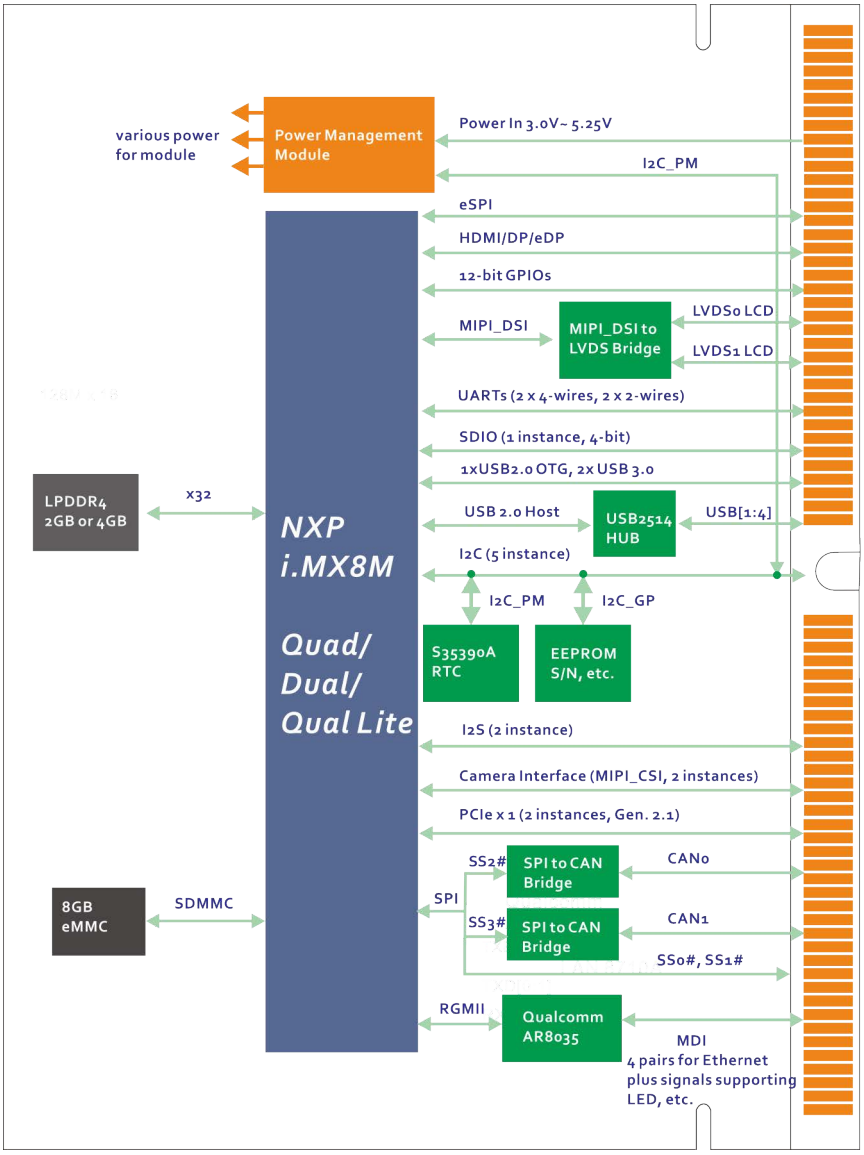
1. "*D*" (dual core, CPU running up to 2 x 1.5GHz)
   "*L*" (quad lite core, CPU running up to 4 x 1.5GHz)
   "*Q*" (quad core, CPU running up to 4 x 1.5GHz)
2. "*2G*" (2GB LPDDR4 memory)
   "*4G*" (4GB LPDDR4 memory)
3. "*I*" Industrial temperature (-40°C~85°C for 2GB *LPDDR4* and -30°C~85°C for 4GB *LPDDR4*), CPU running up to 1.3GHz, support 2GB LPDDR4 only)
   Leave it Blank if commercial temperature
4. "C" (Conformal coating) – Leave it blank if no needs of conformal coating.

For example, *SMARC-iMX8M-D-2G* stands for dual core *i.MX8M* processor running up to 1.5GHz with 2GB LPDDR4 memory in normal operating temperature.

1.3 Block Diagram

The following diagram illustrates the system organization of the *SMARC-iMX8M*. Arrows indicate direction of control and not necessarily signal flow.

Figure 1: SMARC-iMX8M Block Diagram



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

1.4 Software Support / Hardware Abstraction

The Embedian *SMARC-iMX8M* Module is supported by Embedian BSPs (Board Support Package). The first *SMARC-iMX8M* BSP targets Linux (Ubuntu 16.04 LTS, Debian 9, Yocto Build) and Android Oreo 8.1 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* 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*).
- ***Serial ATA Revision 3.1,*** July 18, 2011, Gold Revision, © Serial ATA International Organization *(www.sata-io.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).*
- ***HDMI Specification, Version 1.3a****, November 10, 2006 © Hitachi and other companies (www.hdmi.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 ("Controller Area Network")*** *Bus Standard – ISO 11898*
- ***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-iMX8M User's Manual***
- ***PinMux file for SMARC-iMX8M***
- ***SMARC-iMX8M Schematic Checklist***

1.5.4. NXP Documents

- ***IMX8MDQLQRM, i.MX8M Quad Applications Processor Reference Manual,*** Jan 8th, 2018 (rev. 0)
-
- ***IMX8MDQLQCEC, i.MX 8M Dual / 8M QuadLite / 8M Quad Applications Processors Data Sheet for Industrial Products,*** May 23, 2018 (rev. 0.1)
- ***IMX8MDQLQCEC, i.MX 8M Dual / 8M QuadLite / 8M Quad Applications Processors Data Sheet for Consumer Products,*** May 23, 2018 (rev. 0.1)
- ***IMX8MDQLQHDG, i.MX8M Hardware Developer's Guide,*** *Jan08. 2018 (rev. 0)*
- ***AN12118, i.MX 8M Quad Power Consumption Measurement***, May 23, 2018 (rev. 1)

1.5.5. NXP Development Tools

- ***IOMUX_TOOL v4.1 for ARM® i.MX8M Microprocessors***

1.5.6. NXP Software Documents

- ***Linux 4.9.88_2.0.0***
- ***Android O8.1.0_1.3.0 8MQ GA Documentation***

1.5.7. Embedian Software Documents

- ***Embedian Linux BSP for SMARC-iMX8M Module***
- ***Embedian Android BSP for SMARC-iMX8M 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-iMX8M* specifications.

Section include：

- *SMARC-iMX8M* General Functions
- *SMARC-iMX8M* Debug
- Mechanical Specifications
- Electrical Specification
- Environment Specification

Chapter 2 Specifications

2.1 SMARC-iMX8M General Functions

2.1.1. SMARC-iMX8M Feature Set

This section lists the complete feature set supported by the *SMARC-iMX8M* module.

SMARC Feature Specification	SMARC 2.0 Specification Maximum Number Possible	SMARC-iMX8M Feature Support	SMARC-iMX8M Feature Support Instances
LVDS LCD Display Support	1	Yes	1 (dual channel)[Note1]
DP/eDP	1	Yes	1[Note2]
HDMI Display Support	1	Yes	1
Serial Camera Support	2	Yes	2 (2-lanes and 4-lanes)
USB Interface	6	Yes	5 (1 x USB 2.0 OTG, 4 x USB 2.0 and 2 x USB 3.0)
PCIe Interface	4	Yes	2 (x1 Gen 2.1))
SATA Interface	1	No	N/A
GbE Interface	1	Yes	1
2nd GBE Interface	1	No	N/A
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-iMX8M Feature Support	SMARC-iMX8M Feature Support Instances
CAN	2	Yes	2
VDDIO	1.8V	1.8V	1.8V

Note:

1. Dual channel LVDS interface: 2 x 18 bpp OR 2 x 24 bpp (up to 1,920 × 1,200 @60 fps at 24 bpp). Default configuration is single channel 24-bit. To change this configuration, users need to send i2c command to *SN65DSI84 MIPI_DSI* to *LVDS* bridge. Please refer to Embedian u-boot official BSP release.
2. DP interface is not validated! Awaits NXP formal release.

2.1.2. Form Factor

The *SMARC-iMX8M* module complies with the *SMARC* General Specification module size requirements in an 82mm x 50mm form factor.

2.1.3. CPU

The *SMARC-iMX8M* implements *NXP's i.MX8M* ARM processor.

NXP CPU	i.MX8M Dual	i.MX8M Quad Lite	i.MX8M Quad
ARM Cores[Not 1]	2x 1.5GHz Cortex-A53	4x 1.5GHz Cortex-A53	4x 1.5GHz Cortex-A53
ARM Cores	1x Cortex-M4F	1x Cortex-M4F	1x Cortex-M4F
Memory Speed	LPDDR4-3200	LPDDR4-3200	LPDDR4-3200
L2 Cache	1MB L2	1MB L2	1MB L2
GPU	GC7000Lite 4 shaders OpenGL ES 3.1, OpenCL 1.2, OpenGL 3.0, OpenVG and Vulkan	GC7000Lite 4 shaders OpenGL ES 3.1, OpenCL 1.2, OpenGL 3.0, OpenVG and Vulkan	GC7000Lite 4 shaders OpenGL ES 3.1, OpenCL 1.2, OpenGL 3.0, OpenVG and Vulkan
VPU	4Kp60 HEVC/H.265, H.264, VP9 Decoder 1080p60 MPEG-2, MPEG-4p2, VC-1, VP8, RV9, AVS, MJPEG, H.263 Decoder	NO VPU	4Kp60 HEVC/H.265, H.264, VP9 Decoder 1080p60 MPEG-2, MPEG-4p2, VC-1, VP8, RV9, AVS, MJPEG, H.263 Decoder
HDMI/DP	Up to 4Kp60 video/graphics display	Up to 4Kp60 video/graphics display	Up to 4Kp60 video/graphics display

Note:

1. For industrial temp. boards, the clocking speed is only up to 1.3GHz.
2. The only difference for *Quad Lite* core is that this processor does not have VPU.

2.1.4. Onboard Storage

The *SMARC-iMX8M* module supports an 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 serial number. The Module serial EEPROM data structure conforms to the PICMG® EEEP Embedded *EEPROM* Specification.). The onboard 16GB eMMC flash is used as boot media. The module will always boot up from the on board eMMC flash first. Th efirmware 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 25 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 25 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* CPU RTC (Real Time Clock) and external (S-35390A) RTC.
A 27 MHz *HCSL* oscillator is used as the reference clock for *HDMI* PHY.
The Qualcomm AR8035 PHY, PCIe *HCSL* clock generator and Microchip CAN controllers are provided with a 25 MHz clock using a crystal in normal oscillation mode.

2.1.6 LVDS Interface

The *SMARC-iMX8M* implements two 18 / 24 bit single channel LVDS output streams that are defined in SMARC 2.0 edge connector for the Primary displays from *i.MX8M MIPI_DSI* interface. They can also be configured as an 18 / 24 bit dual-channel LVDS directly out of the *SMARC* Module.

The *LVDS LCD* signals found on the *SMARC-i.MX8M* offers two LVDS channels, with resolutions up to 1,920 × 1,200 @60 fps at 24 bpp. They are generated from *MIPI_DSI* signals from the *NXP® i.MX8M* Cortex A53 processor passing through a TI *SN65DSI84 MIPI® DSI* Bridge To FLATLINK™ LVDS. Each channel consists of one clock pair and four data pairs. The *LVDS* signals support the flow of *MIPI DSI* data from the *i.MX8M* CPU to external display devices through LVDS interface.

The LVDS ports support the following configurations:

- One single channel output
- One dual channel output: single input split to two output channels

Note:

1. The I2C slave address of *SN65DSI84* is 0x2C.
2. The *LVDS* interface can be used either as a single channel or as a dual channel. The default LVDS configuration is 24-bit single channel LVDS. To change this configuration, user need to change 0x18 register bit [2:4]. Please refer to *Embedian* u-boot BSP official release for details.

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

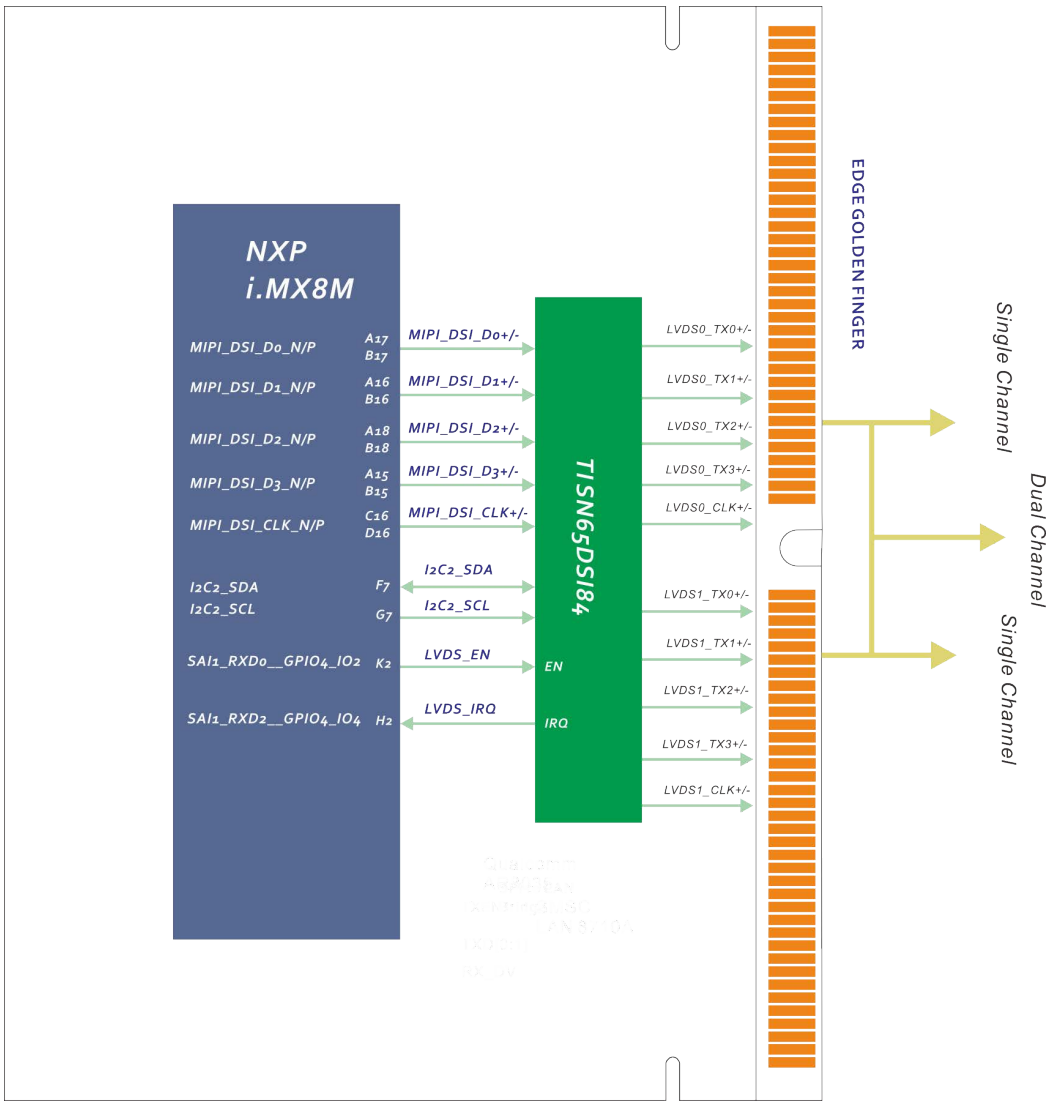


Figure 2: SMARC-iMX8M LVDS LCD Diagram

2.1.6.1 LVDS channel select

SMARC-iMX8 LVDS LCD interface can be configured as an 18-bit/24-bit single-channel *LVDS* output or a dual-channel *LVDS* output by accessing TI *SN65DSI84* 0x18 register via *I2C_GP* bus. The default configuration from software is 24-bit single-channel *LVDS*. User can refer to Embedian official u-boot release and *SN65DSI84* datasheet to figure out how to change to different configuration.

2.1.6.2 LVDS Signals Data Flow

i.MX8M processor and *TI SN65DSI84* implementation is shown in the following table:

NXP i.MX8M CPU			TI SN65DSI84		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
C16	ALT0	MIPI_DSI_CLK_N__ MIPI_DSI_CLK_N	J5	DACN	MIPI_DSI_CLK-	
D16	ALT0	MIPI_DSI_CLK_P__ MIPI_DSI_CLK_P	H5	DACP	MIPI_DSI_CLK+	
A17	ALT0	MIPI_DSI_D0_N__ MIPI_DSI_D0_N	J3	DA0N	MIPI_DSI_D0-	
B17	ALT0	MIPI_DSI_D0_P__ MIPI_DSI_D0_P	H3	DA0P	MIPI_DSI_D0+	
A16	ALT0	MIPI_DSI_D1_N__ MIPI_DSI_D1_N	J4	DA1N	MIPI_DSI_D1-	
B16	ALT0	MIPI_DSI_D1_P__ MIPI_DSI_D1_P	H4	DA1P	MIPI_DSI_D1+	
A18	ALT0	MIPI_DSI_D2_N__ MIPI_DSI_D2_N	J6	DA2N	MIPI_DSI_D2-	
B18	ALT0	MIPI_DSI_D2_P__ MIPI_DSI_D2_P	H6	DA2P	MIPI_DSI_D2+	
A15	ALT0	MIPI_DSI_D3_N__ MIPI_DSI_D3_N	J7	DA3N	MIPI_DSI_D3-	
B15	ALT0	MIPI_DSI_D3_P__ MIPI_DSI_D3_P	H7	DA3P	MIPI_DSI_D3+	
F4	ALT5	SAI3_RXC__ GPIO4_IO29	J9	IRQ	LVDS_IRQ	
G4	ALT5	SAI3_RXFS__ GPIO4_IO28	B1	EN	LVDS_EN	

The path from *TI SN65DSI84* to the golden finger edge connector is show in the following table.

TI SN65DSI84		Golden Finger Edge Connector		Net Names	Note
Pin	Pin Name	Pin#	Pin Name		
SN65DSI84, ChannelA (Default)					
C8	A_Y0P	S125	LVDS0_0+/ eDP0_TX0+/ DSI0_D0+	LVDS0_0+	LVDS0 LCD data channel differential pairs 1
C9	A_Y0N	S126	LVDS0_0-/ eDP0_TX0-/ DSI0_D0-	LVDS0_0-	
D8	A_Y1P	S128	LVDS0_1+/ eDP0_TX1+/ DSI0_D1+	LVDS0_1+	LVDS0 LCD data channel differential pairs 2
D9	A_Y1N	S129	LVDS0_1-/ eDP0_TX1-/ DSI0_D1-	LVDS0_1-	
E8	A_Y2P	S131	LVDS0_2+/ eDP0_TX2+/ DSI0_D2+	LVDS0_2+	LVDS0 LCD data channel differential pairs 3
E9	A_Y2N	S132	LVDS0_2-/ eDP0_TX2-/ DSI0_D2-	LVDS0_2-	
G8	A_Y2P	S134	LVDS0_CK+/ eDP0_AUX+/ DSI0_CLK+	LVDS0_CK+	LVDS0 LCD differential clock pairs
G9	A_Y2N	S135	LVDS0_CK-/ eDP0_AUX-/ DSI0_CLK-	LVDS0_CK-	
F8	A_CLKP	S137	LVDS0_3+/ eDP0_TX3+/ DSI0_D3+	LVDS0_3+	LVDS0 LCD data channel differential pairs 4
F9	A_CLKN	S138	LVDS0_3-/ eDP0_TX3-/ DSI0_D3-	LVDS0_3-	

TI SN65DSI84		Golden Finger Edge Connector		Net Names	Note
Pin	Pin Name	Pin#	Pin Name		
SN65DSI84 ChannelB					
B6	B_CLKP	S108	LVDS1_CK+/ eDP1_AUX+/ DSI1_CLK+	LVDS1_CK+	LVDS1 LCD differential clock pairs
A6	B_CLKN	S109	LVDS1_CK-/ eDP1_AUX-/ DSI1_CLK-	LVDS1_CK-	
B3	B_Y0P	S111	LVDS1_0+/ eDP1_TX0+/ DSI1_D0+	LVDS1_0+	LVDS1 LCD data channel differential pairs 1
A3	B_Y0N	S112	LVDS1_0-/ eDP1_TX0-/ DSI1_D0-	LVDS1_0-	
B4	B_Y1P	S114	LVDS1_1+/ eDP1_TX1+/ DSI1_D1+	LVDS1_1+	LVDS1 LCD data channel differential pairs 2
A4	B_Y1N	S115	LVDS1_1-/ eDP1_TX1-/ DSI1_D1-	LVDS1_1-	
B5	B_Y2P	S117	LVDS1_2+/ eDP1_TX2+/ DSI1_D2+	LVDS1_2+	LVDS1 LCD data channel differential pairs 3
A5	B_Y2N	S118	LVDS1_2-/ eDP1_TX2-/ DSI1_D2-	LVDS1_2-	
B7	B_Y3P	S120	LVDS1_3+/ eDP1_TX3+/ DSI1_D3+	LVDS1_3+	LVDS1 LCD data channel differential pairs 4
A7	B_Y3N	S121	LVDS1_3-/ eDP1_TX3-/ DSI1_D3-	LVDS1_3-	

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.

2.1.6.3 Other LCD Control Signals

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

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
LCD0_VDD_EN	*Output*	*CMOS 1.8V*	*High enables panel VDD*
LCD0_BKLT_EN	*Output*	*CMOS 1.8V*	*High enables panel backlight*
LCD0_BKLT_PWM	*Output*	*CMOS 1.8V*	*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 CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
L1	ALT5	SAI1_RXFS__ GPIO4_IO0	S127	LCD0_BKLT_EN	LCD_ BKLT_EN	High enables panel backlight
K1	ALT5	SAI1_RXC__ GOIO4_IO1	S133	LCD0_VDD_EN	LCD_ VDD_EN	High enables panel VDD
E6	ALT1	SPDIF_EXT_CLK__ PWM1_OUT	S141	LCD0_BKLT_PWM	LCD0_ BKLT_ PWM	Display backlight PWM control
G7	ALT0	I2C2_SCL__ I2C2_SCL	S139	I2C_LCD_CK	I2C_ LCD_CK	I2C data – to read LCD display EDID EEPROMs
F7	ALT0	I2C2_SDA__ I2C2_SDA	S140	I2C_LCK_DAT	I2C_ LCD_DA T	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 HDMI* Video quality can reach full 4K UltraHD resolution and HDR (Dolby Vision, HDR10, and HLG).

A 27 MHz *HCSL* oscillator is used as the reference clock for *HDMI PHY.* The *SMARC-iMX8M* provides *HDMI* connection directly from the *NXP® i.MX8M* 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-iMX8M* 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.

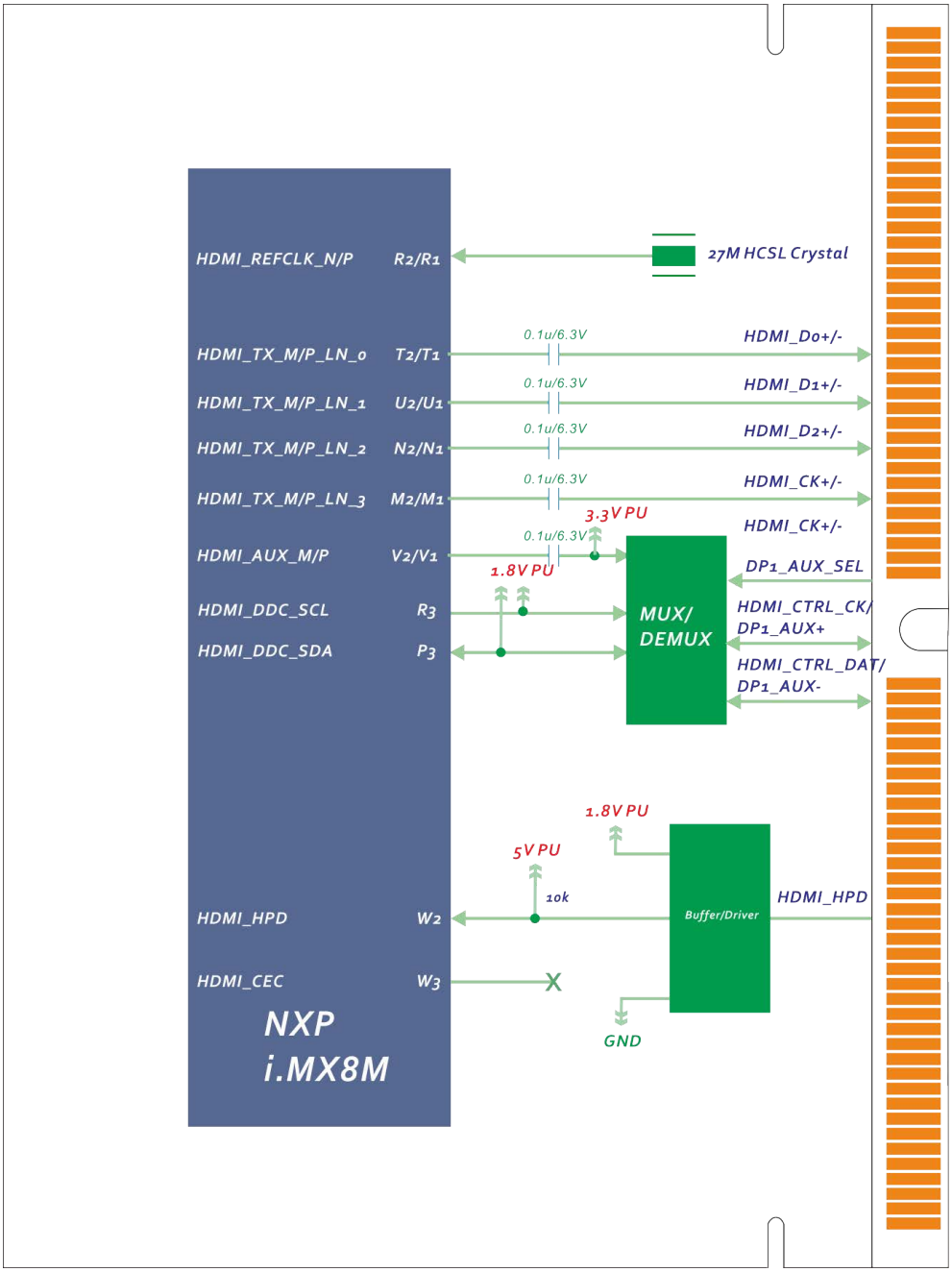


Figure 3: SMARC-iMX8M HDMI Diagram

NXP i.MX8M CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
HDMI						
T2	N/A	HDMI_TX_M_LN_0	P99	HDMI_D0-/ DP1_LANE2-	HDMI_D0-	TMDS / HDMI data differential pair 0
T1	N/A	HDMI_TX_P_LN_0	P98	HDMI_D0+/ DP1_LANE2+	HDMI_D0+	
U1	N/A	HDMI_TX_M_LN_1	P96	HDMI_D1-/ DP1_LANE1-	HDMI_D1-	TMDS / HDMI data differential pair 1
U2	N/A	HDMI_TX_P_LN_1	P95	HDMI_D1+/ DP1_LANE1+	HDMI_D1+	
N1	N/A	HDMI_TX_M_LN_2	P93	HDMI_D2-/ DP1_LANE0-	HDMI_D2-	TMDS / HDMI data differential pair 2
N2	N/A	HDMI_TX_P_LN_2	P92	HDMI_D2+/ DP1_LANE0+	HDMI_D2+	
M2	N/A	HDMI_TX_M_LN_3	P102	HDMI_CK-/ DP1_LANE3-	HDMI_CK-	HDMI differential clock output pair
M1	N/A	HDMI_TX_P_LN_3	P101	HDMI_CK+/ DP1_LANE3+	HDMI_CK+	
W2	N/A	HDMI_HPD	P104	HDMI_HPD/ DP1_HPD	HDMI_HPD	HDMI Hot Plug Detect input
I2C Dedicate for HDMI/DP1						
P3/ V2	N/A	HDMI_DDC_SDA/ HDMI_AUX_N	P106	HDMI_CTRL_DAT/ DP1_AUX-	HDMI_CTRL_DAT	I2C Data
R3/ v1	N/A	HDMI_DDC_SCL/ HDMI_AUX_P	P105	HDMI_CTRL_CLK/ DP1_AUX+	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.

2.1.7.2 DP++ Operation Over SMARC HDMI Pins

The *SMARC-iMX8M HDMI* pins could alternatively be used for DisplayPort++ (DP++) operation. Dual Mode (*HDMI* and DisplayPort on the same pins) implementations could be realized. This is desirable for SOCs that natively implement this capability. The *HDMI_CTRL_DAT* and *HDMI_CTRL_CK* lines are DC coupled, but the *DP_AUX+ /-* pair must be AC coupled. A set of *FET* switches is used on *SMARC-iMX8M* to sort this out. The FET gates are controlled by the *AUX_SEL* pin function.

The table below shows the *AUX_SEL* signals.

NXP i.MX8M CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
			P107	DP1_AUX_SEL	DP1_AUX_SEL	Pulled to GND on Carrier for DP operation in Dual Mode (DP++) implementations. Driven to 1.8V on Carrier for HDMI operation. Terminated on Module through 1M resistor to GND.

Note:

DP interface is implemented on *SMARC-iMX8M*, but not validated! Awaits NXP formal release.

2.1.7.3 Display Port Signals

The table below shows the Display Port related signals.

Edge Golden Finder Signal Name	Direction	Coupling Tolerance	Description
DP1_LANE[0:3]+	Output	AC Coupled off module	DP Data Pair [0:3] positive
DP1_LANE[0:3]-	Output	AC Coupled off module	DP Data Pair [0:3] negative
DP1_HPD	Input	DC coupled CMOS 1.8V	DP Hot Plug Detect input
DP1_AUX-	Bi-Dir	AC Coupled on module	DP AUX Channel (- part of pair)
DP1_AUX+	Bi-Dir	AC Coupled on module	DP AUX Channel (+ part of pair)
DP1_AUX_SEL	Input	DC coupled CMOS 1.8V	Pulled to GND on Carrier for DP operation in Dual Mode (DP++) implementations. Driven to 1.8V on Carrier for HDMI operation. Terminated on Module through 1M resistor to GND.

2.1.8 USB Interface

The *Embedian SMARC-iMX8M* module supports five *USB 2.0* ports (*USB 0:4*) and two *USB 3.0* ports (*USB[2:3]_SS*). The *USB 2.0* and *USB 3.0 IP* in *i.MX8M* processor are independent. A Microchip *USB2514* is used to expand four USB 2.0 ports from *i.MX8M 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 and *USB3* port.

The following figure shows the *USB 0:4 (USB2.0)* and *USB[2:3]_SS (USB3.0)* block diagram.

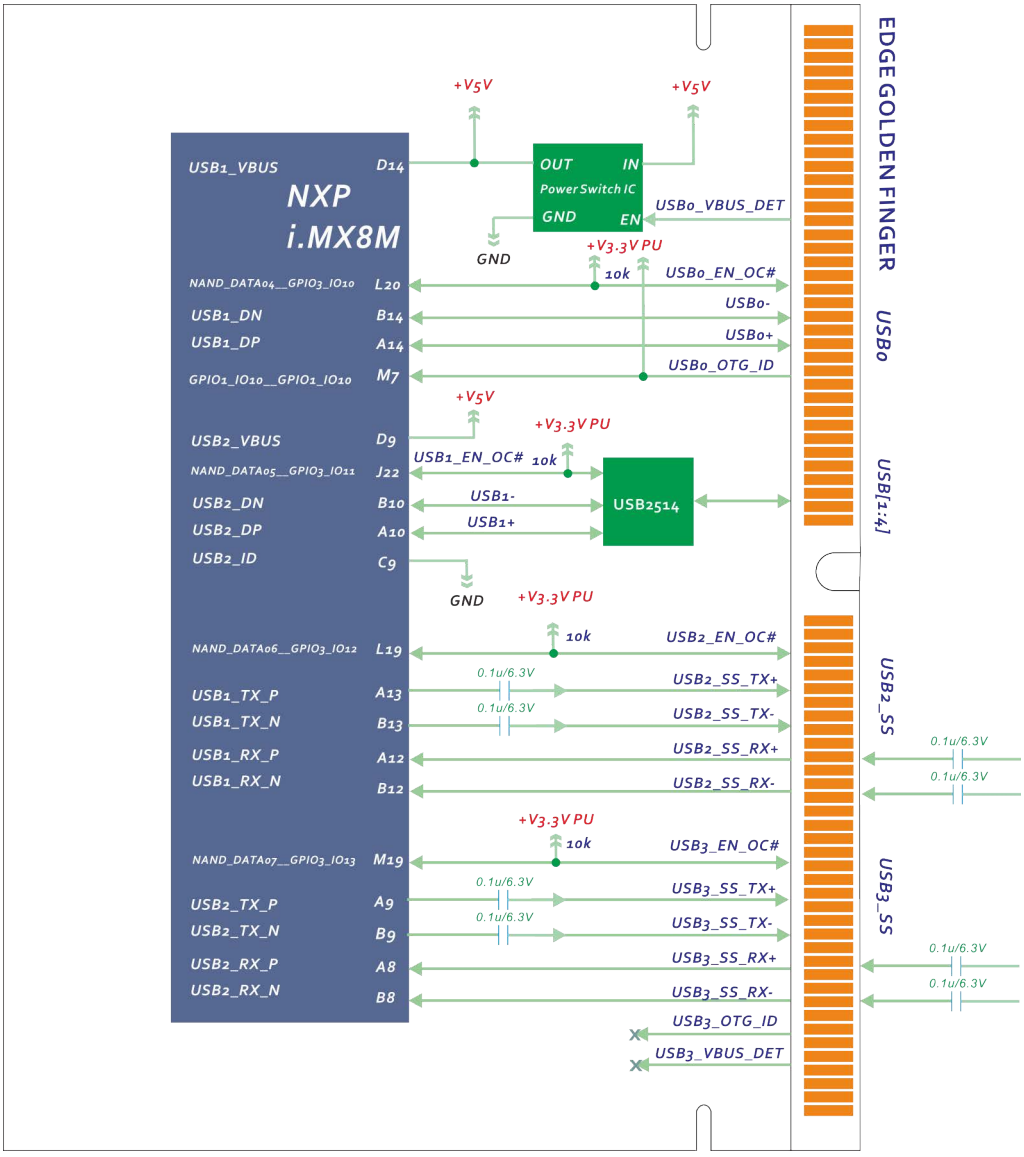


Figure 4. USB0 and USB1 Block Diagram

USB interface signals are exposed on the *SMARC-iMX8M* edge connector as shown below:

NXP i.MX8M CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
USB0 Port (USB 2.0 OTG)						
A14		USB1_DP	P60	USB0+	USB0+	USB0 data pair
B14		USB1_DN	P61	USB0-	USB0-	
L20	ALT5	NAND_DATA04__ GPIO3_IO10	P62	USB0_EN_OC#	USB0_EN_OC#	USB0 power enable/over current indication signal
D14		Turn on USB_OTG_VBUS	P63	USB0_VBUS_ DET	USB0_VBUS_ DET	USB0 host power detection, when this port is used as a device.
M7	ALT5	GPIO1_IO10__ GPIO1_IO10	P64	USB0_OTG_ID	USB0_OTG_ID	USB0 OTG ID input, active high

NXP i.MX8M CPU			SMARC-iMX8M 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-	
J22	ALT5	NAND_DATA05__ GPIO3_IO11			VBUS_DET#	USB2514 HUB VBUS Detect
			P69	USB2+	USB2+	USB_DN1 of USB2514
			P70	USB2-	USB2-	
			S68	USB3+	USB3+	USB_DN2 of USB2514
			S69	USB3-	USB3-	
			S35	USB4+		USB_DN4 of USB2514
			S36	USB4-		
From USB2514			P76	USB4_EN_OC#	USB2_EN_OC#	USB4 power enable/over current indication signal

NXP i.MX8M CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
USB2_SS Port (USB 3.0 Host)						
A13		USB1_TX_P	S71	USB2_SSTX+	USB2_SSTX+	USB2 transmit signal differential pair positive
B13		USB1_TX_N	S72	USB2_SSTX-	USB2_SSTX-	USB2 transmit signal differential pair negative
A12		USB1_RX_P	S74	USB2_SSRX+	USB2_SSRX+	USB2 receive signal differential pair positive
B12		USB1_RX_N	S75	USB2_SSRX-	USB2_SSRX-	USB2 receive signal differential pair negative
L19	ALT5	NAND_DATA06__ GPIO3_IO12	P71	USB2_EN_OC#	USB2_EN_OC#	USB2 power enable/over current indication signal

NXP i.MX8M CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
USB3_SS Port (USB3.0 OTG)						
A9		USB2_TX_P	S62	USB3_SSTX+	USB3_SSTX+	USB3 transmit signal differential pair positive
B9		USB2_TX_N	S63	USB3_SSTX-	USB3_SSTX-	USB3 transmit signal differential pair negative
A8		USB2_RX_P	S65	USB3_SSRX+	USB3_SSRX+	USB3 receive signal differential pair positive
B8		USB2_RX_N	S66	USB3_SSRX-	USB3_SSRX-	USB3 receive signal differential pair negative
M19	ALT5	NAND_DATA07__ GPIO3_IO13	P74	USB3_EN_OC#	USB3_EN_OC#	USB3 power enable/over current indication signal

Note:

1. *USB0 OTG* role switch in *i.MX8M* 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* 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 *USB0* is set as *otg* mode, *USB2_SS* will become *USB 2.0* only.
4. 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.*
USB[2:3]SSRX- *USB[2:3]SSRX+*	*Input*	*USB SS*	*Receive signal differential pairs for SuperSpeed USB data Coupling caps for RX pairs are* ***off-Module (on Carrier Board)***
USB[2:3]SSTX- *USB[2:3]SSTX+*	*Output*	*USBSS*	*Transmit signal differential pairs for SuperSpeed USB data Coupling caps for RX pairs are* ***on-Module***

2.1.8.2 USB[0:3]_EN_OC# Discussion

The Module *USB[0:3]_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[0:3]_EN_OC#* low to disable the power delivery to the *USBx* device.
3) The Module floats *USB[0:3]_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[0:3]_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[0:3]_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[0:3]_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[0:3]_EN_OC# pins may be left unused, or they may be used as *USB[0:3]* power enables, without making use of the over-current detect Module input feature.

The *SMARC-iMX8M* 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[0:3]_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[0:3]_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[0:3]_EN_OC#* pin.

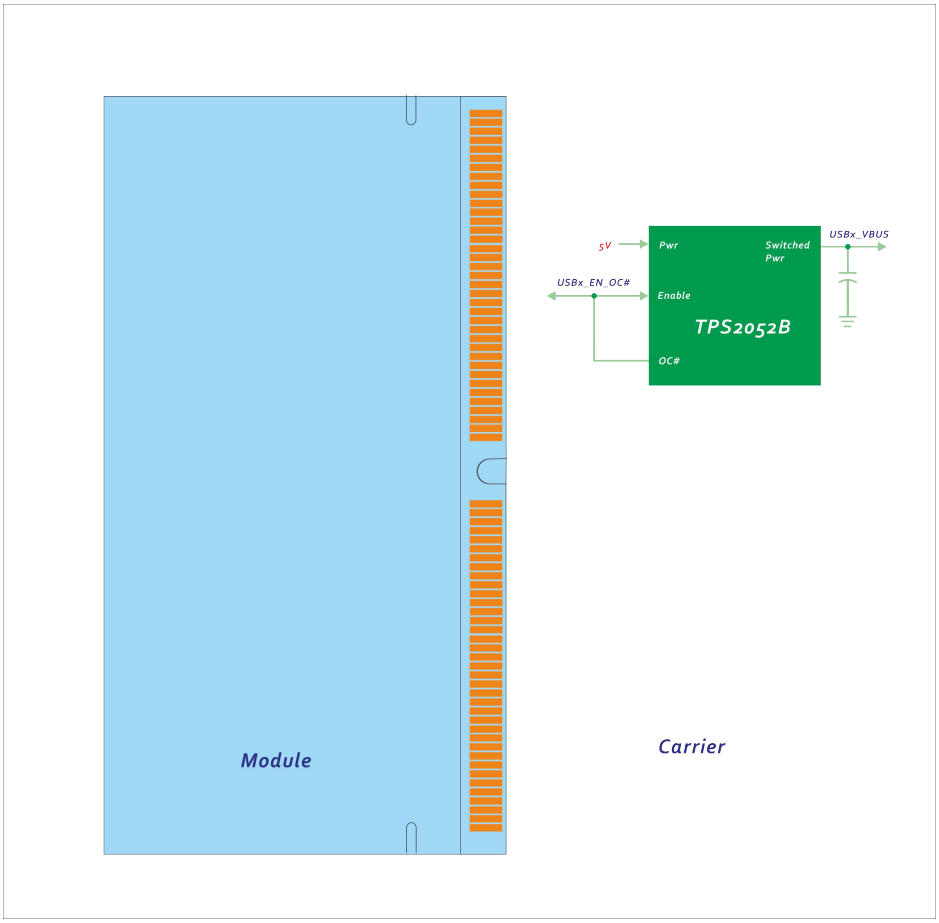


Figure 5. USB Power Distribution Implementation on Carrier

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

The *SMARC-iMX8M* module supports one Gigabit Ethernet (*10/100/1000*Mbps) interfaces. 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_MDI0±* to *GBE0_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.

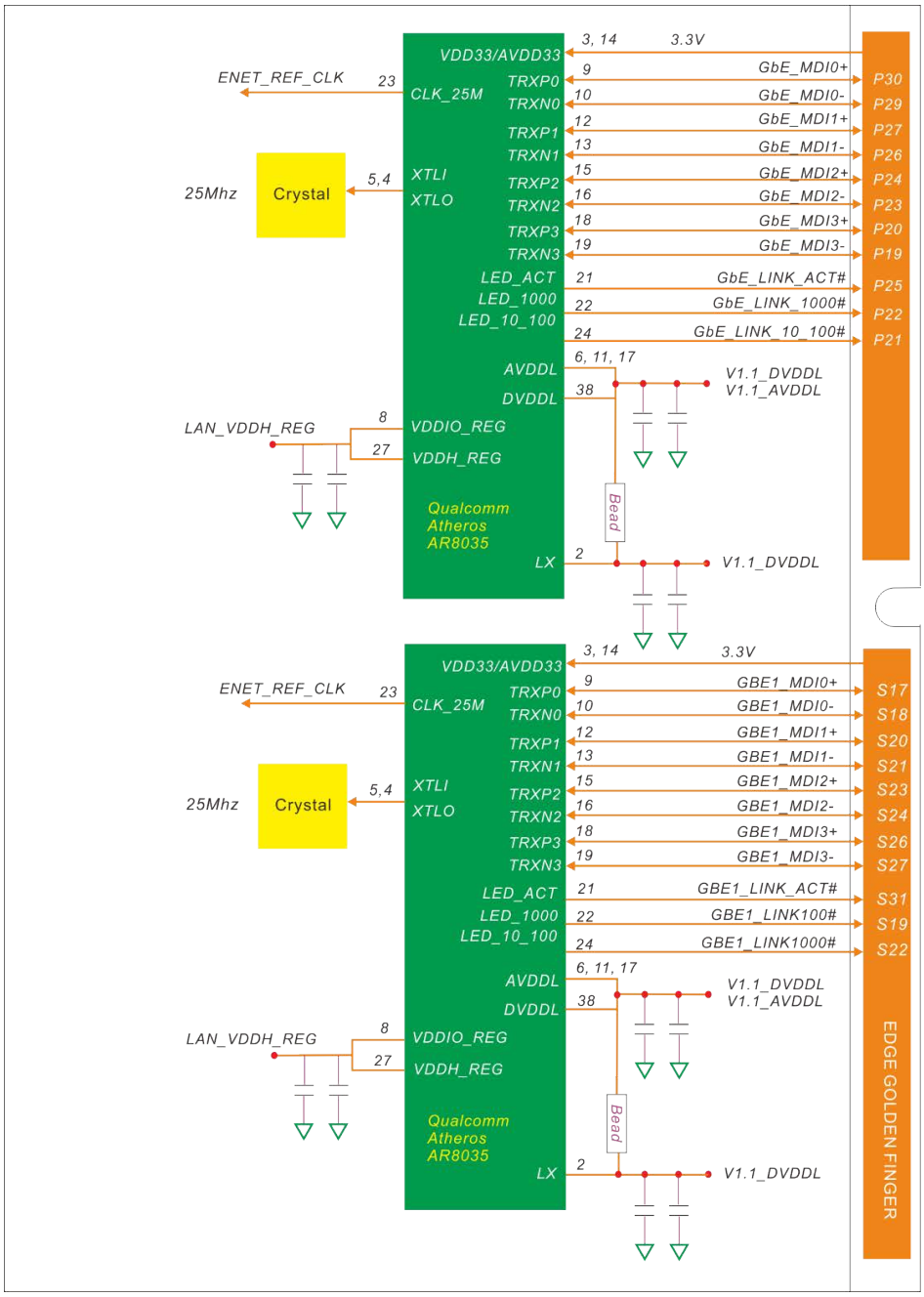


Figure 6: Gigabit Ethernet Connection from i.MX8M to Qualcomm Atheros AR8035

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

NXP i.MX8M CPU			Qualcomm AR8035		Net Names	Note
Ball	**Mode**	**Pin Name**	**Pin#**	**Pin Name**		
Gigabit LAN						
N19	ALT0	ENET_MDIO__ ENET1_MDIO	39	MDIO	ENET_MDIO	Serial Management Interface data input/output
N20	ALT0	ENET_MDC__ ENET1_MDC	40	MDC	ENET_MDC	Serial Management Interface clock
U19	ALT0	ENET_RD0__ ENET1_RGMII_RD0	29	RXD0	RMII_RD0	Bit 0 of the 4 data bits that are sent by the transceiver on the receive path.
U21	ALT0	ENET_RD1__ ENET1RGMII_RD1	28	RXD1	RMII_RD1	Bit 1 of the 4 data bits that are sent by the transceiver on the receive path.
U20	ALT0	ENET_RD2__ ENET1_RGMII_RD2	26	RXD2	RMII1_RD2	Bit 2 of the 4 data bits that are sent by the transceiver on the receive path.
V19	ALT0	ENET_RD3__ ENET1_RGMII_RD3	25	RXD3	RGMII_RD3	Bit 3 of the 4 data bits that are sent by the transceiver on the receive path.
T20	ALT0	ENET_RXC__ ENET1_RGMII_RXC	31	RX_CLK	RGMII_RXC	Reference clock
T21	ALT0	ENET_RX_CTL__ ENET1_RGMII_RX_ CTL	30	RX_DV	RGMII_RX_CTL	Indicates both the receive data valid (RXDV) and receive error (RXER) functions per the RGMII specification.

NXP i.MX8M CPU			Qualcomm AR8035		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
Gigabit LAN						
P19	ALT0	ENET_TX_CTL__ ENET1_RGMII_TX_ CTL	32	TX_EN	RGMII_TX_CT L	Indicates that valid transmission data is present on TXD[3:0].
R20	ALT0	ENET_TD0__ ENET1_RGMII_TD0	34	TXD0	RGMII_TD0	The MAC transmits data to the transceiver using this signal.
R21	ALT0	ENET_TD1__ ENET1_RGMII_TD1	35	TXD1	RGMII_TD1	The MAC transmits data to the transceiver using this signal.
R19	ALT0	ENET_TD2__ ENET1_RGMII_TD2	36	TXD2	RGMII_TD2	The MAC transmits data to the transceiver using this signal.
P20	ALT0	ENET_TD3__ ENET1_RGMII_TD3	37	TXD3	RGMII_TD3	The MAC transmits data to the transceiver using this signal.
T19	ALT0	ENETI_TXC__ ENET1_RGMII_TXC	33	GTX_CLK	RGMII_TXC	Used to latch data from the MAC into the PHY. 1000BASE-T: 125MHz 100BASE-TX: 25MHz 10BASE-T: 2.5MHz

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 PHY					
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 PHY					
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

2.1.9.1. Gigabit LAN Signals

The table below shows the Gigabit LAN related signals.

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
GBE_MDI0+ GBE_MDI0-	Bi-Dir	GBE_MDI	Bi-directional transmit/receive pair 0 to magnetics (Media Dependent Interface)
GBE_MDI1+ GBE_MDI1-	Bi-Dir	GBE_MDI	Bi-directional transmit/receive pair 1 to magnetics (Media Dependent Interface)
GBE_MDI2+ GBE_MDI2-	Bi-Dir	GBE_MDI	Bi-directional transmit/receive pair 2 to magnetics (Media Dependent Interface)
GBE_MDI3+ GBE_MDI3-	Bi-Dir	GBE_MDI	Bi-directional transmit/receive pair 3 to magnetics (Media Dependent Interface)
GBE_100#	Output OD	CMOS 3.3V	Link Speed Indication LED for 100Mbps Could be able to sink 24mA or more Carrier LED current
GBE_1000#	Output OD	CMOS 3.3V	Link Speed Indication LED for 1000Mbps Could be able to sink 24mA or more Carrier LED current
GBE_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
GBE_CTREF	Output	Reference Voltage	Center-Tap reference voltage for GBE0 Carrier board Ethernet magnetic (not required by the Module GBE PHY)

2.1.9.2. 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. PCIe_A and PCIe_B Interfaces

The *SMARC-iMX8M* offers *two PCI* Express *x1* lanes. The *PCIe* signals are routed from the NXP® *i.MX8M* processor to the *PCI* Express port *A* and *B* of the *SMARC-iMX8M* edge finger. These signals support *PCI* Express Gen. 2.1 interfaces at 5 Gb/s and are backward compatible to Gen. 1.1 interfaces at 2.5 Gb/s. Only x1 *PCI Express* link configuration is possible. Diodes *PI6CFGL201B* clock generators are used on each *PCIe* port to make *PCIe* reference clock *HCSL* signals.

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

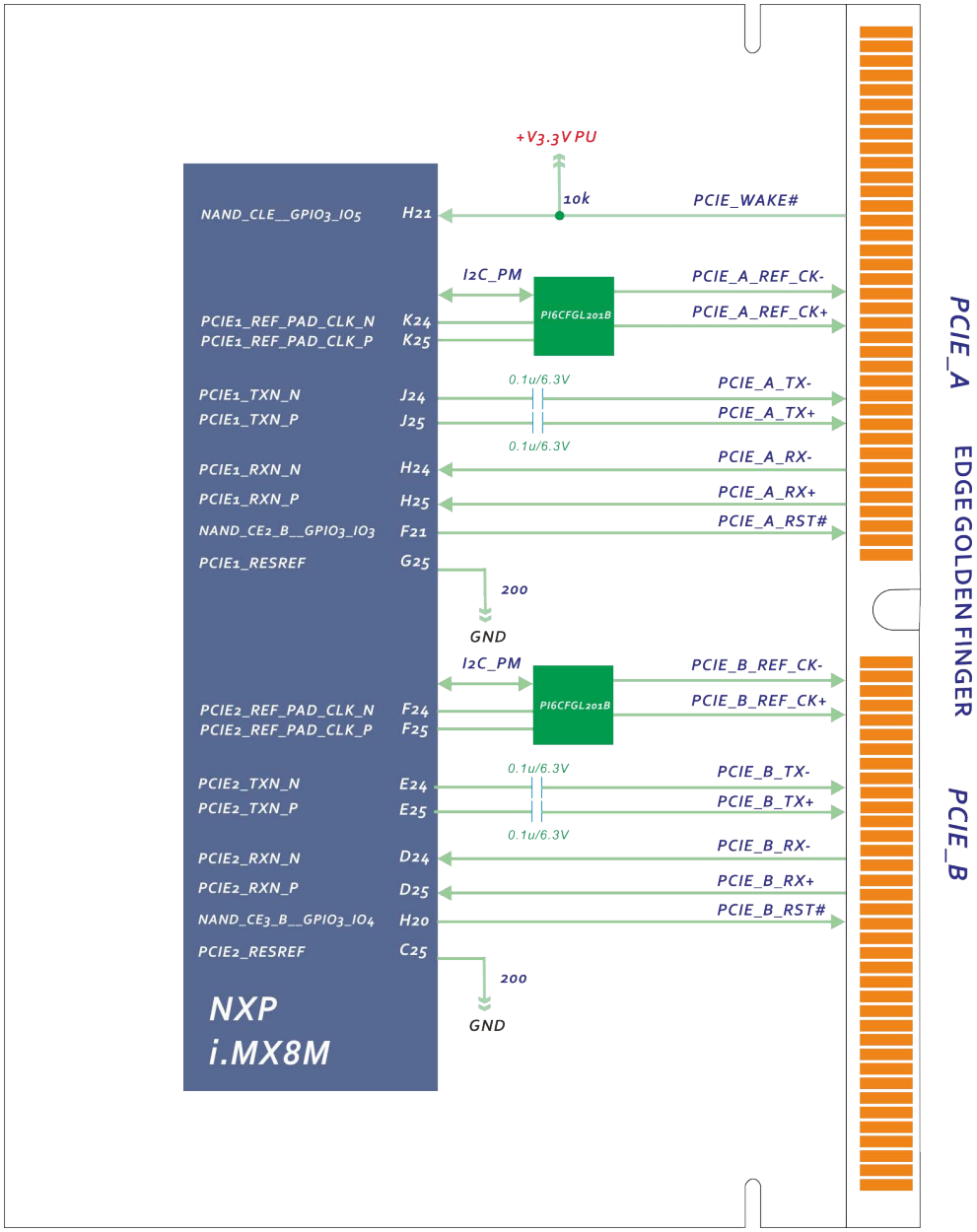


Figure 7. PCI Express Block Diagram

PCI Express interface signals are exposed on the *SMARC-iMX8M* edge connector as shown below:

NXP i.MX8M CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
H21	ALT5	NAND_CLE__ GPIO3_IO5	S146	PCIE_WAKE#	PCIE_WAKE#	PCIe wake up interrupt to host
PCI Express Port A						
F21	ALT5	NAND_CE2_B__ GPIO3_IO3	P75	PCIE_A_RST#	PCIE_A_RST#	Reset Signal for external devices.
K25		PCIE1_REF_PAD _CLK_P	P83	PCIE_A_REFCK+	PCIE_A_REFCK+	Differential PCI Express Reference Clock Signals for Lanes A
K24		PCIE1_REF_PAD _CLK_P	P84	PCIE_A_REFCK-	PCIE_A_REFCK-	
H25		PCIE1_RXN_P	P86	PCIE_A_RX+	PCIE_A_RX+	Differential PCIe Link A receive data pair 0
H24		PCIE1_RXN_N	P87	PCIE_A_RX-	PCIE_A_RX-	
J25		PCIE1_TXN_P	P89	PCIE_A_TX+	PCIE_A_TX+	Differential PCIe Link A transmit data pair 0
J24		PCIE1_TXN_N	P90	PCIE_A_TX-	PCIE_A_TX-	

NXP i.MX8M CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
PCI Express Port B						
H20	ALT5	NAND_CE3_B__ GPIO3_IO4	S76	PCIE_B_RST#	PCIE_B_RST#	Reset Signal for external devices.
F25		PCIE2_REF_PAD _CLK_P	S84	PCIE_B_REFCK+	PCIE_B_REFCK+	Differential PCI Express Reference Clock Signals for Lanes B
F24		PCIE2_REF_PAD _CLK_P	S85	PCIE_B_REFCK-	PCIE_B_REFCK-	
D25		PCIE2_RXN_P	S87	PCIE_B_RX+	PCIE_B_RX+	Differential PCIe Link B receive data pair 0
D24		PCIE2_RXN_N	S88	PCIE_B_RX-	PCIE_B_RX-	
E25		PCIE2_TXN_P	S90	PCIE_B_TX+	PCIE_B_TX+	Differential PCIe Link B transmit data pair 0
E24		PCIE2_TXN_N	S91	PCIE_B_TX-	PCIE_B_TX-	

2.1.10.1. PCIe_Link Signals

The table below shows the *PCIe_Link A and B* 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*
PCI Express Port B			
PCIE_B_TX+ *PCIE_B_TX-*	*Output*	*HCSL PCIe*	*Differential PCIe Link B transmit data pair 0* *Series coupling caps is on the Module* *Caps is 0402 package 0.1uF*
PCIE_B_RX+ *PCIE_B_RX-*	*Input*	*HCSL PCIe*	*Differential PCIe Link B receive data pair 0* *No coupling caps on Module*
PCIE_B_REFCK+ *PCIE_B_REFCK-*	*Output*	*HCSL PCIe*	*Differential PCIe Link B reference clock output* *DC coupled*
PCIE_B_RST#	*Output*	*CMOS* *3.3V*	*PCIe Port B reset output*

2.1.10.2. 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

2.1.11. MIPI/CMOS Serial Camera Interface (MIPI_CSI)

The *NXP® i.MX8M* 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 *SMARC* specification defines serial and parallel camera interface on the same pins. We can either implement it as serial or parallel camera interfaces. The camera interface on *SMARC-iMX8M* 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 4-lane *MIPI-CSI2* supports 5M pixel at 15 fps, 1080p30, 720p60, VGA at 60 fps o Maximum bit rate of 1.5 Gbp.

The following figure shows the serial camera interface block diagram.

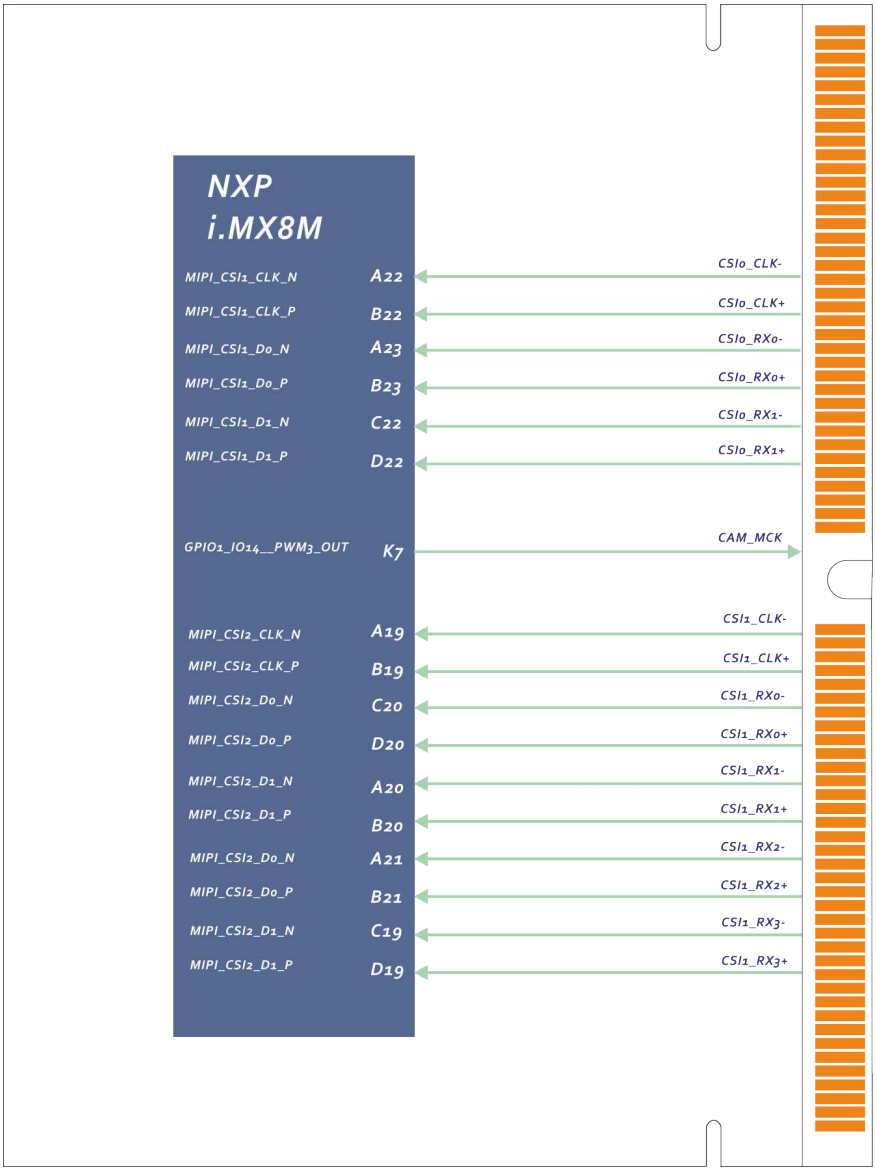


Figure 8. MIPI/Serial Camera Interface Block Diagram

MIPI/Serial Camera interface signals are exposed on the *SMARC-iMX8M* edge connector as shown below:

NXP i.MX8M CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
MIPI/Serial Camera Interface 1 (CSI0)						
K7	ALT5	GPIO1_IO14__ PWM3_OUT	S6	CAM_MCK	CAM_MCK	Master clock output for CSI camera support
A22	ALT0	MIPI_CSI1_CLK_N	S9	CSI0_CK-	CSI0_CK-	CSI0 differential clock inputs
B22	ALT0	MIPI_CSI1_CLK_P	S8	CSI0_CK+	CSI0_CK+	
A23	ALT0	MIPI_CSI1_D0_N	S12	CSI0_RX0-	CSI0_D0-	
B23	ALT0	MIPI_CSI1_D0_P	S11	CSI0_RX0+	CSI0_D0+	CSI0 differential data inputs
C22	ALT0	MIPI_CSI1_D1_N	S15	CSI0_RX1-	CSI0_D1-	
D22	ALT0	MIPI_CSI1_D1_P	S14	CSI0_RX1+	CSI0_D1+	
MIPI/Serial Camera Interface 2 (CSI1)						
A19	ALT0	MIPI_CSI2_CLK_N	P4	CSI1_CK-	CSI1_CK-	CSI1 differential clock inputs
B19	ALT0	MIPI_CSI2_CLK_P	P3	CSI1_CK+	CSI1_CK+	
C20	ALT0	MIPI_CSI2_D0_N	P8	CSI1_RX0-	CSI1_D0-	
D20	ALT0	MIPI_CSI2_D0_P	P7	CSI1_RX0+	CSI1_D0+	
A20	ALT0	MIPI_CSI2_D1_N	P11	CSI1_RX1-	CSI1_D1-	CSI1 differential data inputs
B20	ALT0	MIPI_CSI2_D1_P	P10	CSI1_RX1+	CSI1_D1+	
A21	ALT0	MIPI_CSI2_D2_N	P14	CSI1_RX2-	CSI1_D2-	
B21	ALT0	MIPI_CSI2_D2_P	P13	CSI1_RX2+	CSI1_D2+	
C19	ALT0	MIPI_CSI2_D3_N	P17	CSI1_RX3-	CSI1_D3-	
D19	ALT0	MIPI_CSI2_D3_P	P16	CSI1_RX3+	CSI1_D3+	

2.1.11.1. 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 CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
I2C_CAM0						
G7	ALT0	I2C2_SCL__ I2C2_SCL	S5	CSI0_TX+/ I2C_CAM0_CK	I2C_CAM0_CK	
F7	ALT0	I2C2_SDA__ I2C2_SDA	S7	CSI0_TX-/ I2C_CAM0_DAT	I2C_CAM0_DAT	
I2C_CAM1						
F8	ALT0	I2C4_SCL__ I2C4_SCL	S1	I2C_CAM1_CK	I2C_CAM1_CK	
F9	ALT0	I2C4_SDA__ I2C4_SDA	S2	I2C_CAM1_DAT	I2C_CAM1_DAT	
CSI Clock Output						
K7	ALT6	GPIO1_IO14__ CCMSRCGPCMIX_ 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

2.1.11.2. 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

2.1.12 SD/SDMMC Interface

SMARC-iMX8M 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-iMX8M* 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.

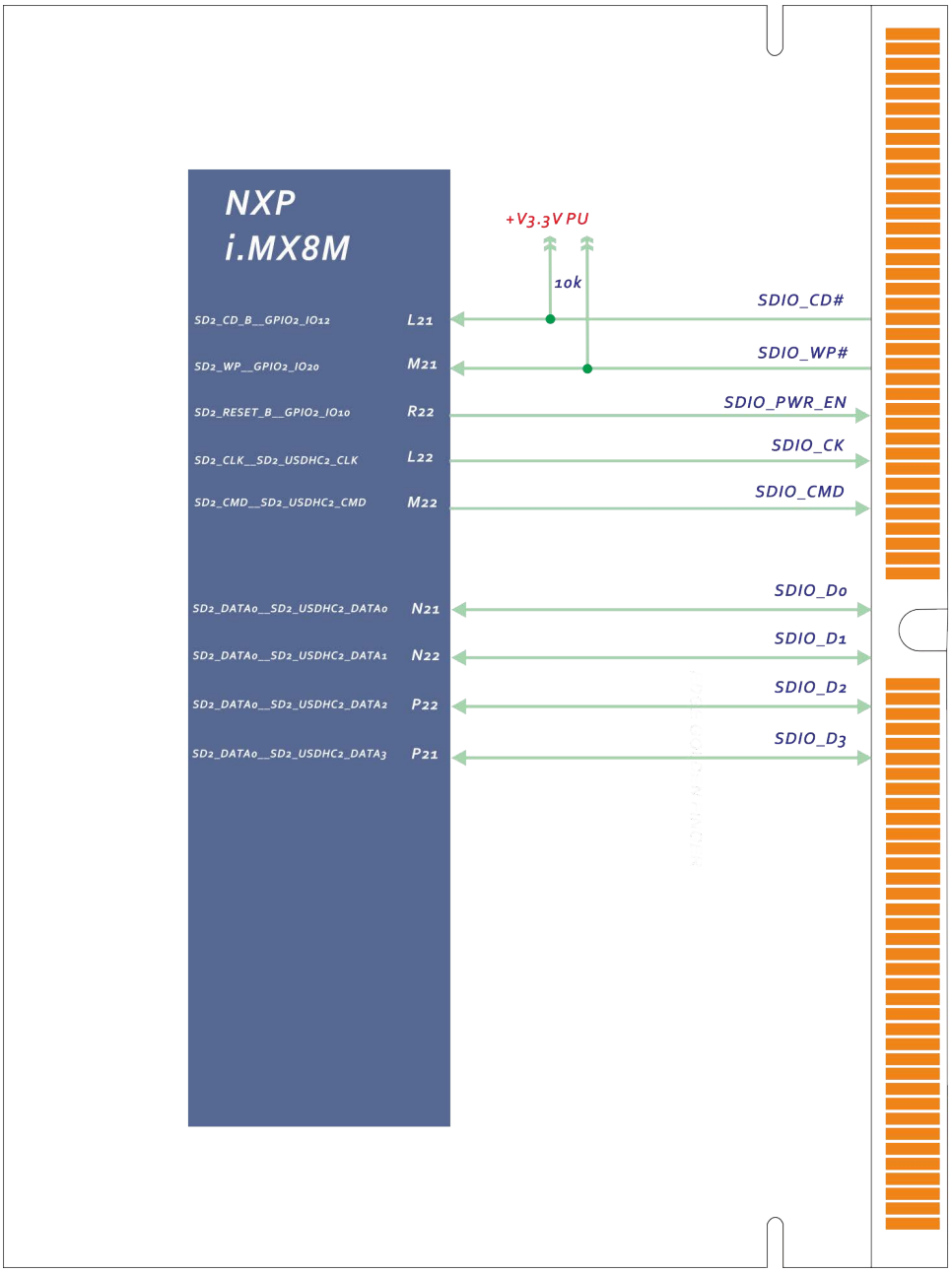


Figure 9. SD/SDIO/eMMC Interface Block Diagram

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						
N22	ALT0	SD2_DATA0__ SD2_USDHC2_ DATA0	P39	SDIO_D0	SDIO_D0	SDIO Data 0
N21	ALT0	SD2_DATA1__ SD2_USDHC2_ DATA1	P40	SDIO_D1	SDIO_D1	SDIO Data 1
P22	ALT0	SD2_DATA2__ SD2_USDHC2_ DATA2	P41	SDIO_D2	SDIO_D2	SDIO Data 2
P21	ALT0	SD2_DATA3__ SD2_USDHC2_ DATA3	P42	SDIO_D3	SDIO_D3	SDIO Data 3
M21	ALT5	SD2_WP__ GPIO2_IO20	P33	SDIO_WP	SDIO_WP	SDIO write protect signal
M22	ALT0	SD2_CMD__ SD2_USDHC2_ CMD	P34	SDIO_CMD	SDIO_CMD	SDIO Command signal
L21	ALT5	SD1_CD__ GPIO2_IO12	P35	SDIO_CD#	SDIO_CD#	SDIO card detect
L22	ALT0	SD2_CLK__ SD2_USDHC2_ CLK	P36	SDIO_CK	SDIO_CK	SDIO Clock Signal
R22	ALT5	SD2_RESET_B__ GPIO2_IO10	P37	SDIO_PWR_EN	SDIO_PWRE N	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.

2.1.12.1. 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*.

2.1.13 SPI/eSPI Interface

The *SMARC-iMX8M* module supports two *NXP i.MX8M SPI* interfaces that are available off-Module for general purpose use. One of them is implemented as *eSPI* interface by *SMARC 2.0* definition. 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. *eSPI* devices will share the "E*SPI_IO_0*", "*ESPI_IO_1*", "*ESPI_IO_2*", "*ESPI_IO_3*" and "E*SPI_CK*" pins, but each device will have its own chip select pin. The chip select signal is also a low active signal.
The *SPI* to *CAN* bus bridge uses the *SPI0* interface with different chip select signals (*SS2#, SS#3*).

The *SPI* interface is diagramed below.

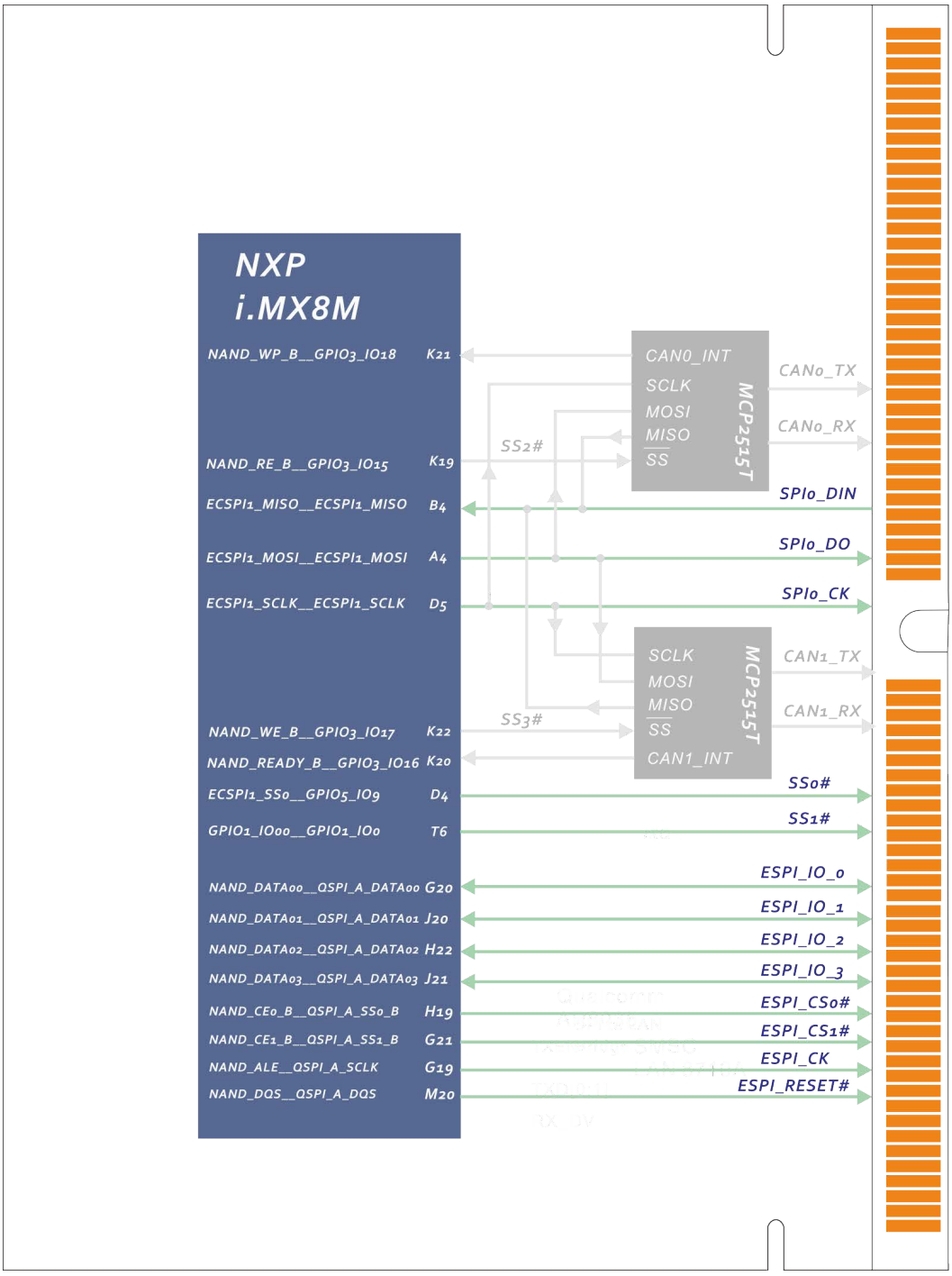


Figure 10: SPI Interface Block Diagram

SPI 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		
SPI0 Port						
D4	ALT5	ECSPI1_SS0__ GPIO5_IO9	P43	SPI0_CS0#	SPI0_CS0#	SPI0 Master Chip Select 0 output
T6	ALT5	GPIO1_IO00__ GPIO1_IO0	P31	SPI0_CS1#	SPI0_CS1#	SPI0 Master Chip Select 1 output
D5	ALT0	ECSPI1_SCLK__ ECSPI1_SCLK	P44	SPI0_CK	SPI0_SCLK	SPI0 Master Clock output
B4	ALT0	ECSPI1_MISO__ ECSPI1_MISO	P45	SPI0_DIN	SPI0_DIN	SPI0 Master Data input (input to CPU, output from SPI device)
A4	ALT0	ECSPI1_MOSI__ ECSPI1_MOSI	P46	SPI0_DO	SPI0_DO	SPI0 Master Data output (output from CPU, input to SPI device)
K19	ALT5	NAND_RE_B__ GPIO3_IO15				Chip select 2 for SPI to CAN0 Bridge
K22	ALT5	NAND_WE_B__ GPIO3_IO17				Chip select 3 for SPI to CAN1 Bridge

NXP i.MX8M CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
eSPI Port						
H19	ALT1	NAND_CE0_B__ QSPI_A_SS0_B	P54	ESPI_CS0#	ESPI_CS0#	QSPI Master Chip Select 0 output
G21	ALT1	NAND_CE1_B__ QSPI_A_SS1_B	P55	ESPI_CS1#	ESPI_CS1#	QSPI Master Chip Select 1 output
G19	ALT1	NAND_ALE__ QSPI_A_SCLK	P56	ESPI_CK	ESPI_SCLK	QSPI Master Clock output
G20	ALT1	NAND_DATA00__ QSPI_A_DATA00	P58	ESPI_IO_0	ESPI_IO_0	QSPI Master Data 0
J20	ALT1	NAND_DATA01__ QSPI_A_DATA01	P57	ESPI_IO_1	ESPI_IO_1	QSPI Master Data 1
G21	ALT1	NAND_DATA02__ QSPI_A_DATA02	S56	ESPI_IO_2	ESPI_IO_2	QSPI Master Data 2
J21	ALT1	NAND_DATA03__ QSPI_A_DATA03	S57	ESPI_IO_3	ESPI_IO_3	QSPI Master Data 3
M20	ALT1	NAND_DQS__ QSPI_A_DQS	S58	ESPI_RESET#	ESPI_RESET#	Reset the eSPI interface for both master and slaves.

2.1.13.1. SPI0 Signals

i.MX8M processor *ROM* 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)*

2.1.13.2. ESPI Signals

i.MX8M processor *ROM* does not support Q*SPI* device boot up either. The Carrier *QSPI* 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#	*Output*	*CMOS 1.8V*	*QSPI Master Chip Select 0 output*
ESPI_CS1#	*Output*	*CMOS 1.8V*	*QSPI Master Chip Select 1 output*
ESPI_CK	*Output*	*CMOS 1.8V*	*QSPI Master Clock output*
ESPI_IO_[0:3]	*Bi-Dir*	*CMOS 1.8V*	*QSPI Master Data input/output*
ESPI_RESET#	*Output*	*CMOS 1.8V*	*QSPI Reset Reset the QSPI interface for both master and slaves.*
ESPI_ALERT[0:1]#	*Input*	*CMOC 1.8V*	*Not Supported*

2.1.14. I2S Interface

The *SMARC-iMX8M* module uses *I2S* format for Audio signals. These signals are derived from the Synchronous Audio Interface (*SAI*) of the *NXP® i.MX8M* 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-iMX8M supports two I2S instances (I2S0 and I2S2). I2S interface signals are exposed on the *SMARC-iMX8M* 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		
H5	ALT0	SAI2_MCLK__ SAI2_MCLK	S38	AUDIO_MCK	AUD_MCLK	Master clock output to Audio codecs
I2S0 interface						
H4	ALT0	SAI2_TXFS__ SAI2_TX_SYNC	S39	I2S0_LRCK	I2S0_LRCK	Left& Right audio synchronization clock
G5	ALT0	SAI2_TXD0__ SAI2_TX_DATA0	S40	I2S0_SDOUT	I2S0_SDOUT	Digital audio Output
H6	ALT0	SAI2_RXD0__ SAI2_RX_DATA0	S41	I2S0_SDIN	I2S0_SDIN	Digital audio Input
J5	ALT0	SAI2_TXC__ SAI2_TX_BCLK	S42	I2S0_CK	I2S0_CK	Digital audio clock
I2S2 interface						
G3	ALT0	SAI3_TXFS__ SAI3_TX_SYNC	S50	HDA_SYNC/ I2S2_LRCK	I2S2_LRCK	Left& Right audio synchronization clock
C3	ALT0	SAI3_TXD__ SAI3_TX_DATA0	S51	HDA_SDO/ I2S2_SDOUT	I2S2_SDOUT	Digital audio Output
F3	ALT0	SAI3_RXD__ SAI3_RX_DATA0	S52	HDA_SDI/ I2S2_SDIN	I2S2_SDIN	Digital audio Input
C4	ALT0	SAI3_TXC__ 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.

2.1.14.1 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

2.1.15. Asynchronous Serial Port (UARTs)

The *SMARC-iMX8M* 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-iMX8M* 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 CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
SER0 Port						
E5	ALT1	ECSPI2_MOSI__ UART4_DCE_TX	P129	SER0_TX	SER0_TX	Asynchronous serial port data out
C5	ALT1	ECSPI2_SCLK__ UART4_DCE_RX	P130	SER0_RX	SER0_RX	Asynchronous serial port data in
B5	ALT1	ECSPI2_MISO__ UART4_DCE_ CTS_B	P131	SER0_RTS#	SER0_RTS#	Request to Send handshake line for SER0
A5	ALT1	ECSPI2_SS0__ UART4_DCE_ RTS_B	P132	SER0_CTS#	SER0_CTS#	Clear to Send handshake line for SER0
SER1 Port						
B7	ALT0	UART3_TXD__ UART3_DCE_TX	P134	SER1_TX	SER1_TX	Asynchronous serial port data out
A6	ALT0	UART3_RXD__ UART3_DCE_RX	P135	SER1_RX	SER1_RX	Asynchronous serial port data in

NXP i.MX8M CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
SER2 Port						
D6	ALT0	UART2_TXD__ UART2_DCE_TX	P136	SER2_TX	SER2_TX	Asynchronous serial port data out
B6	ALT0	UART2_RXD__ UART2_DCE_RX	P137	SER2_RX	SER2_RX	Asynchronous serial port data in
C6	ALT1	UART4_RXD__ UART2_DCE_CTS	P138	SER2_RTS#	SER2_RTS#	Request to Send handshake line for SER2
D7	ALT1	UART4_TXD__ UART2_DCE_RTS	P139	SER2_CTS#	SER2_CTS#	Clear to Send handshake line for SER2
SER3 Port (Debugging Port)						
A7	ALT0	UART1_TXD__ UART1_DCE_TX	P140	SER3_TX	SER3_TX	Asynchronous serial port data out
C7	ALT0	UART1_RXD__ UART1_DCE_RX	P141	SER3_RX	SER3_RX	Asynchronous serial port data in

2.1.15.1. 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*

2.1.16. 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), CAM1 (Camera 1) and *HDMI*. *SMARC-iMX8M* does not have *HDMI* interface, it defines five out of the six *I2C* buses and supports multiple masters and slaves in fast mode (400 KHz operation).

All *I2C* interfaces are implemented directly from *NXP i.MX8M* processor interfaces.

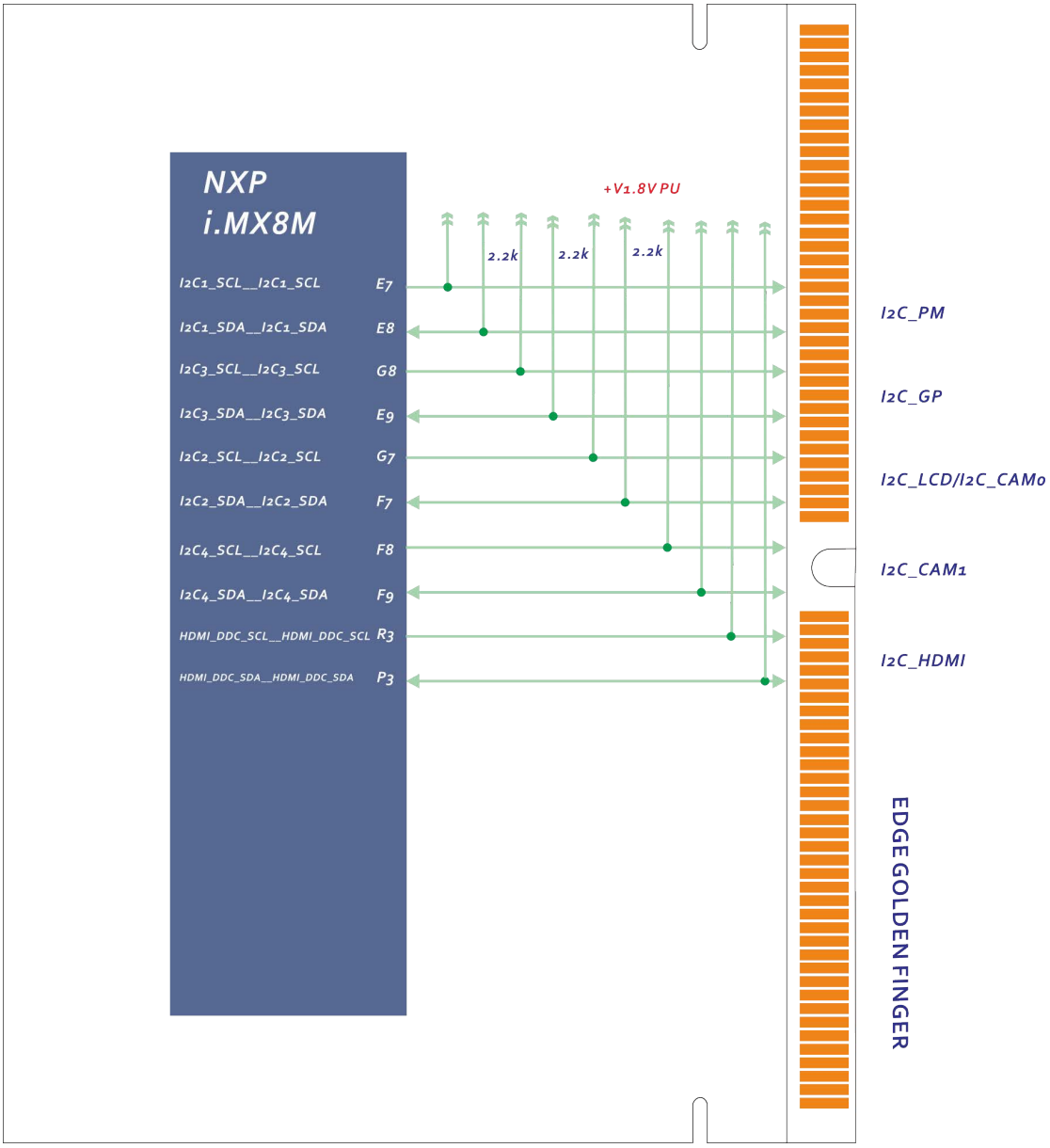


Figure 11. I2C Interface Block Diagram

This will be summarized below.

I2C Port		Primary Purpose	Alternative Use	I/O Voltage Level
Golden Finger Connector	i.MX8M 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	I2C2	Serial camera 0	General Purpose	CMOS 1.8V
I2C_CAM1	I2C4	Serial camera 1	General Purpose	CMOS 1.8V

Note:

1. The *2.2k* pull-up resistors for *I2C_SCL* and *I2C_SDA* signals are on module.
2. *I2C_LCD* and *I2C_CAM0* are using the same *I2C2* bus to avoid from adding an additional *I2C* switch IC on module.

The *I2C* 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		
I2C_PM						
E7	ALT0	I2C1_SCL__ I2C1_SCL	P121	I2C_PM_CK	I2C_PM_CK	Power management I2C bus clock
E8	ALT0	I2C1_SDA__ I2C1_SDA	P122	I2C_PM_DAT	I2C_PM_SDA	Power management I2C bus data
I2C_GP						
G8	ALT0	I2C3_SCL__ I2C3_SCL	S48	I2C_GP_CK	I2C_GP_CK	General purpose I2C bus clock
E9	ALT0	I2C3_SDA__ I2C3_SDA	S49	I2C_GP_DAT	I2C_GP_DAT	General purpose I2C bus data
I2C_LCD						
G7	ALT0	I2C2_SCL__ I2C2_SCL	S139	I2C_LCD_CK	I2C_LCD_CK	LCD display I2C bus clock
F7	ATL0	I2C2_SDA__ I2C2_SDA	S140	I2C_LCD_DAT	I2C_LCD_DAT	LCD display I2C bus data
I2C_CAM0						
G7	ALT0	I2C2_SCL__ I2C2_SCL	S5	I2C_CAM0_CK	I2C_CAM0_CK	Camera 0 I2C bus clock
F7	ALT0	I2C2_SDA__ I2C2_SDA	S7	I2C_CAM0_ DAT	I2C_CAM0_ DAT	Camera 0 I2C bus data
I2C_CAM1						
F8	ALT0	I2C4_SCL__ I2C4_SCL	S1	I2C_CAM1_CK	I2C_CAM1_CK	Camera 1 I2C bus clock
F8	ALT0	I2C4_SDA__ I2C4_SDA	S2	I2C_CAM1_ DAT	I2C_CAM1_ DAT	Camera 1 I2C bus data

NXP i.MX8M CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
I2C_HDMI						
F8	N/A	I2C4_SCL__ I2C4_SCL	P105	HDMI_CTRL_ CK	HDMI_CTRL_ CK	HDMI I2C bus clock
F8	N/A	I2C4_SDA__ I2C4_SDA	P106	HDMI_CTRL_ DAT	HDMI_CTRL_ DAT	HDMI I2C bus data

Note:

All *I2C* bus and 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_PM (I2C1) Bus						
1	Pericom PI6CFGL201BZDIE	PCIe Gen 1-2-3 Clock Generator	0x68	0xD1	0xD0	Clock Generator for PCIe Lane A
2	Pericom PI6CFGL201BZDIE	PCIe Gen 1-2-3 Clock Generator	0x6A	0xD5	0XD4	Clock Generator for PCIe Lane B
2	NXP MC34PF4210A1ES	PMIC	0x08	0x01	0x00	PMIC
3	Seiko S-35390A	Real-time clock IC	0x30	0x61	0x60	Real-Time Clock
I2C_GP						
1	On Semiconductor CAT24C32	EEPROM	0x50	0xA1	0xA0	General purpose parameter EEPROM, Serial number, etc in PICMG EEEP format
I2C_LCD						
1	TI SN65DSI84	MIPI_DSI to LVDS Bridge	0x2C	0x59	0x58	MIPI_DSI to LVDS Bridge IC

Note:

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

2.1.17. CAN Bus Interface

The Controller Area Network (*CAN*) is a serial communications protocol which efficiently supports distributed real-time control with a high level of security. The *SMARC-iMX8M* module implements two *CAN* bus interfaces from Microchip *MCP2515 SPI* to *CAN* interface IC.

The SPI bus used to interface with *MCP2515 CAN* controller is *SPI0*. The chip select *SS2#* is reserved for *CAN0* and *SS3#* is reserved for *CAN1*. Chip selects *SS0#* and *SS1#* are connected to *MXM* golden finger connector for users to use. The logic level for *CAN0/1 TX/RX* is 1.8V as defined in *SMARC* 2.0.

The following figure shows the *CAN bus* block diagram.

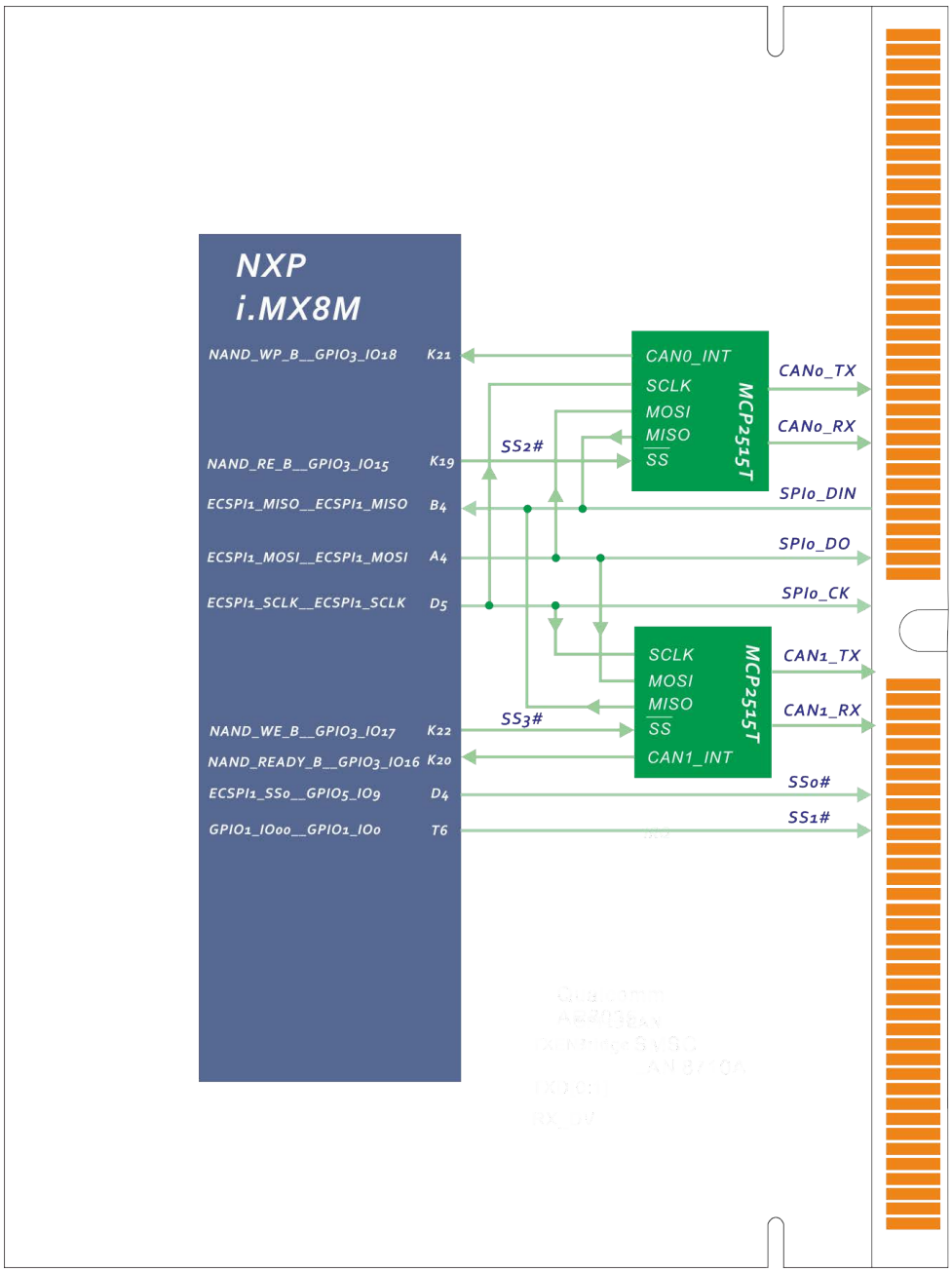


Figure 12: SMARC-iMX8M CAN Bus Diagram

2.1.17.1 CAN0 Bus Signals Data Flow

i.MX8M processor and *Micochip MXP2515T* implementation for CAN0 is shown in the following table:

NXP i.MX8M CPU			Microchip MCP2515T		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
B4	ALT0	ECSPI1_MISO__ ECSPI1_MISO	15	SO	SPI_CAN_SO	
A4	ALT0	ECSPI1_MOSI__ ECSPI1_MOSI	14	SI	SPI_CAN_SI	
D5	ALT0	ECSPI1_SCLK__ ECSPI1_SCLK	12	SCK	SPI_CAN_SCLK	
K19	ALT5	NAND_RE_B__ GPIO3_IO15	16	CS#	ECSPI1_SS2#	
K21	ALT5	NAND_WP_B__ GPIO3_IO18	11	INT#	CAN0_INT#	

2.1.17.2 CAN1 Bus Signals Data Flow

i.MX8M processor and *Micochip MXP2515T* implementation for CAN1 is shown in the following table:

NXP i.MX8M CPU			Microchip MCP2515T		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
B4	ALT0	ECSPI1_MISO__ ECSPI1_MISO	15	SO	SPI_CAN_SO	
A4	ALT0	ECSPI1_MOSI__ ECSPI1_MOSI	14	SI	SPI_CAN_SI	
D5	ALT0	ECSPI1_SCLK__ ECSPI1_SCLK	12	SCK	SPI_CAN_SCLK	
K22	ALT5	NAND_WE_B__ GPIO3_IO17	16	CS#	ECSPI1_SS3#	
K20	ALT5	NAND_READY_B__ GPIO3_IO16	11	INT#	CAN1_INT#	

2.1.17.3 SPI to CAN Bridge Signals Data Flow

The path from *Mrcrochip MCP2515T* to the golden finger edge connector is show in the following table.

Microchip MCP2515T		Golden Finger Edge Connector		Net Names	Note
Pin	Pin Name	Pin#	Pin Name		
CAN0 Bus					
19	*TXCAN*	*P143*	*CAN0_TX*	*CAN0_TX*	*CAN0 Transmit output*
20	*RXCAN*	*P144*	*CAN0_RX*	*CAN0_RX*	*CAN0 Receive input*
CAN1 Bus					
19	*TXCAN*	*P145*	*CAN1_TX*	*CAN1_TX*	*CAN1 Transmit output*
20	*RXCAN*	*P146*	*CAN1_RX*	*CAN1_RX*	*CAN1 Receive input*

By *SMARC* hardware specification, *CAN0* bus error condition signaling should be supported on the Module *GPIO8* (*P116*) pin. This is an active low input to the Module from the CAN bus transceiver. CAN1 bus error condition signaling should be supported on the Module *GPIO9* (*P117*) pin. This is an active low input to the Module from the CAN bus transceiver

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 ISO1050 for more details.

2.1.17.4. CAN0 BUS Signals

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
CAN0_TX	Output	CMOS 1.8V	CAN0 Transmit output
CAN0_RX	Input	CMOS 1.8V	CAN0 Receive input

2.1.17.5. CAN1 BUS Signals

Edge Golden Finder Signal Name	Direction	Type Tolerance	Description
CAN1_TX	Output	CMOS 1.8V	CAN1 Transmit output
CAN1_RX	Input	CMOS 1.8V	CAN1 Receive input

2.1.18. GPIOs

The *SMARC-iMX8M* 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, CAN Error Signaling 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 CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
GPIOs						
K4	ALT5	SAI5_MCLK__ GPIO3_IO25	P108	GPIO0/CAM0_PWR#	GPIO0	Camera 0 Power Enable, active low output
N4	ALT5	SAI5_RXFS__ GPIO3_IO19	P109	GPIO1/CAM1_PWR#	GPIO1	Camera 1 Power Enable, active low output
L5	ALT5	SAI5_RXC__ GPIO3_IO20	P110	GPIO2/CAM0_RST#	GPIO2	Camera 0 Reset, active low output
M5	ALT5	SAI5_RXD0__ GPIO3_IO21	P111	GPIO3/CAM1_RST#	GPIO3	Camera 1 Reset, active low output
L4	ALT5	SAI5_RXD1__ GPIO3_IO22	P112	GPIO4/HDA_RST#	GPIO4	HD Audio Reset, active low output
F6	ALT5	SPDIF_TX__ GPIO5_IO3	P113	GPIO5/PWM_OUT	GPIO5	PWM output
G6	ALT5	SPDIF_RX__ GPIO5_IO4	P114	GPIO6/TACHIN	GPIO6	Tachometer input (used with the GPIO5 PWM)
M4	ALT5	SAI5_RXD2__ GPIO3_IO23	P115	GPIO7/PCAM_FLD	GPIO7	PCAM_FLD (Field) signal input
K5	ALT5	SAI5_RXD3__ GPIO3_IO24	P116	GPIO8/CAN0_ERR#	GPIO8	CAN0 Error signal, active low input
E1	ALT5	SAI1_TXC__ GPIO4_IO11	P117	GPIO9/CAN1_ERR#	GPIO9	CAN1 Error signal, active low input
H1	ALT5	SAI1_TXFS__ GPIO4_IO10	P118	GPIO10	GPIO10	
A3	ALT5	SAI1_MCLK__ GPIO4_IO20	P119	GPIO11	GPIO11	

2.1.18.1. 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* 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	

2.1.19 Watchdog Timer Interface

i.MX8M features an internal *WDT*. Embedian's Linux kernel enables the internal *i.MX8M 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 CPU			SMARC-iMX8M Edge Golden Finger		Net Names	Note
Ball	Mode	Pin Name	Pin#	Pin Name		
Watchdog Timer						
J6	ALT5	GPIO1_IO15__ CCMSRCGPCMIX _CLKO2	S145	WDT_TIME_ OUT#	WDT_TIME_OUT#	Watchdog-Timer Output

2.1.20 JTAG

Figure 13 shows the *SMARC-iMX8M JTAG* connectors location and pin out.

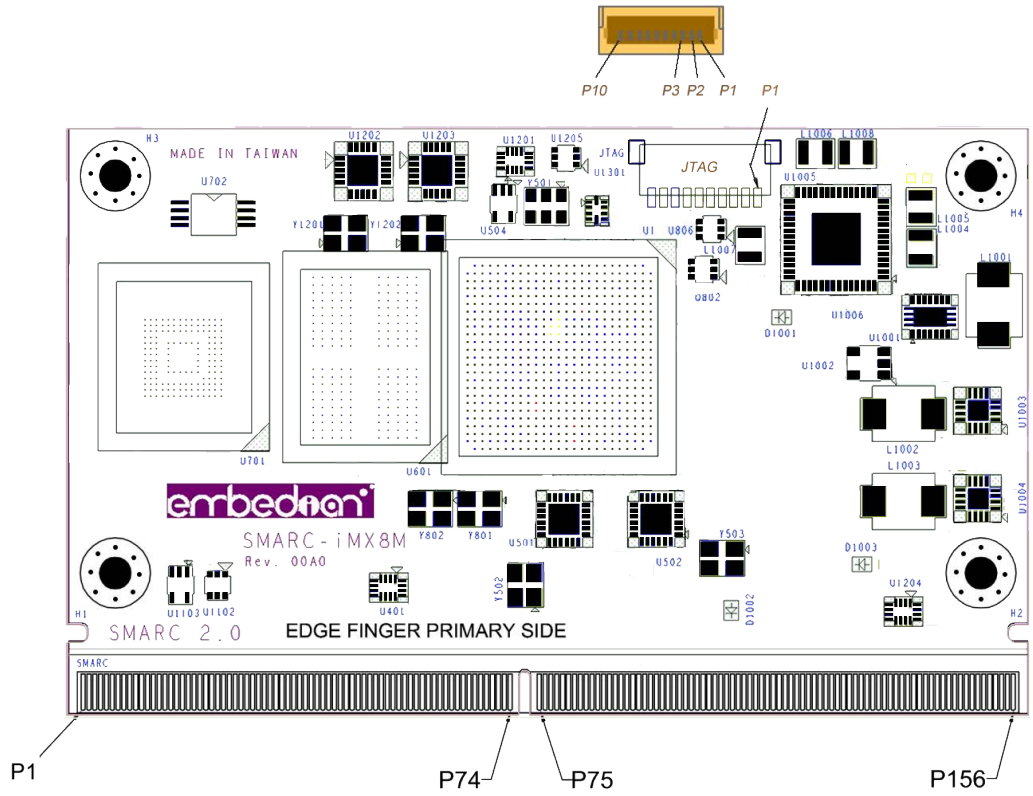


Figure 13: JTAG Connector Location and Pinout

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 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)
U6	ATL0	JTAG_TRST_B	2	nTRST	I	JTAG Reset, active low
V5	ALT0	JTAG_TMS	3	TMS	I	JTAG mode select
U5	ALT0	JTAG_TDO	4	TDO	O	JTAG data out
W5	ALT0	JTAG_TDI	5	TDI	I	JTAG data in
T5	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

2.1.21 Boot ID EEPROM

The *SMARC-iMX8M* 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 "SMC8MQ2G" = Embedian SMARC-iMX8M Computer on Module with Dual/Quad Core and 2GB LPDDR4 Configuration "SMC8MQ4G" = Embedian SMARC-iMX8M Computer on Module with Dual/Quad Core and 4GB LPDDR4 Configuration "SMC8ML2G" = Embedian SMARC-iMX8M Computer on Module with Quad Lite Core and 2GB LPDDR4 Configuration "SMC8ML4G" = Embedian SMARC-iMX8M Computer on Module with Quad Lite 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: WWYYMSD1nnnn Where: WW = 2 digit week of the year of production YY = 2 digit year of production MS = Module Serial Number D1/Q1/D2/Q2/UC/SC = CPU Core and DDR Configuration Variants nnnn = incrementing board number
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 (if any)
Available	32720	Available space for other non-volatile codes/data

2.2 SMARC-iMX8M 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* 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-iMX8M* Evaluation Carrier. The default baud rate setting is *115,200 8N1*.

SER3 pin out of the *SMARC-iMX8M* is shown below:

NXP i.MX8M CPU		SMARC-iMX8M Edge Golden Finger		Net Names	Notes
mode	Pin Name	Pin#	Pin Name		
SER3 (Debugging Port)					
ALT0	UART1_TXD__ UART1_DCE_TX	P140	SER3_TX	SER3_TX	Asynchronous serial port data out
ALT0	UART1_RXD__ UART1_DCE_RX	P141	SER3_RX	SER3_RX	Asynchronous serial port data in

2.3 Mechanical Specifications

2.3.1. Module Dimensions

The *SMARC-iMX8M* 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-iMX8M* Mechanical Drawings (Top View) and Figure 15: *SMARC-iMX8M* Mechanical Drawings (Bottom View))

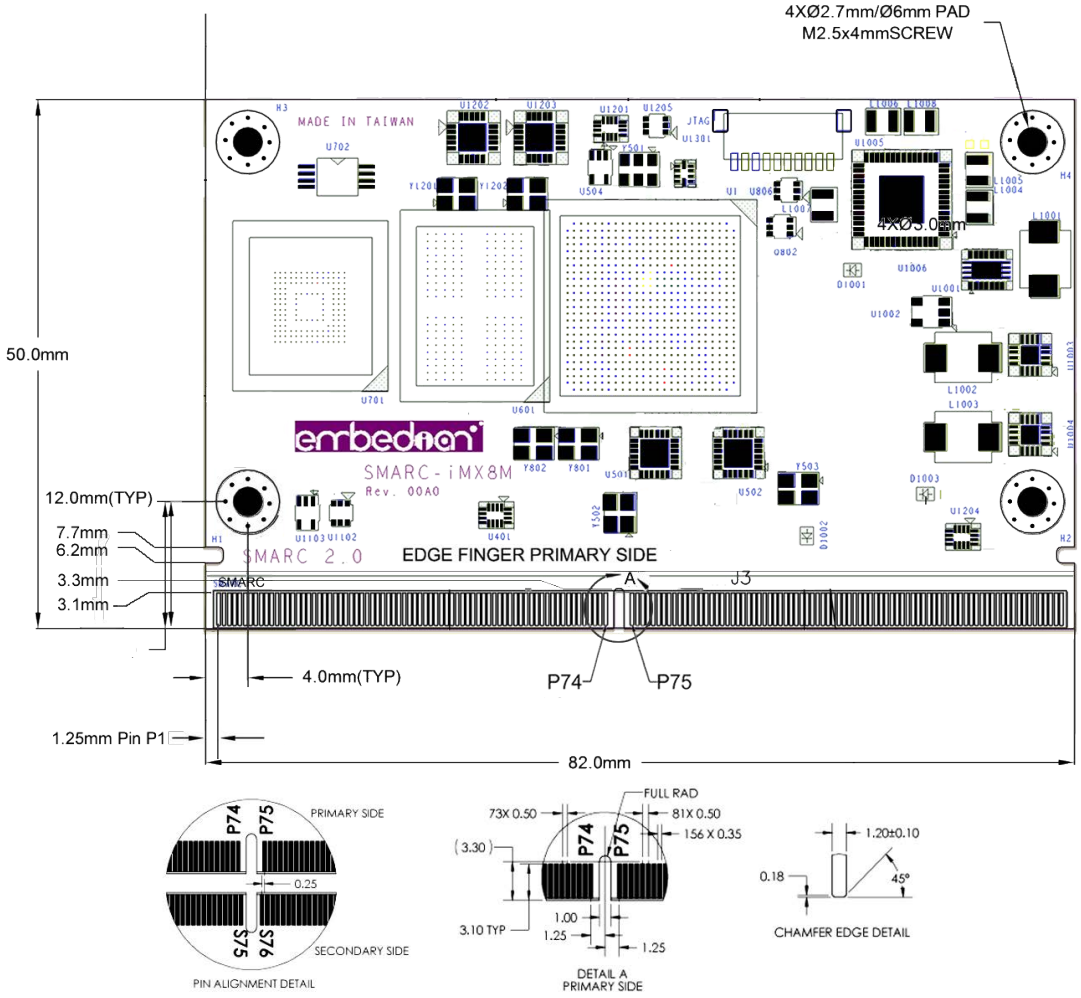


Figure 14. SMARC-iMX8M Mechanical Drawings (Top View)

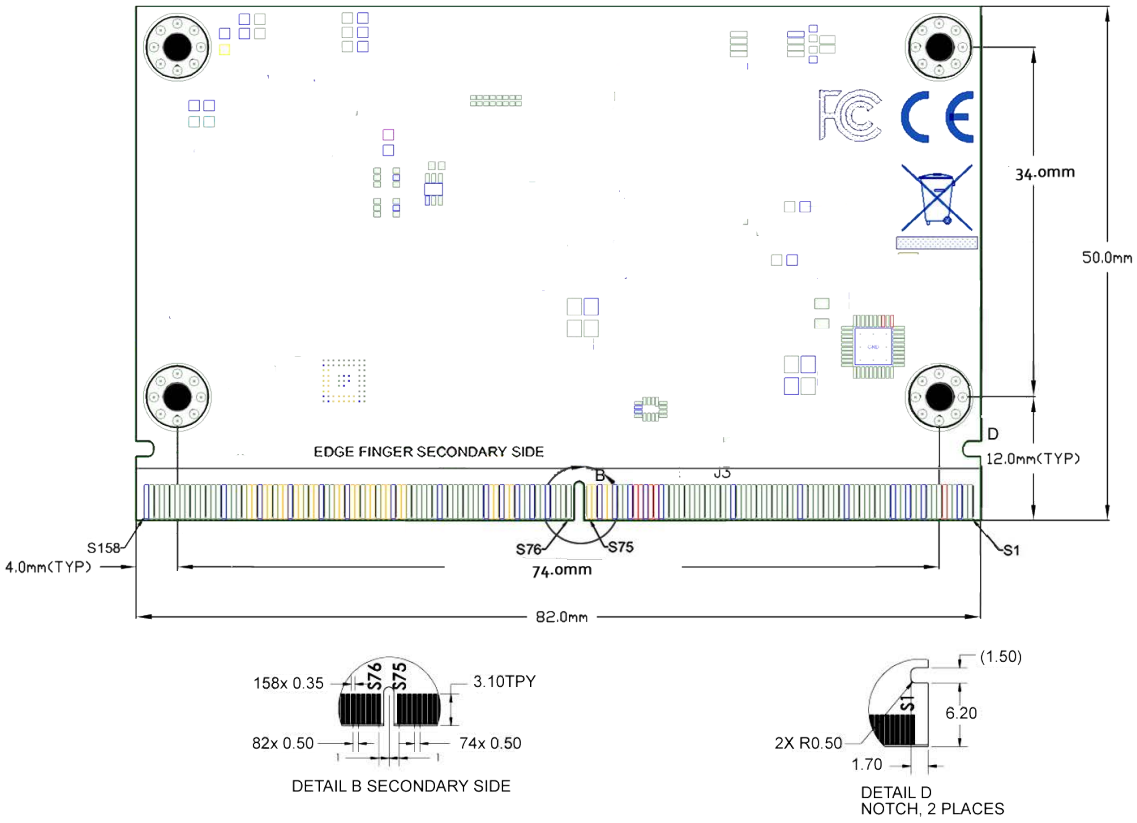


Figure 15. SMARC-iMX8M Mechanical Drawings (Bottom View)

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

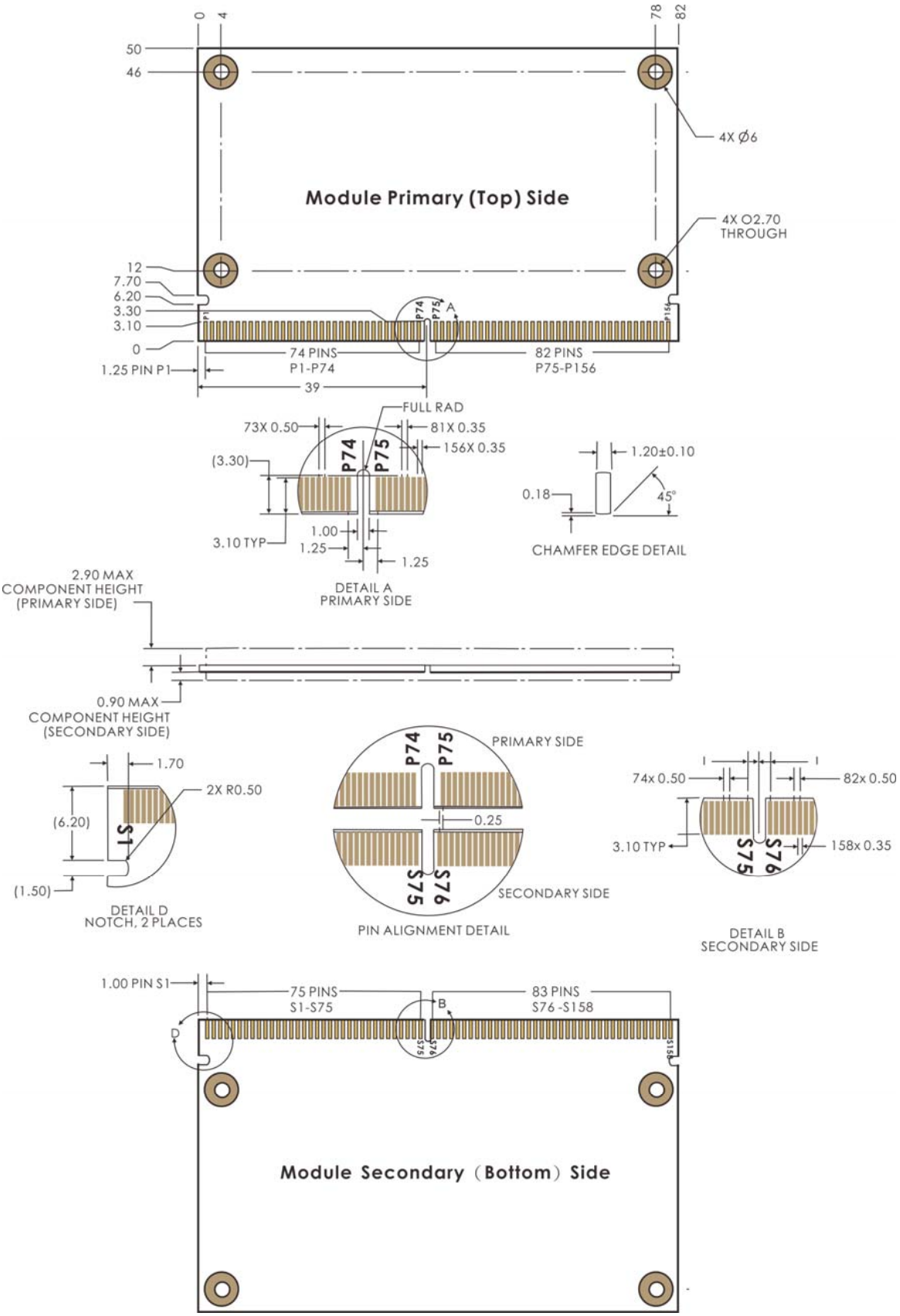


Figure 16: SMARC-iMX8M Module Mechanical Outline

Top side major component (IC and Connector) information is shown in Figure 17: *SMARC-iMX8M* Top side components.

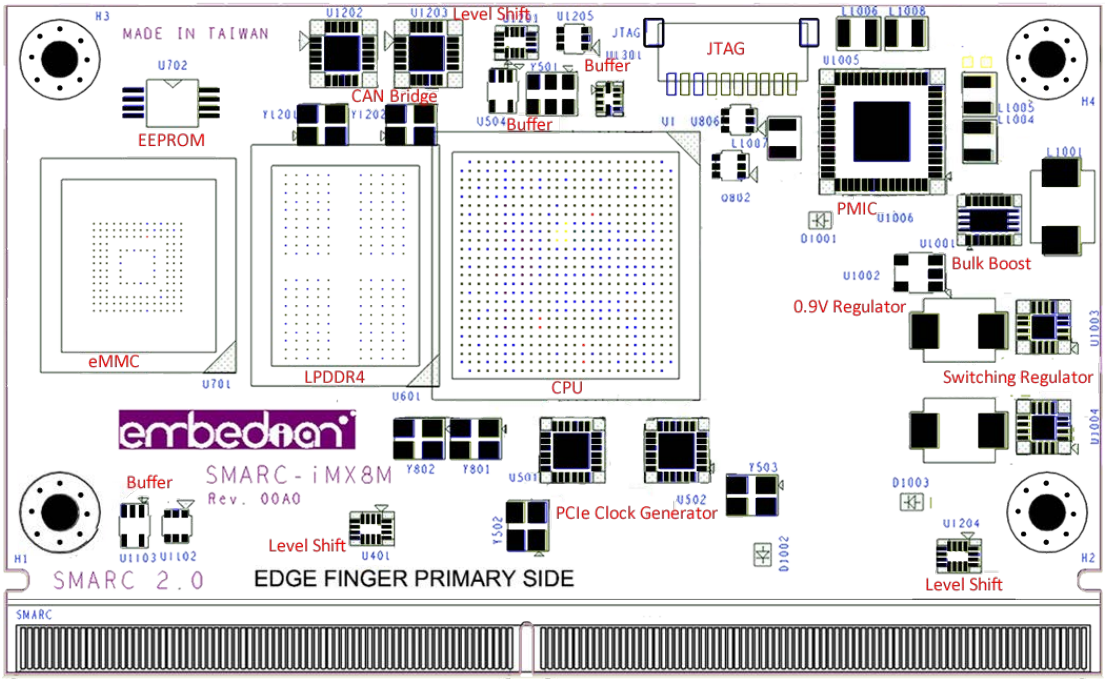


Figure 17. SMARC-iMX8M Top Side Components

Bottom side major component (IC and Connector) information is shown in Figure 18: *SMARC-iMX8M* Bottom side components.

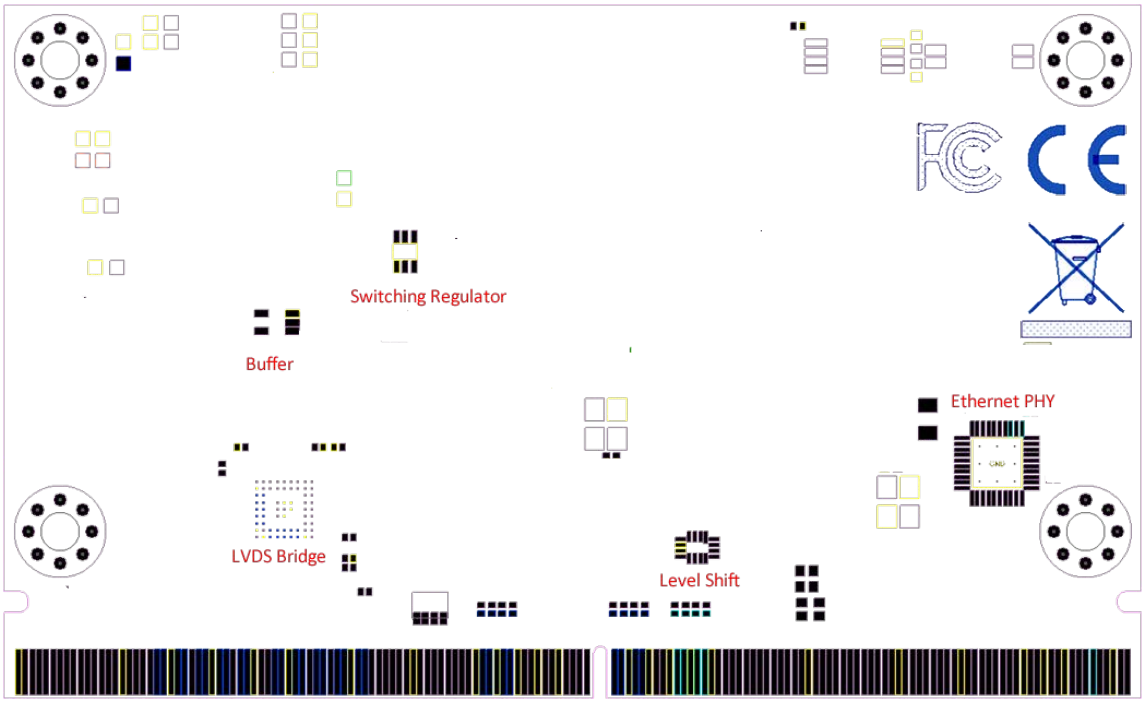


Figure 18. SMARC-iMX8M Bottom Side Components

SMARC-iMX8M height information from Carrier board Top side to tallest Module component is shown in Figure 19: *SMARC-iMX8M* Minimum "Z" Height:

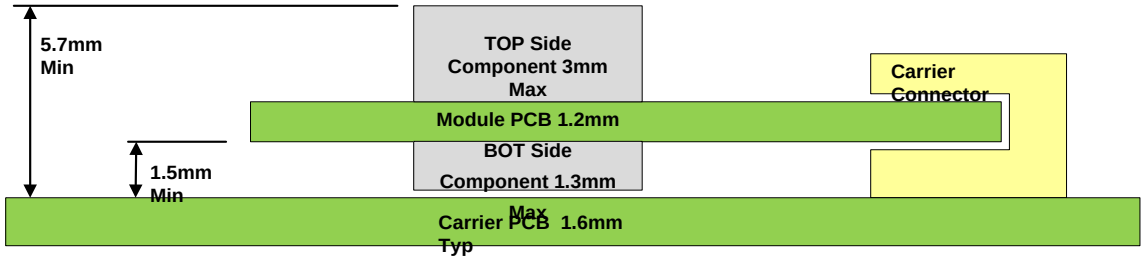


Figure 19. SMARC-iMX8M Minimum "Z" Height

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

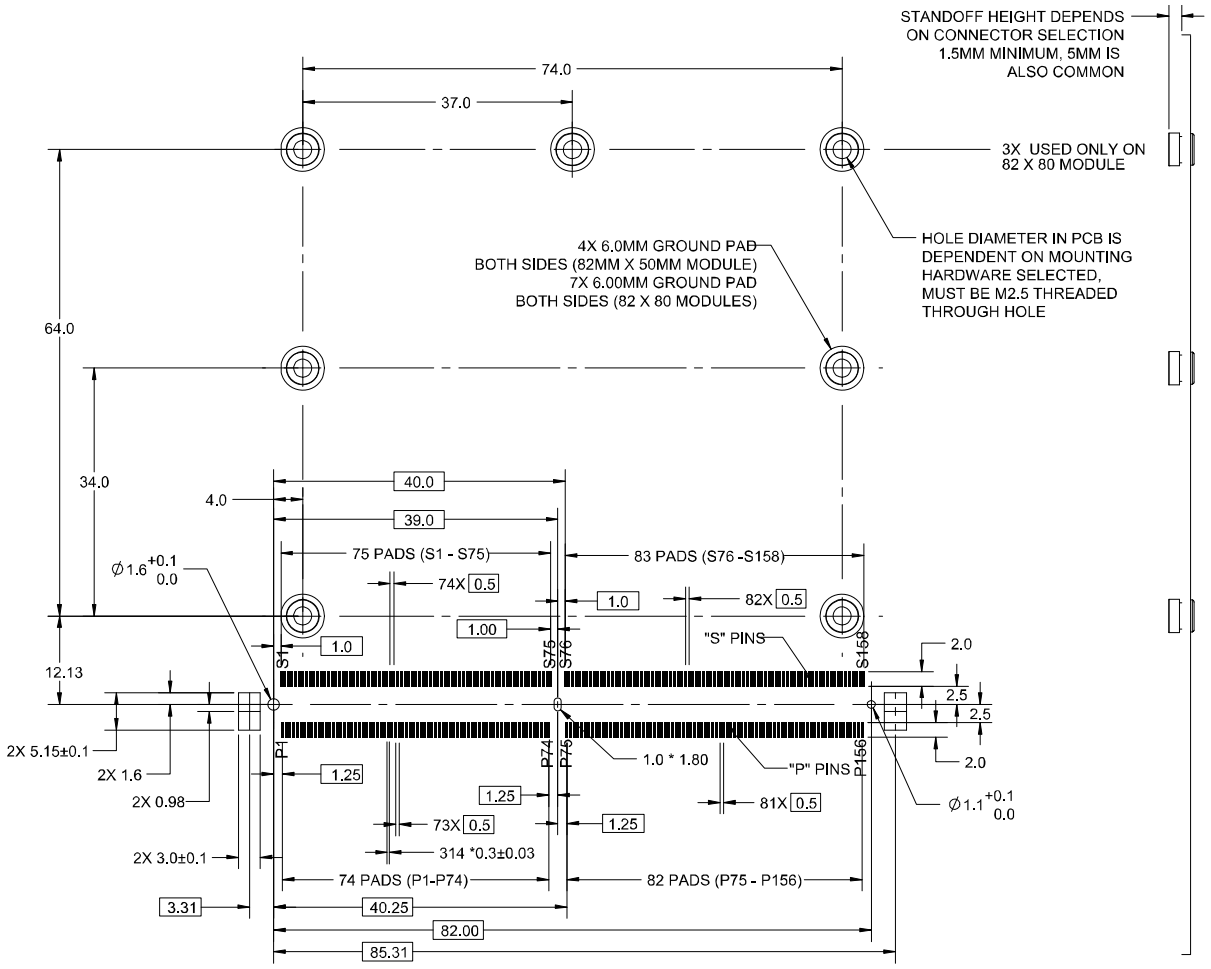


Figure 20: Carrier Board Connector PCB Footprint

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-iMX8M* 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-iMX8M* Mechanical Drawings (Top View)

2.3.7. Carrier Board Standoffs

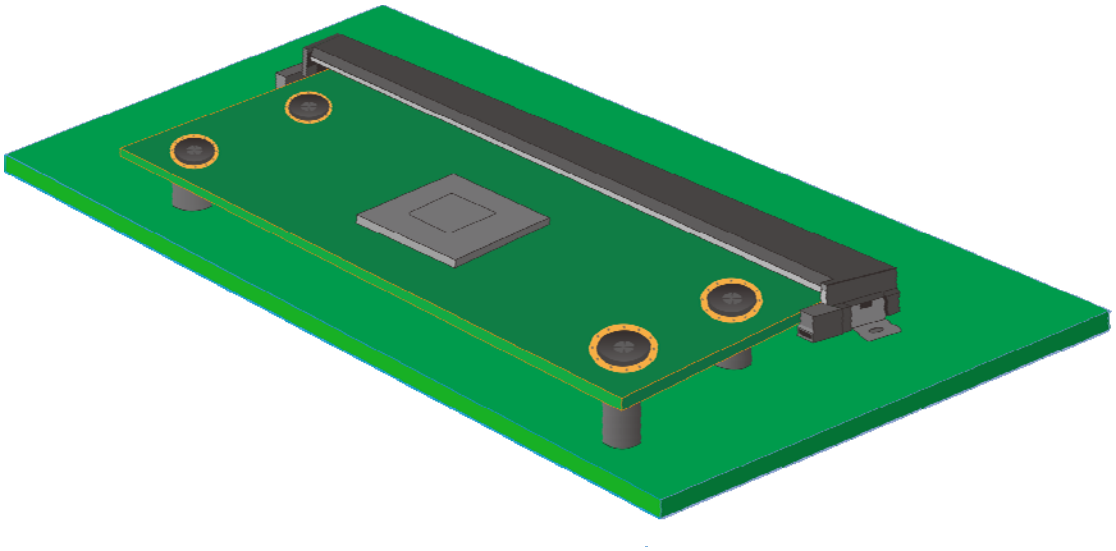


Figure 21: Screw Fixation

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

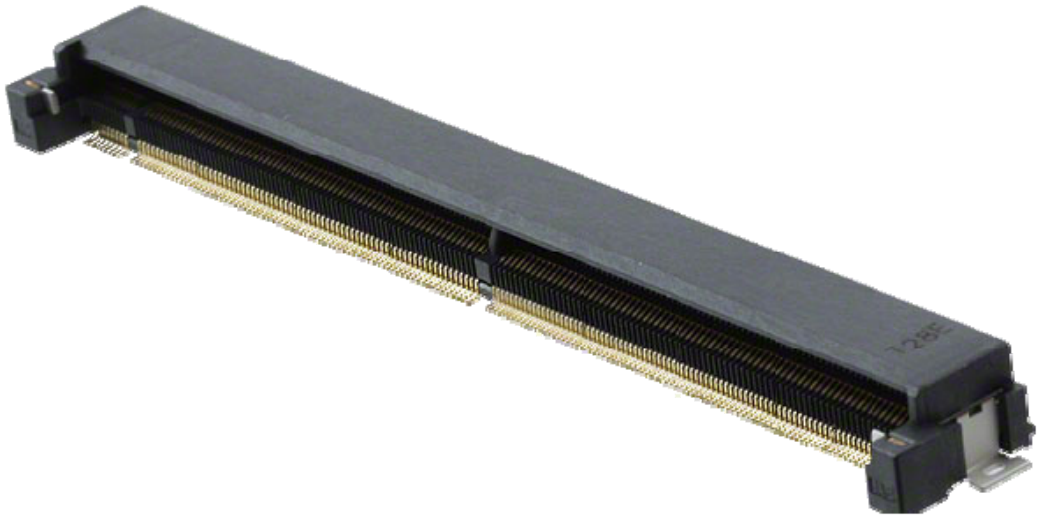


Figure 22: MXM3 Carrier Connector

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	7.8mm	Flash	Std	Black
Foxconn	AS0B821-S78N - *H	5.0mm	7.8mm	Flash	Std	Ivory
Foxconn	AS0B826-S78B - *H	5.0mm	7.8mm	10 u-in	Std	Black
Foxconn	AS0B826-S78N - *H	5.0mm	7.8mm	10 u-in	Std	Ivory
Yamaichi [(1)]	CN113-314-2001	5.0mm	7.8mm	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.

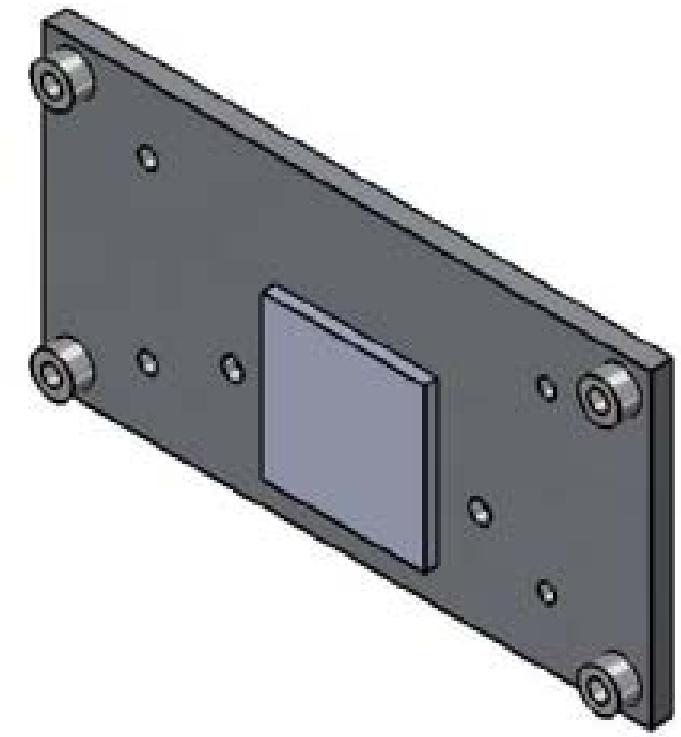


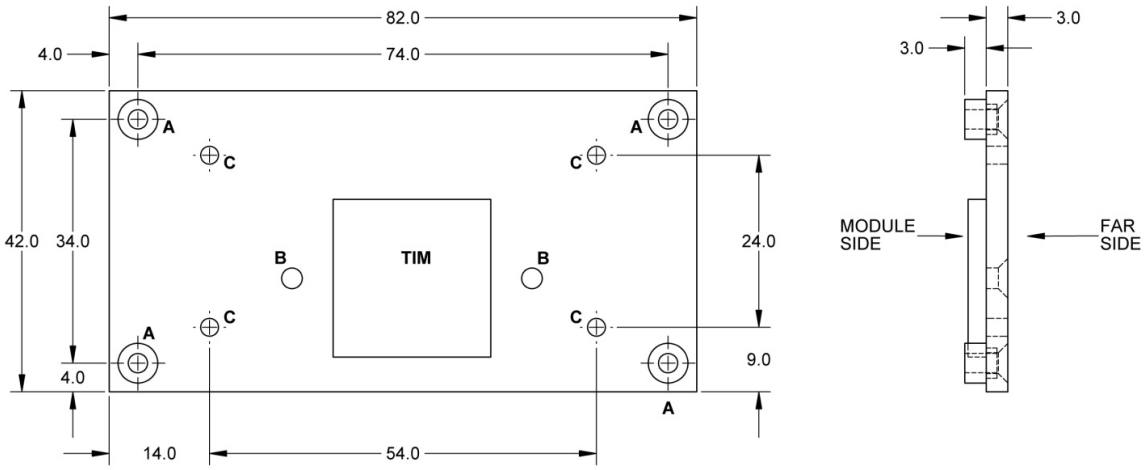
Figure 23: Heat Spreader

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-iMX8M* 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 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-iMX8M* 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-iMX8M* module supports 1.8V (*SMARC* v1.1 compliant) or 3.3V (*SMARC* v1.0 compliant) level I/O voltage depending on the part number that users selected.

2.4.5. MTBF

The *SMARC-iMX8M* 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-iMX8M* module, carrier board is *EVK-STD-CARRIER-S20* with *HDMI* monitor, 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 Morty. 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 Morty

- 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-iMX8M*.

SMARC Part Number	*Desktop Idle*	*100% workload*	*Max. power consumption (Amp/Watts)*
SMARC-iMX8M-D-2G	*TBD*	*TBD*	*TBD*
SMARC-iMX8M-Q-2G	*TBD*	*TBD*	*TBD*
SMARC-iMX8M-L-2G	*TBD*	*TBD*	*TBD*
SMARC-iMX8M-Q-4G	*TBD*	*TBD*	*TBD*

2.5 Environmental Specifications

2.5.1. Operating Temperature

The *SMARC-iMX8M* module operates from 0°C to 80°C air temperature, without a passive heat sink arrangement. Industrial temperature (-40°C ~85°C is also available with different part number *SMARC-iMX8M-X-XX-I*).

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-iMX8M* module is compliant to the *2002/95/EC RoHS* directive and *REACH* directive.

Connector PinOut

This Chapter gives detail pinout of *SMARC-iMX8M* golden finger edge connector.

Section include：

- *SMARC-iMX8M* 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-iMX8M* 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-iMX8M Connector Pin Mapping

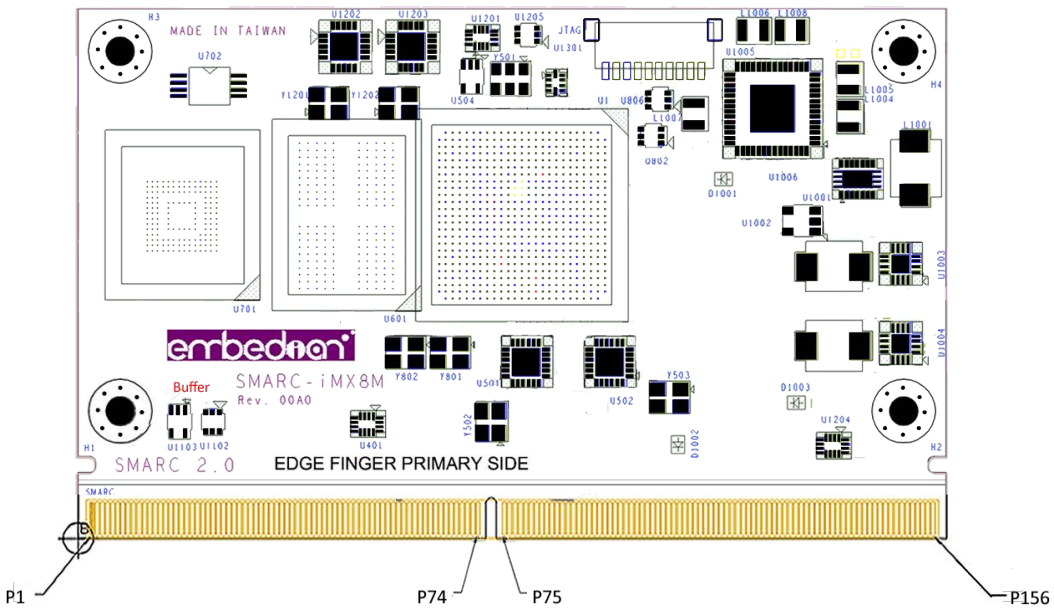


Figure 24: SMARC-iMX8M edge finger primary pins

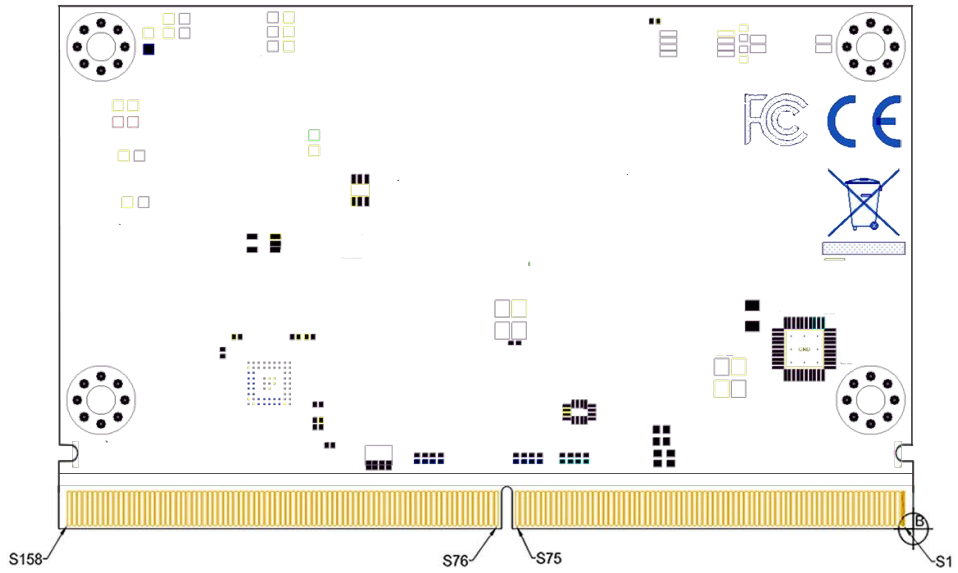


Figure 25: SMARC-iMX8M edge finger secondary pins

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 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* 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 CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P1	SMB_ALERT_1V8#					Not used
P2	GND				P	Ground
P3	CSI1_CK+	B19		MIPI_CSI2_CLK_P	I	CSI1 differential clock inputs
P4	CSI1_CK-	A19		MIPI_CSI2_CLK_N	I	CSI1 differential clock inputs
P5	GBE1_SDP					Not used
P6	GBE0_SDP					Not used
P7	CSI1_RX0+	D20		MIPI_CSI2_D0_P	I	CSI1 differential data inputs 0 (positive)
P8	CSI1_RX0-	C20		MIPI_CSI2_D0_N	I	CSI1 differential data input 0 (negative)
P9	GND				P	Ground
P10	CSI1_RX1+	B20		MIPI_CSI2_D1_P	I	CSI1 differential data input 1 (positive)
P11	CSI1_RX1-	A20		MIPI_CSI2_D1_N	I	CSI1 differential data inputs 1 (negative)
P12	GND				P	Ground
P13	CSI1_RX2+	B21		MIPI_CSI2_D2_P		CSI1 differential data inputs 2 (positive)
P14	CSI1_RX2-	A21		MIPI_CSI2_D2_N		CSI1 differential data inputs 2 (negative)
P15	GND				P	Ground
P16	CSI1_RX3+	D19		MIPI_CSI2_D3_P		CSI1 differential data inputs 3 (positive)
P17	CSI1_RX3-	C19		MIPI_CSI2_D3_N		CSI1 differential data inputs 3 (negative)

P18	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
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
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

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
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 CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P31	SPI0_CS1#	T6	ALT5	GPIO1_IO00__ GPIO1_IO0	O	SPI0 Master Chip Select 1 output.
P32	GND				P	Ground
P33	SDIO_WP	M21	ALT5	SD2_WP__ GPIO2_IO20	I	Write Protect
P34	SDIO_CMD	M22	ALT0	SD2_CMD__ SD2_USDHC2_ CMD	IO	Command Line
P35	SDIO_CD#	L21	ALT5	SD1_CD__ GPIO2_IO12	I	Card Detect
P36	SDIO_CK	L22	ALT0	SD2_CLK__ SD2_USDHC2_ CLK	O	Clock
P37	SDIO_PWR_EN	R22	ALT5	SD2_RESET_B__ GPIO2_IO10	O	SD card power enable
P38	GND				P	Ground
P39	SDIO_D0	N22	ALT0	SD2_DATA0__ SD2_USDHC2_ DATA0	IO	Data path
P40	SDIO_D1	N21	ALT0	SD2_DATA1__ SD2_USDHC2_ DATA1	IO	Data path
P41	SDIO_D2	P22	ALT0	SD2_DATA2__ SD2_USDHC2_ DATA2	IO	Data path
P42	SDIO_D3	P21	ALT0	SD2_DATA3__ SD2_USDHC2_ DATA3	IO	Data path

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P43	SPI0_CS0#	D4	ALT5	ECSPI1_SS0__ GPIO5_IO9	O	SPI0 Master Chip Select 0 output,
P44	SPI0_CK	D5	ALT0	ECSPI1_SCLK__ ECSPI1_SCLK	O	SPI0 Master Clock output
P45	SPI0_DIN	B4	ALT0	ECSPI1_MISO__ ECSPI1_MISO	I	SPI0 Master Data input (input to CPU, output from SPI device)
P46	SPI0_DO	A4	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 CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P54	ESPI1_CS0#	H19	ALT1	NAND_CE0_B__ QSPI_A_SS0_B	O	SPI1 Master Chip Select 0 output
P55	ESPI1_CS1#	G21	ALT1	NAND_CE1_B__ QSPI_A_SS1_B	O	SPI1 Master Chip Select 1 output
P56	ESPI1_CK	G19	ALT1	NAND_ALE__ QSPI_A_SCLK	O	SPI1 Master Clock output
P57	ESPI1_IO_1	J20	ALT1	NAND_DATA01__ QSPI_A_DATA01	I	SPI1 Master Data input (input to CPU, output from SPI device)
P58	ESPI1_IO_0	G20	ALT1	NAND_DATA00__ QSPI_A_DATA00	O	SPI1 Master Data output (output from CPU, input to SPI device)
P59	GND				P	Ground
P60	USB0+	A14		USB1_DP	AIO	Differential USB0 data
P61	USB0-	B14		USB1_DN	AIO	Differential USB0 data
P62	USB0_EN_OC#	L20	ALT5	NAND_DATA04__ GPIO3_IO10	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 CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P63	USB0_VBUS_DET	D14		Turn on USB_OTG_VBUS	I	USB host power detection, when this port is used as a device
P64	USB0_OTG_ID	M7		GPIO1_IO10__ GPIO1_IO10	I	USB OTG ID input, active high
P65	USB1+				IO	Differential USB1 data pair (from USB2514)
P66	USB1-				IO	Differential USB1 data pair (from USB2514)
P67	USB1_EN_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
P68	GND				P	Ground
P69	USB2+				IO	Differential USB2 data pair (from USB2514)
P70	USB2-				IO	Differential USB2 data pair (from USB2514)

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P71	USB2_EN_OC#	L19	ALT5	NAND_DATA06__ GPIO3_IO12	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#	M19	ALT5	NAND_DATA07__ GPIO3_IO13	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 CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P75	PCIE_A_RST#	F21	ALT5	NAND_CE2_B__ GPIO3_IO3	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	RSVD					Not used
P78	RSVD					Not used
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 CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P83	PCIE_A_REFCK+	K25		PCIE1_REF_PAD_CLK_P	O	Differential PCI Express Reference Clock Signals for Lanes A
P84	PCIE_A_REFCK-	K24		PCIE1_REF_PAD_CLK_P	O	Differential PCI Express Reference Clock Signals for Lanes A
P85	GND				P	
P86	PCIE_A_RX+	H25		PCIE1_RXN_P	I	Differential PCIe Link A receive data pair 0
P87	PCIE_A_RX-	H24		PCIE1_RXN_N	I	Differential PCIe Link A receive data pair 0
P88	GND				P	Ground
P89	PCIE_A_TX+	J25		PCIE1_TXN_P	O	Differential PCIe Link A transmit data pair 0
P90	PCIE_A_TX-	J24		PCIE1_TXN_N	O	Differential PCIe Link A transmit data pair 0
P91	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P92	HDMI_D2+ / DP1_LANE0+	N2	N/A	HDMI_TX_P_LN_2	O	TMDS / HDMI data differential pair 2 / DP Data Pair 0+
P93	HDMI_D2- / DP1_LANE0-	N1	N/A	HDMI_TX_M_LN_2	O	TMDS / HDMI data differential pair 2/ DP Data Pair 0-
P94	GND				P	Ground
P95	HDMI_D1+/ DP1_LANE1+	U2	N/A	HDMI_TX_P_LN_1	O	TMDS / HDMI data differential pair 1
P96	HDMI_D1-/ DP1_LANE1-	U1	N/A	HDMI_TX_M_LN_1	O	TMDS / HDMI data differential pair 1
P97	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P98	HDMI_D0+/ DP1_LANE2+	T1	N/A	HDMI_TX_P_ LN_0	O	TMDS / HDMI data differential pair 0
P99	HDMI_D0-/ DP1_LANE2-	T2	N/A	HDMI_TX_M_ LN_0	O	TMDS / HDMI data differential pair 0
P100	GND				P	Ground
P101	HDMI_CK+/ DP1_LANE3+	M1	N/A	HDMI_TX_P_ LN_3	O	HDMI differential clock output pair
P102	HDMI_CK-/ DP1_LANE3-	M2	N/A	HDMI_TX_M_ LN_3	O	HDMI differential clock output pair
P103	GND				P	Ground
P104	HDMI_HPD/ DP1_HDP	W2	N/A	HDMI_HPD	I	HDMI Hot Plug Detect input
P105	HDMI_CTRL_CK/ DP1_AUX+	R3/ v1	ALT4	HDMI_DDC_SCL/ HDMI_AUX_P	IO OD	I2C Clock
P106	HDMI_CTRL_DAT/ DP1_AUX-	P3/ V2	N/A	HDMI_DDC_SDA/ HDMI_AUX_N	IO OD	I2C Data
P107	DP1_AUX_SEL				II	Pulled to GND on Carrier for DP operation in Dual Mode implementations.

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P108	GPIO0 / CAM0_PWR#	K4	ALT5	SAI5_MCLK__ GPIO3_IO25	IO	Camera 0 Power Enable, active low output
P109	GPIO1 / CAM1_PWR#	N4	ALT5	SAI5_RXFS__ GPIO3_IO19	IO	Camera 1 Power Enable, active low output
P110	GPIO2 / CAM0_RST#	L5	ALT5	SAI5_RXC__ GPIO3_IO20	IO	Camera 0 Reset, active low output
P111	GPIO3 / CAM1_RST#	M5	ALT5	SAI5_RXD0__ GPIO3_IO21	IO	Camera 1 Reset, active low output
P112	GPIO4 / HDA_RST#	L4	ALT5	SAI5_RXD1__ GPIO3_IO22	IO	HD Audio Reset, active low output
P113	GPIO5 / PWM_OUT	F6	ALT5	SPDIF_TX__ GPIO5_IO3	IO	PWM output
P114	GPIO6 / TACHIN	G6	ALT5	SPDIF_RX__ GPIO5_IO4	IO	Tachometer input (used with the GPIO5 PWM)
P115	GPIO7 / PCAM_FLD	M4	ALT5	SAI5_RXD2__ GPIO3_IO23	IO	PCAM_FLD (Field) signal input
P116	GPIO8 / CAN0_ERR#	K5	ALT5	SAI5_RXD3__ GPIO3_IO24	IO	CAN0 Error signal, active low input
P117	GPIO9 / CAN1_ERR#	E1	ALT5	SAI1_TXC__ GPIO4_IO11	IO	CAN1 Error signal, active low input
P118	GPIO10	H1	ALT5	SAI1_TXFS__ GPIO4_IO10	IO	
P119	GPIO11	A3	ALT5	SAI1_MCLK__ GPIO4_IO20	IO	
P120	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P121	I2C_PM_CK	E7	ALT0	I2C1_SCL__ I2C1_SCL	IO OD	Power management I2C bus clock
P122	I2C_PM_DAT	E8	ALT0	I2C1_SDA__ I2C1_SDA	IO OD	Power management I2C bus data
P123	BOOT_SEL0#	N7	ALT0	GPIO1_IO08__ GPIO1_IO8	I	SYSBOOT and Line De-multiplexer Logic Pulled up on Module. Driven by OD part on Carrier.
P124	BOOT_SEL1#	P7	ALT0	GPIO1_IO05__ GPIO1_IO5	I	SYSBOOT and Line De-multiplexer Logic Pulled up on Module. Driven by OD part on Carrier.
P125	BOOT_SEL2#	N5	ALT0	GPIO1_IO06__ GPIO1_IO6	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 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	E5	ALT1	ECSPI2_MOSI__ UART4_DCE_TX	O	Asynchronous serial port data out
P130	SER0_RX	C5	ALT1	UART4_RXD__ UART4_DCE_RX	I	Asynchronous serial port data in
P131	SER0_RTS#	B5	ALT1	ECSPI2_MISO__ UART4_DCE_ CTS_B	O	Request to Send handshake line for SER0
P132	SER0_CTS#	A5	ALT1	ECSPI2_SS0__ UART4_DCE_ RTS_B	I	Clear to Send handshake line for SER0
P133	GND				P	Ground
P134	SER1_TX	B7	ALT0	UART3_TXD__ UART3_DCE_TX	O	Asynchronous serial port data out
P135	SER1_RX	A6	ALT0	UART3_RXD__ UART3_DCE_RX	I	Asynchronous serial port data in

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
P136	SER2_TX	D6	ALT0	UART2_TXD__ UART2_DCE_TX		Asynchronous serial port data out
P137	SER2_RX	B6	ALT0	UART2_RXD__ UART2_DCE_RX		Asynchronous serial port data in
P138	SER2_RTS#	C6	ALT1	UART4_RXD__ UART2_DCE_ CTS_B		Request to Send handshake line for SER2
P139	SER2_CTS#	D7	ALT1	UART4_TXD__ UART2_DCE_ RTS_B		Clear to Send handshake line for SER2
P140	SER3_TX	A7	ALT0	UART1_TXD__ UART1_DCE_TX	O	Asynchronous serial port data out
P141	SER3_RX	C7	ALT0	UART1_RXD__ UART1_DCE_RX	I	Asynchronous serial port data in
P142	GND				P	Ground
P143	CAN0_TX				O	CAN0 Transmit output from MCP2515T
P144	CAN0_RX				I	CAN0 Receive input from MCP2515T
P145	CAN1_TX				O	CAN1 Transmit output from MCP2515T
P146	CAN1_RX				I	CAN1 Receive input from MCP2515T

SMARC Edge Finger		NXP i.MX8M 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 CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S1	CS1_TX+/ I2C_CAM1_CK	F8	ALT0	I2C4_SCL__ I2C4_SCL	IO OD	Camera1 I2C bus clock
S2	CS1_TX-/ I2C_CAM1_DAT	F9	ALT0	I2C4_SDA__ I2C4_SDA	IO OD	Camera1 I2C bus data
S3	GND				P	Ground
S4	RSVD					Not used
S5	CSI0_TX+ / I2C_CAM0_CK	G7	ALT0	I2C2_SCL__ I2C2_SCL	IO OD	Camera0 I2C bus clock
S6	CAM_MCK	K7	ALT6	GPIO1_IO14__ CCMSRCGPCMIX _CLKO1	O	Master clock output for CSI camera support
S7	CSI0_TX- / I2C_CAM0_DAT	F7	ALT0	I2C2_SDA__ I2C2_SDA	IO OD	Camera0 I2C bus data
S8	CSI0_CK+	B22	ALT0	MIPI_CSI1_CLK_ P	I	CSI0 differential clock inputs
S9	CSI0_CK-	A22	ALT0	MIPI_CSI1_CLK_ N	I	CSI0 differential clock inputs
S10	GND				P	Ground
S11	CSI0_RX0+	B23	ALT0	MIPI_CSI1_D0_ P	I	CS0 differential data inputs 0 (positive)
S12	CSI0_RX0-	A23	ALT0	MIPI_CSI1_D0_ N	I	CSI0 differential data input 0 (negative)
S13	GND				P	Ground
S14	CSI0_RX1+	D22	ALT0	MIPI_CSI1_D1_ P	I	CSI0 differential data input 1 (positive)
S15	CSI0_RX1-	C22	ALT0	MIPI_CSI1_D1_ N	I	CSI0 differential data inputs 1 (negative)
S16	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S17	GbE1_MDI0+					Not used
S18	GbE1_MDI0-					Not used
S19	GbE1_LINK100#					Not used
S20	GbE1_MDI1+					Not used
S21	GbE1_MDI1-					Not used
S22	GbE1_LINK1000#					Not used
S23	GbE1_MDI2+					Not used
S24	GbE1_MDI2-					Not used
S25	GND				P	Ground
S26	GbE1_MDI3+					Not used
S27	GbE1_MDI3-					Not used
S28	GbE1_CTREF					Not used
S29	PCIE_D_TX+					Not used
S30	PCIE_D_TX-					Not used
S31	GBE1_LINK_ACK#					Not used
S32	PCIE_D_RX+					Not used
S33	PCIE_D_RX-					Not used
S34	GND					Ground

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S35	USB4+					Differential USB4 data pair (from USB2514)
S36	USB4-					Differential USB4 data pair (from USB2514)
S37	USB3_VBUS_DET					Not used
S38	AUDIO_MCK	H5	ALT0	SAI2_MCLK__ SAI2_MCLK	O	Master clock output to Audio codecs
S39	I2S0_LRCK	H4	ALT0	SAI2_TXFS__ SAI2_TX_SYNC	IO	Left& Right audio synchronization clock
S40	I2S0_SDOUT	G5	ALT0	SAI2_TXD0__ SAI2_TX_DATA0	O	Digital audio Output
S41	I2S0_SDIN	H6	ALT0	SAI2_RXD0__ SAI2_RX_DATA0	I	Digital audio Input
S42	I2S0_CK	J5	ALT0	SAI2_TXC__ 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 CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S48	I2C_GP_CK	G8	ALT0	I2C3_SCL__ I2C3_SCL	IO OD	Port1 of TCA9546 General purpose I2C bus clock
S49	I2C_GP_DAT	E9	ALT0	I2C3_SDA__ I2C3_SDA	IO OD	Port1 of TCA9546 General purpose I2C bus clock
S50	HDA_SYNC/ I2S2_LRCK	G3	ALT0	SAI3_TXFS__ SAI3_TX_SYNC	IO	Left& Right audio synchronization clock
S51	HDA_SDO/ I2S2_SDOUT	C3	ALT0	SAI3_TXD__ SAI3_TX_DATA0	O	Digital audio Output
S52	HDA_SDI/ I2S2_SDIN	F3	ALT0	SAI3_RXD__ SAI3_RX_DATA0	I	Digital audio Input
S53	HAD_CK/ I2S2_CK	C4	ALT0	SAI3_TXC__ SAI3_TX_BCLK	IO	Digital audio clock
S54	SATA_ACT#					Not used
S55	USB5_EN_OC#					Not used
S56	ESPI_IO_2	G21	ALT1	NAND_DATA02_ _ QSPI_A_DATA02	IO	QSPI Master Data 2
S57	ESPI_IO_3	J21	ALT1	NAND_DATA03_ _ QSPI_A_DATA03	IO	QSPI Master Data 3
S58	ESPI_RESET#	M20	ALT1	NAND_DQS__ QSPI_A_DQS	O	Reset the eSPI interface for both master and slaves.
S59	USB5+					Not used
S60	USB5-					Not used
S61	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S62	USB3_SSTX+	A9		USB2_TX_P	AO	USB3 data transmit signal differential pairs positive
S63	USB3_SSTX-	B9		USB2_TX_N	AO	USB3 data transmit signal differential pairs negative
S64	GND				P	Ground
S65	USB3_SSRX+	A8		USB2_RX_P	AI	USB3 data receive signal differential pairs positive
S66	USB3_SSRX-	B8		USB2_RX_N	AI	USB3 data receive signal differential pairs negative
S67	GND				P	Ground
S68	USB3+					Differential USB3 data pair (from USB2514)
S69	USB3-					Differential USB3 data pair (from USB2514)
S70	GND				P	Ground
S71	USB2_SSTX+	A13		USB1_TX_P	AO	USB2 data transmit signal differential pairs positive
S72	USB2_SSTX--	B13		USB1_TX_N	AO	USB2 data transmit signal differential pairs negative
S73	GND				P	Ground
S74	USB2_SSRX+	A12		USB1_TX_P	AI	USB2 data receive signal differential pairs positive
S75	USB2_SSRX-	B12		USB1_TX_N	AI	USB2 receive signal differential pairs negative

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S76	PCIE_B_RST#	H20	ALT5	NAND_CE3_B__ GPIO3_IO4	O	Reset Signal for external devices.
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
S84	PCIE_B_REFCK+	F25		PCIE2_REF_PAD _CLK_P	O	Differential PCI Express Reference Clock Signals for Lanes A
S85	PCIE_B_REFCK-	F24		PCIE2_REF_PAD _CLK_P	O	Differential PCI Express Reference Clock Signals for Lanes A
S86	GND				P	Ground
S87	PCIE_B_RX+	D25		PCIE2_RXN_P	I	Differential PCIe Link B receive data pair 0
S88	PCIE_B_RX-	D24		PCIE2_RXN_N	I	Differential PCIe Link B receive data pair 0
S89	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S90	PCIE_B_TX+	E25		PCIE2_TXN_P	O	Differential PCIe Link A transmit data pair 0
S91	PCIE_B_TX-	E24		PCIE2_TXN_N	O	Differential PCIe Link A transmit data pair 0
S92	GND				P	Ground
S93	DP0_LAN0+					Not used
S94	DP0_LAN0-					Not used
S95	DP0_AUX_SEL					Not used
S96	DP0_LANE1+					Not used
S97	DP0_LANE1-					Not used
S98	DP0_HPD					Not used
S99	DP0_LANE2+					Not used
S100	DP0_LANE2-					Not used
S101	GND				P	Ground
S102	DP0_LANE3+					Not used
S103	DP0_LANE3-					Not used
S104	USB3_OTG_ID					Not used
S105	DP0_AUX+					Not used
S106	DP0_AUX-					Not used
S107	LCD1_BKLT_EN					Not used

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S108	LVDS1_CK+ / eDP1_AUX+ / DSI1_CLK+				O	LVDS1 LCD differential clock pairs
S109	LVDS1_CK- / eDP1_AUX- / DSI1_CLK-				O	LVDS1 LCD differential clock pairs
S110	GND				P	Ground
S111	LVDS1_0+ / eDP1_TX0+ / DSI1_D0+				AIO	LVDS1 LCD data channel differential pairs 1
S112	LVDS1_0- / eDP1_TX0- / DSI1_D0-				AIO	LVDS1 LCD data channel differential pairs 1
S113	eDP1_HPD					Not used
S114	LVDS1_1+ / eDP1_TX1+ / DSI1_D1+				AIO	LVDS1 LCD data channel differential pairs 2
S115	LVDS1_1- / eDP1_TX1- / DSI1_D1-				AIO	LVDS1 LCD data channel differential pairs 2
S116	LCD1_VDD_EN					Not used
S117	LVDS1_2+ / eDP1_TX2+ / DSI1_D2+				AIO	LVDS1 LCD data channel differential pairs 3
S118	LVDS1_2- / eDP1_TX2- / DSI1_D2-				AIO	LVDS1 LCD data channel differential pairs 3
S119	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S120	LVDS1_3+ / eDP1_TX3+ / DSI1_D3+				AIO	LVDS1 LCD data channel differential pairs 4
S121	LVDS1_3- / eDP1_TX3- / DSI1_D3-				AIO	LVDS1 LCD data channel differential pairs 4
S122	LCD1_BKLT_ PWM					Not used
S123	RSVD					Not used
S124	GND				P	Ground
S125	LVDS0_0+ / eDP0_TX0+ / DSI0_D0+				AIO	LVDS0 LCD data channel differential pairs 1
S126	LVDS0_0- / eDP0_TX0- / DSI0_D0-				AIO	LVDS0 LCD data channel differential pairs 1
S127	LCD_BKLT_EN	L1	ALT5	SAI1_RXFS__ GPIO4_IO0	O	High enables panel backlight
S128	LVDS0_1+ / eDP0_TX1+ / DSI0_D1+				AIO	LVDS0 LCD data channel differential pairs 2
S129	LVDS0_1- / eDP0_TX1- / DSI0_D1-				AIO	LVDS0 LCD data channel differential pairs 2
S130	GND				P	Ground

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S131	LVDS0_2+ / eDP0_TX2+ / DSI0_D2+				AIO	LVDS0 LCD data channel differential pairs 3
S132	LVDS0_2- / eDP0_TX2- / DSI0_D2-				AIO	LVDS0 LCD data channel differential pairs 3
S133	LCD_VDD_EN	K1	ALT5	SAI1_RXC__ GOIO4_IO1	O	High enables panel VDD
S134	LVDS0_CK+ / eDP0_AUX+ / DSI0_CLK+				O	LVDS0 LCD differential clock pairs
S135	LVDS0_CK- / eDP0_AUX- / DSI0_CLK-				O	LVDS0 LCD differential clock pairs
S136	GND				P	Ground
S137	LVDS0_3+ / eDP0_TX3+ / DSI0_D3+				AIO	LVDS0 LCD data channel differential pairs 4
S138	LVDS0_3- / eDP0_TX3- / DSI0_D3-				AIO	LVDS0 LCD data channel differential pairs 4

SMARC Edge Finger		NXP i.MX8M CPU			Type	Description
Pin#	Pin Name	Ball	Mode	Signal Name		
S139	I2C_LCD_CK	G7	ALT0	I2C2_SCL__ I2C2_SCL	IO OD	LCD display I2C bus clock
S140	I2C_LCD_DAT	F7	ALT0	I2C2_SDA__ I2C2_SDA	IO OD	LCD display I2C bus clock
S141	LCD_BKLT_PWM	E6	ALT1	SPDIF_EXT_CLK__ PWM1_OUT	O	Display backlight PWM control
S142	RSVD					Not used
S143	GND				P	Ground
S144	eDP0_HPD					Not used
S145	WDT_TIME_OUT#	J6	ALT5	GPIO1_IO15__ CCMSRCGPCMIX_ CLKO2	O	Watchdog-Timer Output
S146	PCIE_WAKE#	H21	ALT5	NAND_CLE__ GPIO3_IO5	I	PCI Express Wake Event: Sideband wake signal asserted by components requesting wakeup.

SMARC Edge Finger		NXP i.MX8M 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#	M6	ALT0	GPIO1_IO09__ GPIO1_IO9	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#	L7	ALT0	GPIO1_IO12__ GPIO1_IO12	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 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#	J22	ALT5	NAND_DATA05__ GPIO3_IO11	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#	H3	ALT0	SAI2_RXC__ GPIO4_IO22	I	Held low by Carrier if DC input for battery charger is present.
S153	CARRIER_STBY#	D3	ALT5	SAI3_MCLK__ GPIO5_IO2	O	The Module shall drive this signal low when the system is in a standby power state

SMARC Edge Finger		NXP i.MX8M 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#	J4	ALT0	SAI2_RXFS__ GPIO4_IO21	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

Chapter 4

Power Control Signals between SMARC Module and Carrier

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

Section include：

- *SMARC-iMX8M* 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-iMX8M 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-iMX8M Module Power

4.1.1. Input Voltage / Main Power Rail

The allowable Module DC input voltage range for *SMARC-iMX8M* 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-iMX8M* 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-iMX8M* 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-iMX8M* 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

SMARC 1.0 specification defines the I/O voltage to be 1.8V or 3.3V or both. The 3.3V *VDD_IO* is depreciated from *SMARC* 1.1 specification.

SMARC-iMX8M 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.

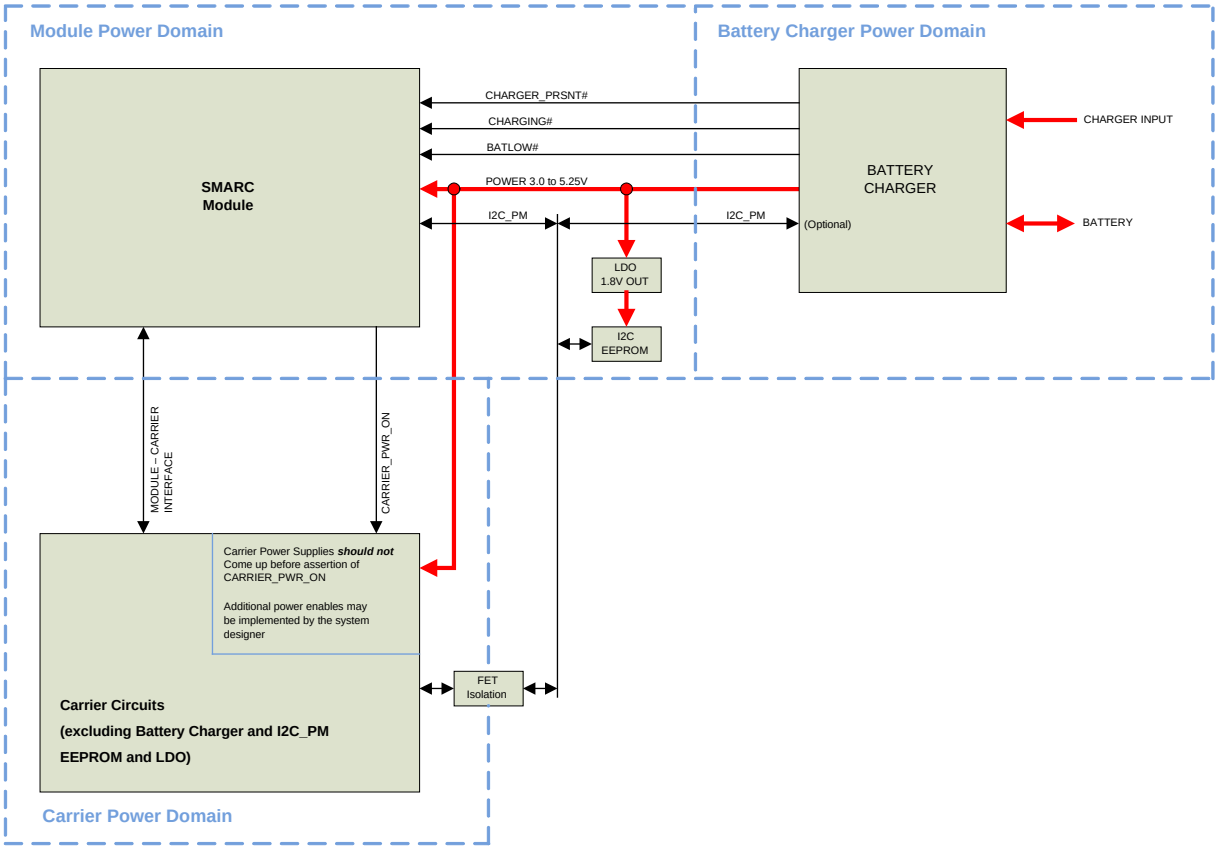


Figure 26 System Power Domains

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#)

i.MX8M processor does not support to boot up from SPI NOR flash. *SMARC-iMX8M* 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* 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.

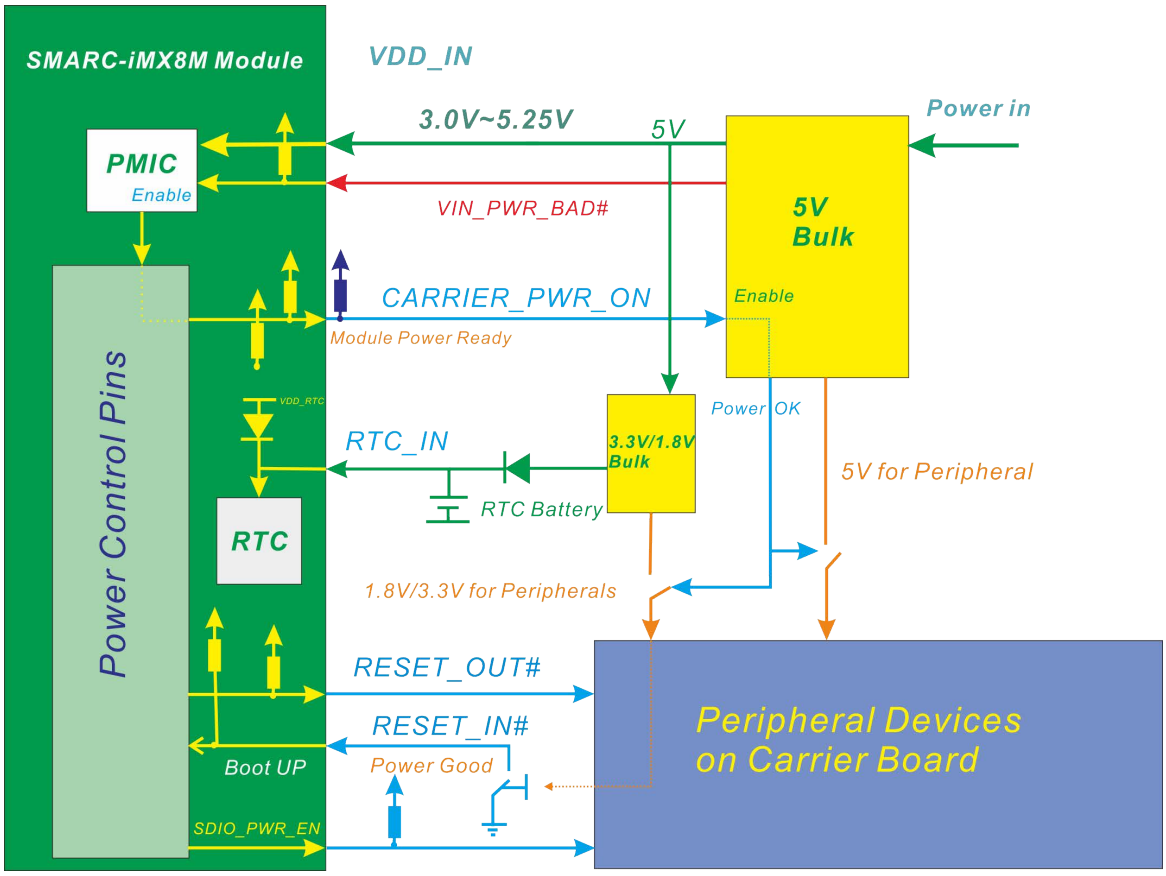


Figure 27: Power Block Diagram

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-iMX8M* 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-iMX8M* 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-iMX8M* 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.

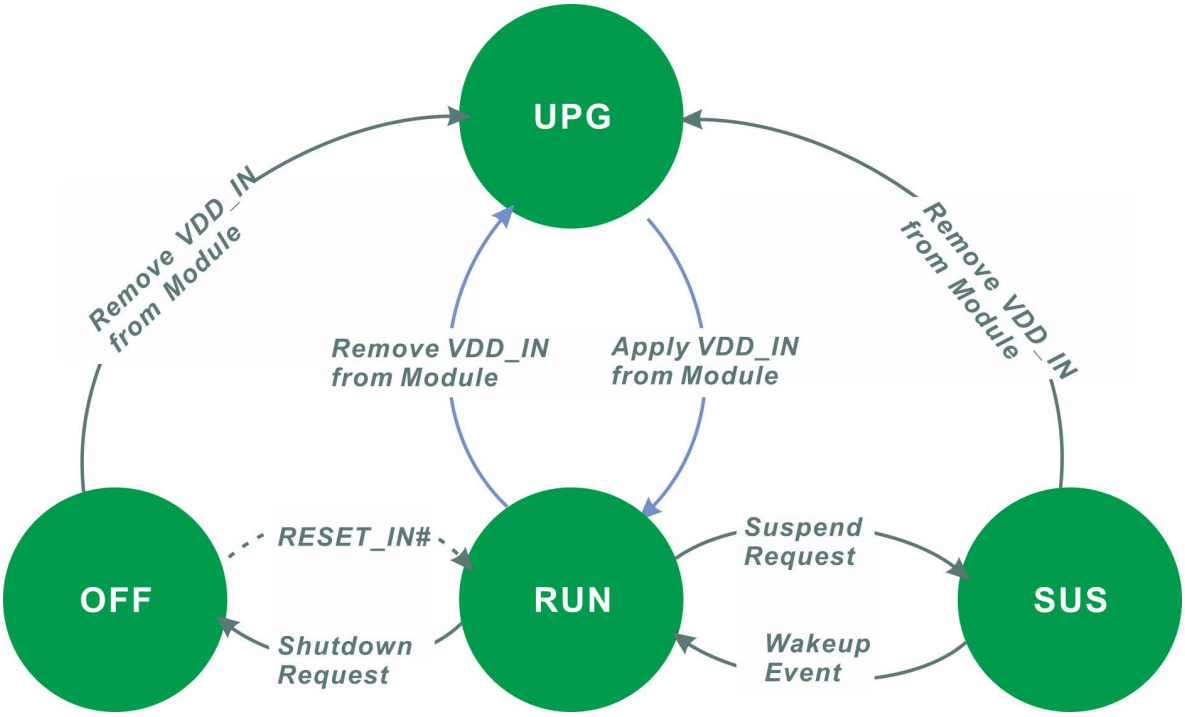


Figure 28: Power States and Transitions

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-iMX8M* 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-iMX8M* 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.

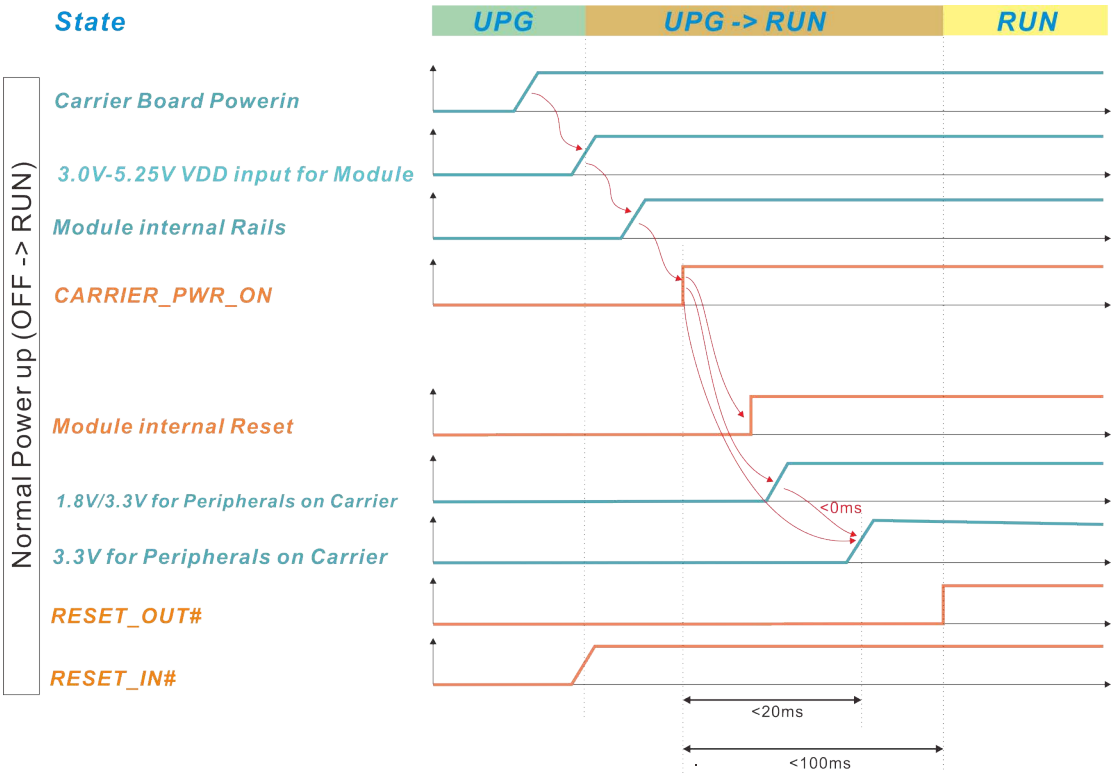


Figure 29: Power-Up Sequence

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).

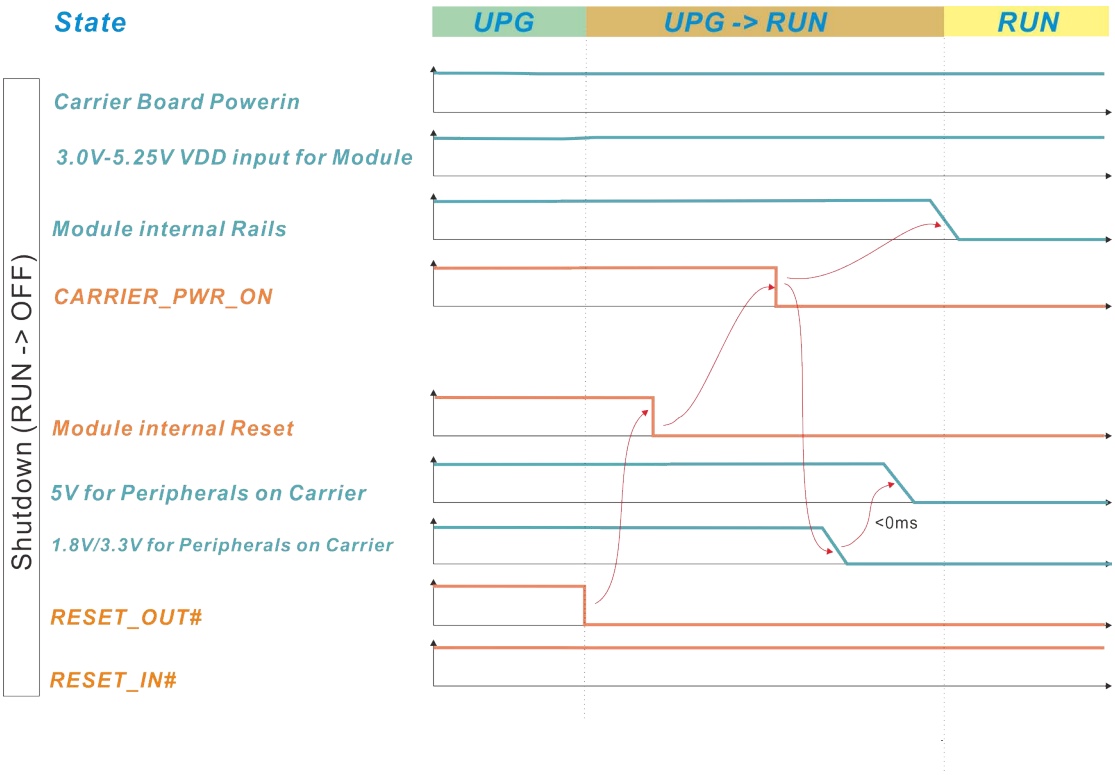


Figure 30: Shutdown Sequence

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.

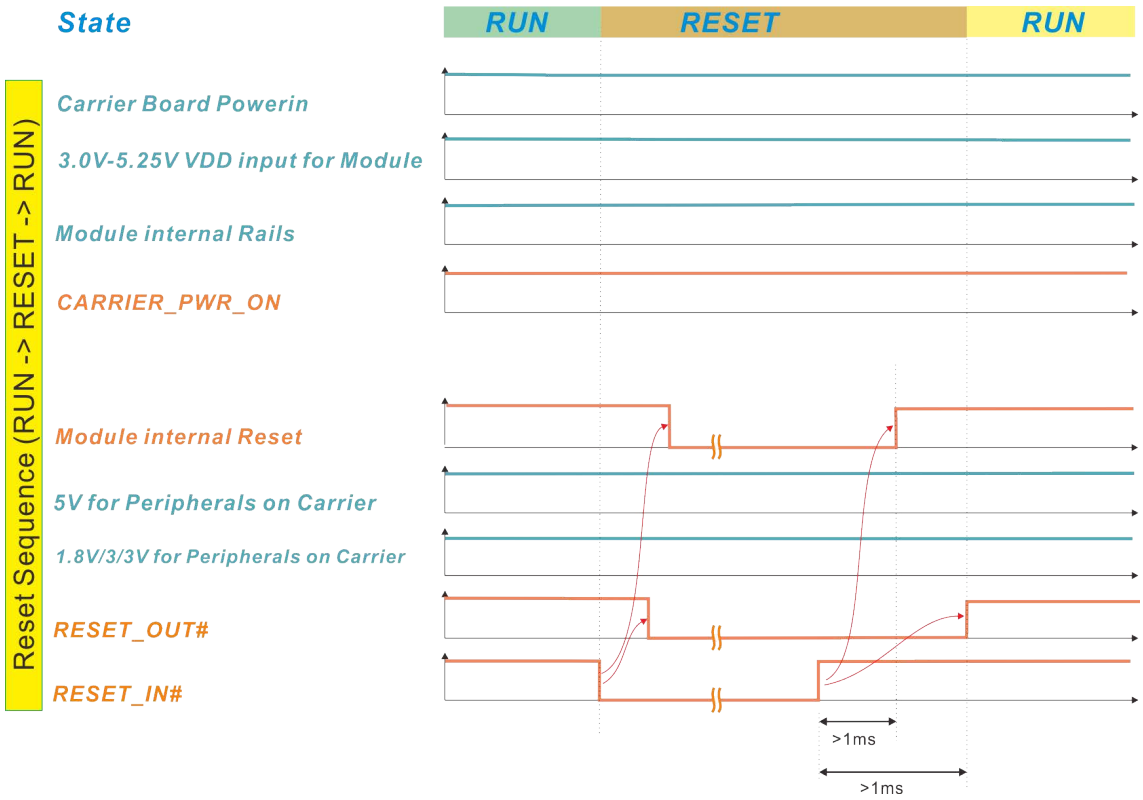


Figure 31: Reset Sequence

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:B]_TX+	0.2uF Capacitor		
PCIE_[A:B]_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)*		
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)*		

Module Signal Group Name	Carrier Series Termination	Carrier Parallel Termination	Notes
DP1_AUX_SEL		*Carrier DP1_AUX_SEL should be connected to pin 13 of the DisplayPort connector to enable a dual-mode DisplayPort interface.*	
DP1_LANE[0:3]+ DP1_LANE[0:3]-		*DC blocking capacitors shall be placed on the Carrier for the DP1_LANE[0:3] signals.*	
DP1_HPD		*The carrier shall include a blocking FET on DP1_HPD to prevent back-drive current from damaging the module.*	

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. *i.MX8M* processor does not support boot up from SPI flash. If *TEST#* is not shunt cross to GND, the first stage of bootloader on *SMARC-iMX8M* 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-iMX8M* will boot up from off-module *SD* card. This is a back door to restore/upgrade the firmware in on-module *eMMC*.