UM12018

FRDM-MCXN947 Board User Manual

Rev. 1 — 20 January 2024 **User manual**

Document information

Information	Content
Keywords	FRDM-MCXN947, UM12018, MCXN947
Abstract	The NXP FRDM-MCXN947 board is a low-cost design and evaluation board based on MCXN947 device. This document describes the hardware for the FRDM-MCXN947 board.

1 FRDM-MCXN947 overview

The NXP FRDM-MCXN947 board is a low-cost design and evaluation board based on the MCXN947 device.

The MCXN947 device integrates a dual Arm Cortex-M33 microcontroller and a neural processing unit (NPU) into a single package. NXP supports the MCXN947 device with tools and software that include hardware evaluation boards, software development IDE, example applications, and drivers.

The FRDM-MCXN947 board consists of one MCXN947 device with a 64 Mbit external serial flash (provided by Winbond). The board also features P3T1755DP I3C temperature sensor, TJA1057GTK/3Z CAN PHY, Ethernet PHY, SDHC circuit (card slot is DNP), RGB LED, touch pad, high-speed USB circuit, push buttons, and MCU-Link debug probe circuit. The board is compatible with the Arduino shield modules, Pmod boards, and mikroBUS. The board also supports a camera module and NXP lost-cost LCD module PAR-LCD-S035.

The onboard MCU-Link debug probe is based on the LPC55S69 MCU. Before using the MCU-Link functionality, ensure that it is programmed with the required firmware. For details, see Section 3.5.

1.1 Block diagram

Figure 1 shows the FRDM-MCXN947 board block diagram.

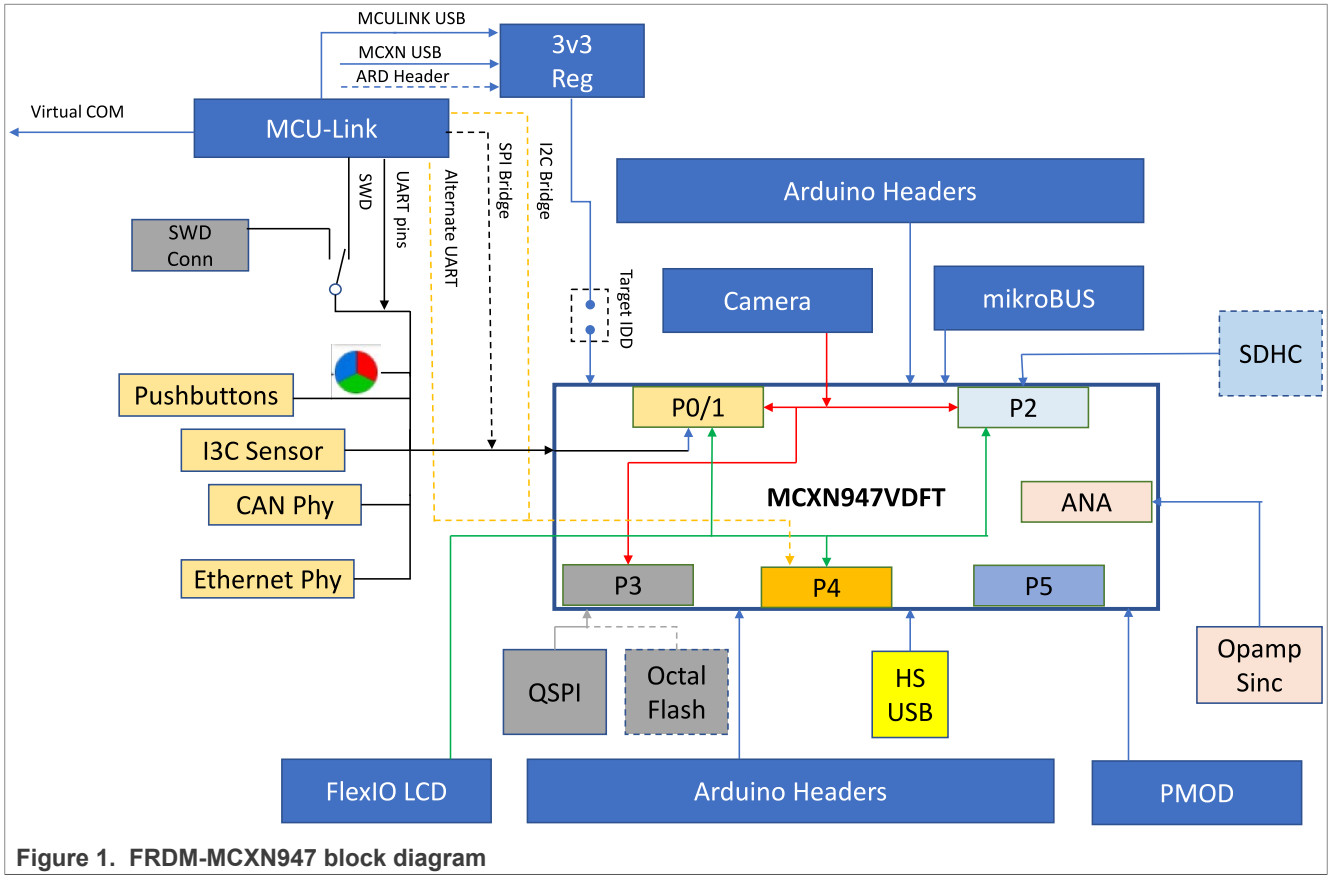

Figure 1. FRDM-MCXN947 block diagram

1.2 Board features

Table 1 describes the features of the FRDM-MCXN947 board.

Table 1. FRDM-MCXN947 features

Board feature	Target MCU features used	Description
MCXN947 MCU (target MCU)		The MCXN947 MCU is based on dual high-performance Arm Cortex-M33 cores running up to 150 MHz, with 2 MB of Flash with configurable full ECC RAM, a DSP coprocessor and an integrated proprietary neural processing unit (NPU). For details, see the *MCX Nx4x Reference Manual*.
Power supply		• P5V0 (5 V) input power supply using one of the following power sources: – High-speed USB2.0 Type-C connector – 5 V regulator populated at 3-pin jumper – Arduino Shield compatible header – MCU-Link USB2.0 Type-C connector • One LDO for 3.3 V power supply • Jumpers and resistors configuration for different power supplies
Clock		Crystal oscillators for: • 24 MHz system reference clock • 32.768 kHz real-time clock (RTC) • 50 MHz Ethernet PHY clock input • 16 MHz clock input for MCU-Link
USB	High-speed (HS) USB module	One USB Type-C connector interfaced with high-speed USB controller and PHY module
Flash memory	FlexSPI controller	Supports either of the following two options: • Winbond W25Q64JVSSIQ - 64 Mbit QSPI flash memory (populated) • Micron MT35XL512ABA1G12 - 512 Mbit Octal flash memory (DNP)
Temperature sensor	Improved inter-integrated circuit (I3C)	Supports NXP P3T1755DP temperature sensor
Ethernet	Ethernet controller (ENET0)	10 / 100 Mbit/s (RMII) LAN8741 Ethernet PHY and RJ45 connector
I/O headers		Headers compatible with: • Arduino shields (outer rows) and FRDM header (inner rows) • Mikroe click boards • Camera module • LCD module • Peripheral module (Pmod)
CAN	FlexCAN controller	One TJA1057GTK/3Z CAN PHY and one 4-pin CAN FD connector
Mass storage	µSDHC	MicroSD card connector (DNP by default)
Touch pad	Touch sensing input (TSI)	One Touch Slider for touch sensing detection; connects to the TSI input channel through the P1_3 pin
Debug		• Onboard MCU-Link debug probe with CMSIS-DAP and SEGGER J-Link protocol options. It can connect to the target MCU through a USB-to-UART, USB-to-SPI, or USB-to-I2C bridge. • 10-pin Arm JTAG/SWD connector for connecting an external debug probe

Table 1. FRDM-MCXN947 features*...continued*

Board feature	Target MCU features used	Description
PCB		118 mm x 55 mm
Orderable part number		FRDM-MCXN947

1.3 Board kit contents

The FRDM-MCXN947 board kit contains the following items:

- FRDM-MCXN947 board hardware assembly
- A 3 ft micro USB Type A to USB Type C cable

1.4 Board pictures

Figure 2 shows the top view of FRDM-MCXN947.

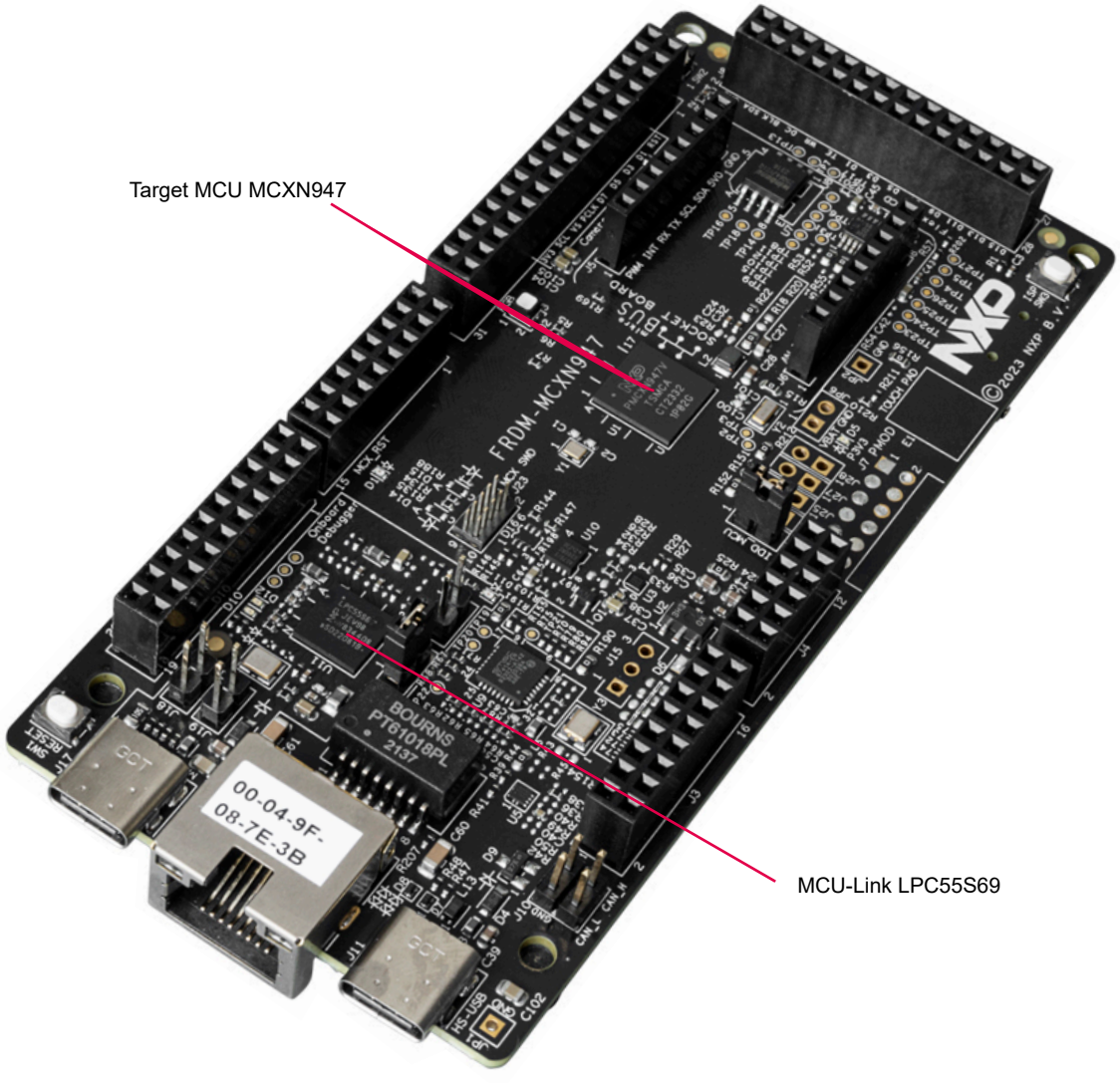

Figure 2. FRDM-MCXN947 top view

Figure 3 shows the top-side view of the FRDM-MCXN947 board, with connectors, push buttons, and LEDs highlighted.

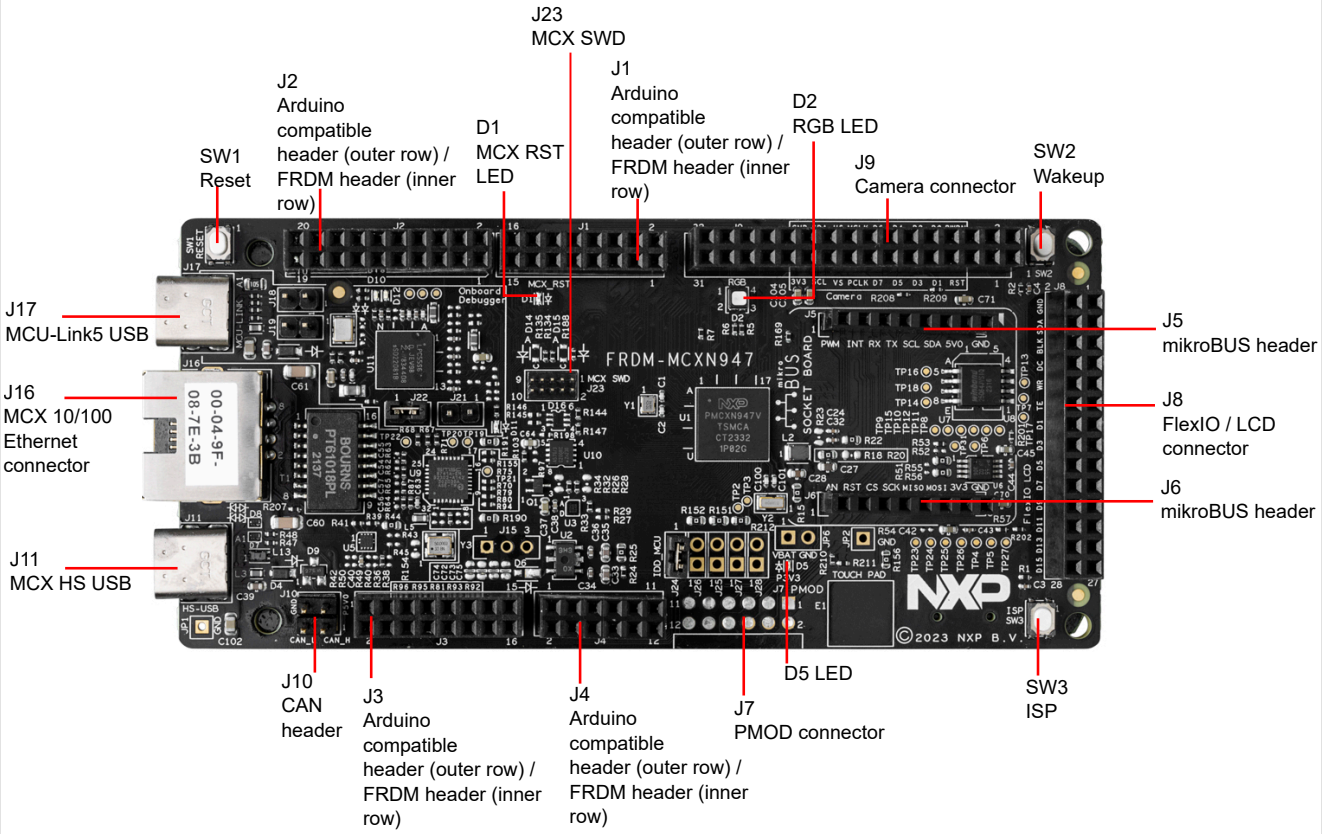

Figure 3. FRDM-MCXN947 connectors, push buttons, and LEDs (top-side view)

Figure 4 shows the jumpers on the FRDM-MCXN947 board.

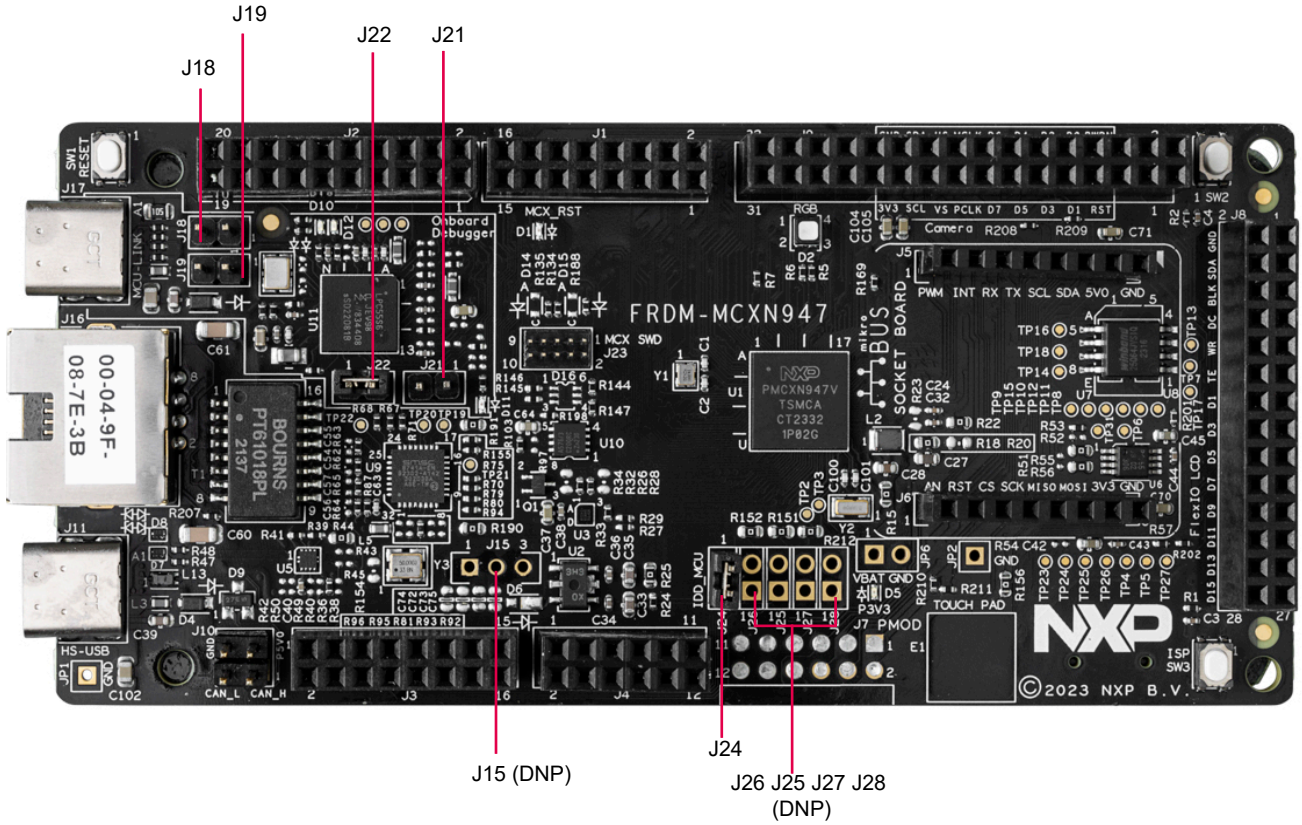

Figure 4. FRDM-MCXN947 jumpers

1.5 Connectors

Table 2 describes the FRDM-MCXN947 connectors. The connectors are shown in Figure 3.

Table 2. FRDM-MCXN947 connectors

Part identifier	Connector type	Description	Reference section
J1	2 x 8 pin header	Arduino compatible I/O header (outer rows) and FRDM header (inner rows)	Section 2.9
J2	2 x 10 pin header		
J3	2 x 8 pin header		
J4	2 x 6 pin header		
J5	1 x 8 position receptacles	mikroBUS socket connector	Section 2.11
J6	1 x 8 position receptacles	mikroBUS socket connector	Section 2.11
J7	2 x 6 header (DNP)	Pmod connector	Section 2.13
J8	2 x 14 pin header	FlexIO header for LCD screen connection	Section 2.10
J9	2 x 16 pin header	Used for connecting a camera	Section 2.12
J10	2 x 2 pin header	Connects to the CAN bus and allows external connection with the bus	Section 2.5
J11	USB Type-C connector	USB1 high-speed connector	Section 2.3
J12	SDHC connector (DNP)	To connect an external SD card	Section 2.7

Table 2. FRDM-MCXN947 connectors*...continued*

Part identifier	Connector type	Description	Reference section
J16	RJ45 connector	Shielded RJ45 connector jack with magnetic built-in	Section 2.4
J17	USB connector	MCU-Link USB connector	Section 3.7
J23	2 x 5 pin header	Debug (JTAG / SWD) connector to connect an external debug probe or external debug target	Section 3.8

1.6 Jumpers

Table 3 describes the FRDM-MCXN947 jumpers. The jumpers are shown in Figure 4.

Table 3. FRDM-MCXN947 jumpers

Part identifier	Jumper type	Description	Reference section
J18	1x2 pin header	• Open (**default setting**): MCU-Link VCOM port is enabled. • Shorted: Sends a low signal on LPC_HW_VER_6 to disable MCU-Link VCOM port	Section 3.8
J19	1x2 pin header	• Open (**default setting**): Enables the MCU-Link SWD feature • Shorted: Sends a low signal on LPC_HW_VER_7 to disable the onboard MCU-Link SWD feature ***Note:*** *This configuration is required to enable target MCU debug through an external debug probe.*	Section 3.3
J21	1x2 pin jumper	MCU-Link (LPC55S69) force ISP mode jumper: • Open (**default setting**): MCU-Link follows the normal boot sequence (MCU-Link boots from internal flash if a boot image is found). With the internal flash erased, the MCU-Link normal boot sequence falls through to ISP boot mode. • Shorted: MCU-Link is forced to ISP mode (USB). Use this setting to reprogram the MCU-Link internal flash with a new image or use the MCUXpresso IDE with the CMSIS-DAP protocol. ***Note:*** *By default, MCU-Link flash is preprogrammed with a version of CMSIS-DAP firmware.*	Section 3.5
J22	1x2 pin jumper	MCU-Link SWD clock enable jumper: • Open: MCU-Link SWD clock is disabled. • Shorted **(default setting)**: MCU-Link SWD clock is enabled.	For more information on this jumper, see the FRDM-MCXN947 schematic
J24	1x2 pin header	Pin 1-2 shorted (**default setting**): P3V3_MCU is sourced from the P3V3 power supply	Section 2.1.1
J25 (DNP)	1x2 pin jumper	DNP by default. If shorted, VDD_IO_USB is powered by P3V3_MCU through the jumper. ***Note:*** *BY default, VDD_IO_USB is powered from the P3 V3_MCU supply through a zero-ohm shunt resistor.*	Section 2.1.1
J26 (DNP)	1x2 pin jumper	DNP by default.	Section 2.1.1

Table 3. FRDM-MCXN947 jumpers*...continued*

Part identifier	Jumper type	Description	Reference section
		If shorted, VDD_ANA is powered by P3V3_MCU through the jumper. ***Note:*** *BY default, VDD_ANA is powered from the P3V3_MCU supply through a zero-ohm shunt resistor.*	
J27 (DNP)	1x2 pin jumper	DNP by default. If shorted, VDD_BAT is powered by P3V3_MCU through the jumper. ***Note:*** *BY default, VDD_BAT is powered from the P3V3_MCU supply through a zero-ohm shunt resistor.*	Section 2.1.1
J28 (DNP)	1x2 pin jumper	DNP by default. If shorted, VDD_CORE_SYS is powered by P3V3_MCU through the jumper. ***Note:*** *BY default, VDD_CORE_SYS is powered from the P3V3_MCU supply through a zero-ohm shunt resistor.*	Section 2.1.1

1.7 Push buttons

Tactile buttons are populated on the FRDM-MCXN947 board for human machine interaction (HMI). Each of the SW[1:3] buttons has a 0.1 µF bypass capacitor for debouncing and pads for external pull-up resistors.

Table 4 describes the FRDM-MCXN947 push buttons. The push buttons are shown in Figure 3.

Table 4. FRDM-MCXN947 push buttons

Part identifier	Switch name	Description
SW1	Reset button (MCXN947 RST)	Pressing SW1 resets the target MCU that causes board peripherals to reset to their default states and execute the boot code. When SW1 is pressed, the reset LED D1 turns ON.
SW2	Wakeup button	SW2 is a general-purpose input and a low-power wake up unit (WUU) pin. Pressing SW2 gives a low level on P0_23/WAKEUP_B, otherwise, it is a high level on P0_23/WAKEUP_B.
SW3	In-system programming (ISP) mode switch	SW3 is an ISP mode switch and can also act as a general-purpose input. Pressing SW3 gives a low level on P0_6/ISPMODE_N-DEBUG, otherwise, it is a high level on P0_6/ISPMODE_N-DEBUG.

Figure 5 shows the circuit diagrams of the FRDM-MCXN947 push buttons.

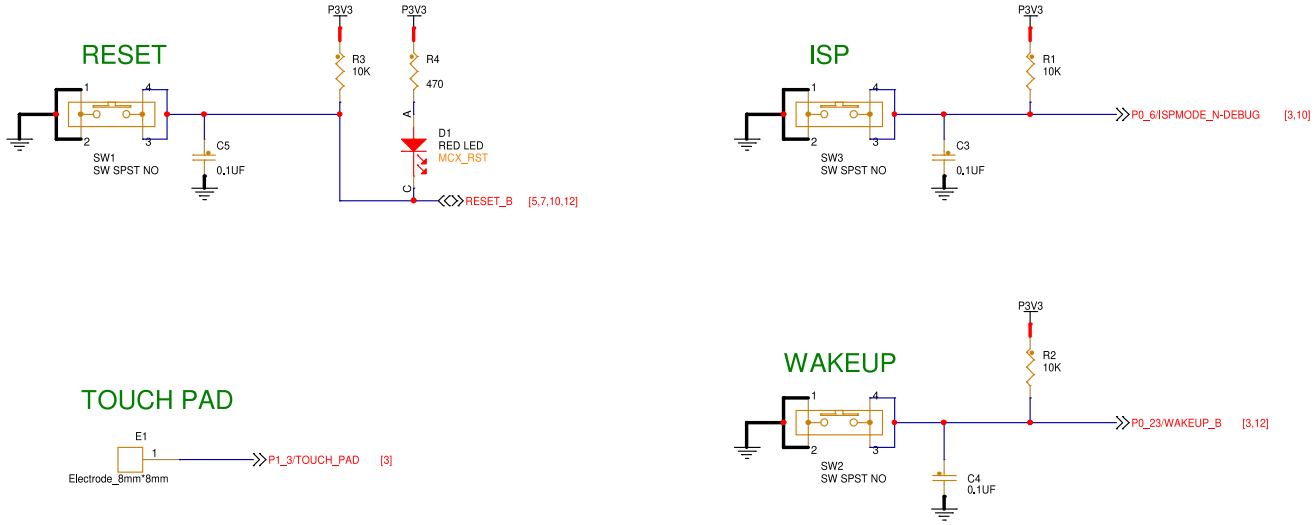

Figure 5. Push button circuit diagram

1.8 LEDs

Table 5 describes the FRDM-MCXN947 light-emitting diodes (LEDs) that correspond to the target MCU. The board also has some MCU-Link-specific LEDs, which are described in Section 3.10. The LEDs are shown in Figure 3.

Note: *The FRDM-MCXN947 board also has three status indicator LEDs for MCU-Link. For details, see Section 3.10.*

Table 5. FRDM-MCXN947 LEDs

Part identifier	LED color	LED name / function	Description
D1	Red	Reset LED	Indicates system reset activity. When board reset is initiated, for example, by pressing the SW1 reset button, the D1 LED turns ON.
D2	Red/green/blue	RGB LED	User application LEDs. Each of these LEDs can be controlled through a user application. • Red LED connects to target MCU pin P0_10 • Green LED connects to target MCU pin P0_27 • Blue LED connects to target MCU pin P1_2
D5	Green	P3V3 PWR ON	Indicates P3V3 power on status. When P3V3 is available on board, D5 turns ON.

Figure 6 shows the circuit diagram of the RGB LEDs described in Table 5.

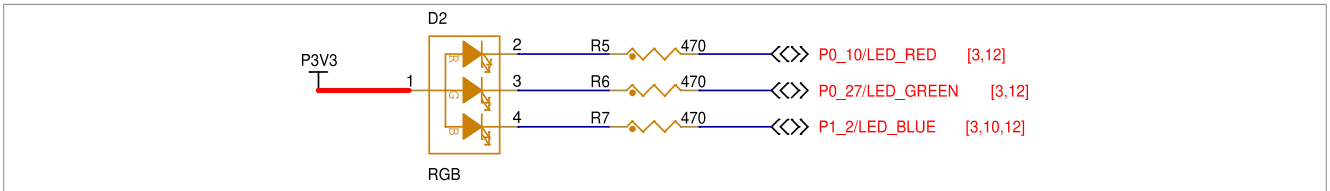

Figure 6. RGB LED circuit diagram

2 FRDM-MCXN947 functional description

This section describes the features and functions of the FRDM-MCXN947 board. You can use the functionality described in this section as a reference while designing your own target board.

Note: *For more details on the MCXN947 MCU, see MCX N94x, N54x Product Family Data Sheet and MCX Nx4x Reference Manual.*

2.1 Power supplies

The FRDM-MCXN947 board is powered with a P5V0 (5 V) power supply using one of the following source options:

- P5V_USB_HS supply from high-speed (HS) USB2.0 Type-C connector (J11)
- P5V_HDR_IN supply from 5 V regulator populated at 3-pin jumper (J15) (Not populated by default)
- P5V0 supply from Arduino Shield compatible header, J3 (pin 10)
- P5V_MCU_LINK_USB supply from MCU-Link USB2.0 Type-C connector (J17)

The P5V0 supply is an input power supply on the board and is a source for secondary power supplies.

Other power supplies in the FRDM-MCXN947 board are through voltage regulators or are connected through jumpers, which can be used to enable/disable a power supply.

Section 2.1 shows the system power circuit on the FRDM-MCXN947 board.

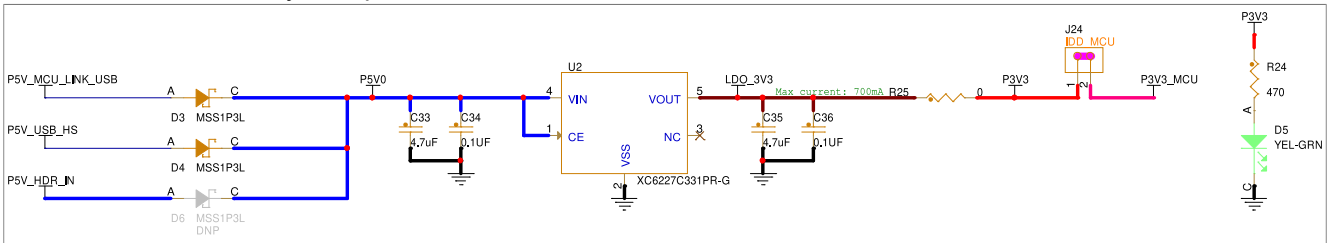

Figure 7. FRDM-MCXN947 system power circuit diagram

5 V power sources and selection

Table 6 describes the 5 V input power sources and their output power supplies.

Table 6. 5 V power sources

Part identifier	Device / power source	Output power supply	Description
J17	MCU-Link USB2.0 Type-C connector	P5V_MCU_LINK_USB	• One of the sources of P5V0 (5 V) supply **(default option)** • USB regulator input power supply for MCU-Link microcontroller LPC55S69
J11	HS USB2.0 Type-C connector	P5V_USB_HS	One of the sources for P5V0 (5 V) supply
J15	5 V power regulator populated at J15	P5V_HDR_IN	One of the sources for P5V0 (5 V) supply
J3 (pin 10)	Arduino shield compatible header	P5V0	Power supply for: • XC6227C331PR-G LDO voltage regulator (U2) • TJA1057 CAN PHY (U10) and CAN 2x2-pin header (J10) • mikroBUS connector (J5)

Table 6. 5 V power sources...*continued*

Part identifier	Device / power source	Output power supply	Description
			• HS USB connector power switch NX5P3090 UK (U3)

3.3 V power sources and selection

Table 7 describes the 3.3 V input power sources and their output power supplies.

Table 7. P3V3 power sources

Part identifier	Device / power source	Output power supply	Description
U2	XC6227C331PR-G (TOREX)	LDO_3V3	One of the sources for P3V3 supply **(default selection)**
J3 (pin 8)	Arduino shield compatible header	P3V3	• Power supply for: – PTN5150A USB Type-C CC logic – P3T1755DP I3C sensor – QSPI Flash memory (W25Q64JVSSIQ) – SD card connector J12 (DNP) – Crystal oscillator (Y3) – CAN transceiver TJA1057 (U10) – RGB LED (D2) – mikroBUS connector (J6) – Pmod connector (J7) – Parallel camera connector (J9) – FLEXIO / LCD connector (J8) – MCU-Link LPC55S69 (U11) • Power source for the P3V3_MCU supply through the J24 jumper. For details, see Section 1.6 • Power source for EPHY_VDDIO and EPHY_VDDA supplies for Ethernet transceiver LAN8741 (U9)

2.1.1 Power supply configuration

Once the main power configurations are set, the target MCU power configurations must be made. The MCU power is configured by a network of jumpers or by a combination of resistors, capacitors, and diodes as shown in Figure 8.

These jumpers provide access to insert ammeters in all the supplies connecting to the MCX N947 device. They also provide a means of connecting external supplies to any of the MCX power pins.

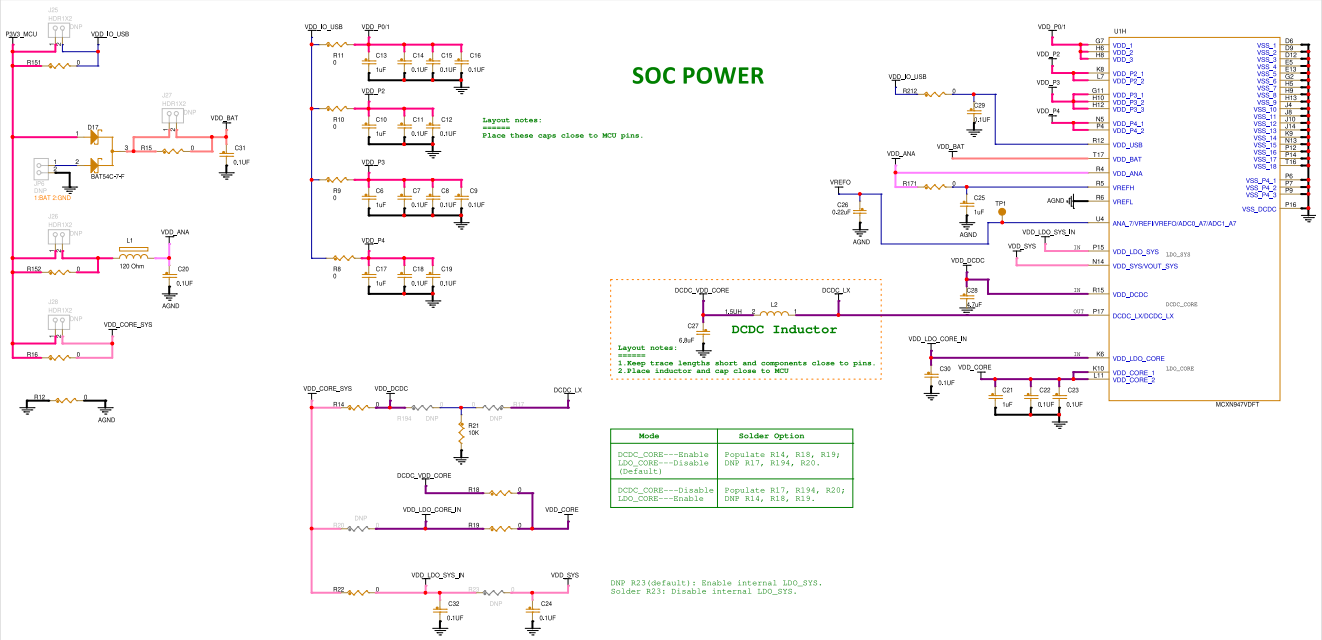

Figure 8. Power configuration

Table 8 describes the power supply configurations for MCU analog, USB and other operations.

Table 8. MCU power supplies

Power source	Zero-ohm resistor or Jumper used	Power supply rail	Description
P3V3_MCU	• R151 resistor (installed) • R212 resistor (installed) • J25 jumper (DNP)	VDD_IO_USB	• Power supply for the MCU power pin VDD_USB • Power source for VDD_P0/1, VDD_P2, VDD_P3, and VDD_P4 power supplies
	• R15 resistor (installed) • J27 jumper (DNP)	VDD_BAT	Power supply for VBAT domain and Port 5
	• R152 resistor (installed) • J26 jumper (DNP)	VDD_ANA	Power supply for: • MCU analog power pin VDD_ANA • Arduino header J2 pin 16
	• R16 resistor (installed) • J28 jumper (DNP)	VDD_CORE_SYS	Power source for: • VDD_DCDC • VDD_LDO_CORE_IN • VDD_LDO_SYS_IN

Table 9 describes the typical power supply configurations for DCDC and LDO operations.

Table 9. Power supply configurations for DCDC and LDO operation

Mode	Solder options for resistors					
	R14	R18	R19	R17	R194	R20
• DCDC_CORE enable • LDO_CORE disable **(Default setting)**	Populate	Populate	Populate	DNP	DNP	DNP

Table 9. Power supply configurations for DCDC and LDO operation*...continued*

Mode	Solder options for resistors					
	R14	R18	R19	R17	R194	R20
• DCDC_CORE disable • LDO_CORE enable	DNP	DNP	DNP	Populate	Populate	Populate

2.1.2 DC-DC inductor

The FRDM-MCXN947 board uses a 1.5 µH DC-DC inductor L2 (SHENZHEN SUNLORD SPH252012H1R5MT). The inductor is enabled when the board is configured in DC-DC Buck mode. Figure 8 shows the DC-DC inductor circuit diagram of the FRDM-MCXN947 board.

Choosing the right DC-DC inductor for your target board is important. When selecting a DC-DC inductor, refer to the specifications mentioned in the *MCX N94x, N54x Product Family Data Sheet*.

2.2 Clocks

The FRDM-MCXN947 board provides crystal oscillators to provide accurate time bases for the device and different components on the board.

Table 10 describes the clock sources available on the FRDM-MCXN947 board.

Table 10. FRDM-MCXN947 clocks

Clock generator	Clock frequency	Destination	Description
Crystal oscillator, Y1 (830108212309, Wurth Electronics)	24 MHz	• Port 1 pin 30 (XTAL48M) of target MCU MCXN947 • Port 1 pin 31 (EXTAL48M) of target MCU MCXN947	For high-frequency accurate timebase • Required external load capacitors are provided • Small package size (2.0 mm x 1.6 mm) • Low-ESR (100 Ω max) crystal
Crystal oscillator, Y2 (830009678, Wurth Electronics)	32.768 kHz	• Port 5 pin 1 (XTAL32K) of target MCU MCXN947 • Port 5 pin 0 (EXTAL32K) of target MCU MCXN947	For accurate low-power timebase • Internal load capacitors provide the entire crystal load capacitance • To measure the 32.768 kHz oscillator frequency, enable the RTC_CLKOUT signal to be available on the P5_3 pin. It can be observed at pin5 of connector J2
Crystal oscillator, Y3 (ECS-2333-500-BN-TR)	50 MHz	RMII 10/100 Mbit/s Ethernet transceiver LAN8741A-EN (U9)	Provides clock at XTAL1/CLKIN pin of Ethernet PHY depending upon the R154 resistor setting (DNP by default). Also, it provides a provision to feed the clock back into the target MCU (MCXN947) through Port P1 pin 4 (P1_4/ENET_TXCLK)
Crystal oscillator, Y4 (830064296, Wurth Electronics)	16 MHz	XTAL32M_N/P pins of LPC55S69 MCU-Link	Option for external clock input

2.3 USB interface

The target MCU (MCXN947) features two USB modules (FS USB and HS USB), each with device and host capabilities and a built-in transceiver.

On the FRDM-MCXN947 board, only the HS USB controller and PHY interface is used and it is connected to USB Type-C connector (J11).

Figure 9 shows the HS USB circuit diagram.

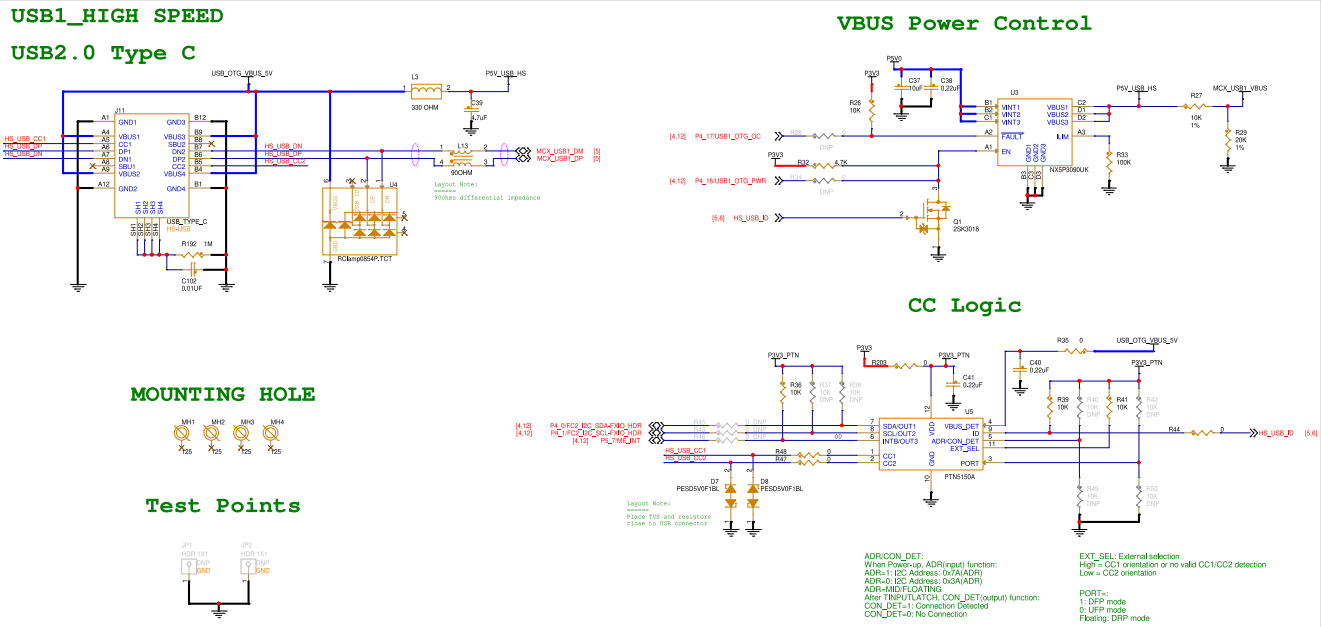

Figure 9. HS USB circuit diagram

Table 11 describes the devices used for connection between HS USB controller and USB Type-C connector.

Table 11. USB Ports

Part identifier	Connector type	Description
J11	USB2.0 Type-C connector	Port can connect in both Host and Device mode. In Device mode, this port provides the 5 V power supply (P5V_USB_HS) source to the board.
U3	NX5P3090UK	USB Power Delivery (PD) and type C current-limited power switch
U5	PTN5150A	CC Logic chip supporting the USB Type-C connector application with Configuration Channel (CC) control logic detection and indication functions • ADR/CON_DET pin configuration: – When pull up to P3V3_PTN with 10 kΩ resistor (R40), ADR (input) function: ADR=1: I2C Address: 0x7A (ADR) – When pull down to GND with 10 kΩ resistor (R49), ADR (input) function: ADR=0: I2C Address: 0x3A (ADR) – ARD=Mid/Floating **(default setting)**, this pin automatically switches from input to CON_DET output in non-I2C mode after TINPUTLATCH – CON_DET (output) function: – CON_DET=1: Connected detected – CON_DET=0: No connection • PORT pin configuration: – When pull up to P3V3_PTN with 10 kΩ resistor (R42), PORT=1: DFP mode – When pull down to GND with 10 kΩ resistor (R50), PORT=0: UFP mode – When Floating **(default setting)**: DRP mode

On the FRDM-MCXN947 board, the USB1_DM and USB1_DP signals from the target MCU connect to the onboard USB connector (J11) directly through a common mode choke. The common mode choke is included for noise suppression on the DM / DP signals.

2.4 Ethernet interface

The target MCU (MCXN947) features one Ethernet controller (ENET0) module.

On the FRDM-MCXN947 board, the Ethernet controller connects to an RJ45 connector through an Ethernet PHY transceiver. The transmit, receive, and other Ethernet signals are on the P1 port pins. The FRDM-MCXN947 only supports RMII configuration. For this reason, the TXD3 and TXD2 pins of the Ethernet PHY (LAN8741A-EN) have been grounded through resistors R68 and R67, respectively.

Table 12 describes the onboard devices supporting the Ethernet interface.

Table 12. Ethernet interface devices

Part identifier	Part name and Manufacturer	Description
J16	Heling MJ88B-B011-RVL11-P	Shielded RJ45 connector jack with magnetic built-in to connect to an Ethernet cable
U9	Microchip Technology LAN8741 A-EN	Single-chip 10 /100 Mbit/s RMII Ethernet PHY compliant with IEEE802.3/802.3u (Fast Ethernet), ISO 802-3/IEEE 802.3 (10 BASE-T), and Energy-Efficient Ethernet IEEE 802.3az
T1	BOURNS PT61018PEL	Dual-channel 16-pin Ethernet transformer for LAN 10/100 Base-Tx

Input to the XTAL1/CLKIN pin of the Ethernet PHY is a 50 MHz clock from an external 50 MHz crystal oscillator (Y3). The oscillator is enabled by default. The clock circuit also provides a provision to feed the clock back into the target MCU (MCXN947) through Port P1 pin 4 (P1_4/ENET_TXCLK).

Note: *The 50 MHz oscillator can be disabled by populating the R154 resistor.*

2.5 FlexCAN interface

The controller area network (FlexCAN) is a full implementation of the CAN protocol specification, the CAN with flexible data rate (CAN FD) protocol, and the CAN 2.0 version B protocol, which supports both standard and extended message frames and long payloads. The target MCU (MCXN947) supports two CAN (w/wo FD) controllers (CAN0 to CAN1).

On FRDM-MCXN947, only the CAN0 controller is used. The CAN0 controller connects to a 4-pin CAN header through a CAN transceiver (TJA1057GTK/3Z). The CAN0_TXD and CAN0_RXD signals are through ports P1_10 and P1_11, respectively.

Table 13 describes the HS CAN transceiver and 4-pin CAN header used on the board.

Table 13. High-speed CAN transceiver and header

Part identifier	Manufacturing part number	Description
U10	TJA1057GTK/3Z	High-speed CAN transceiver. It provides an interface between the CAN0 controller and the physical two-wire CAN0 bus.
J10	-	4-pin CAN header. It is connected to the CAN0 bus and allows external connection with the bus.

Figure 10 shows the FlexCAN interface schematic.

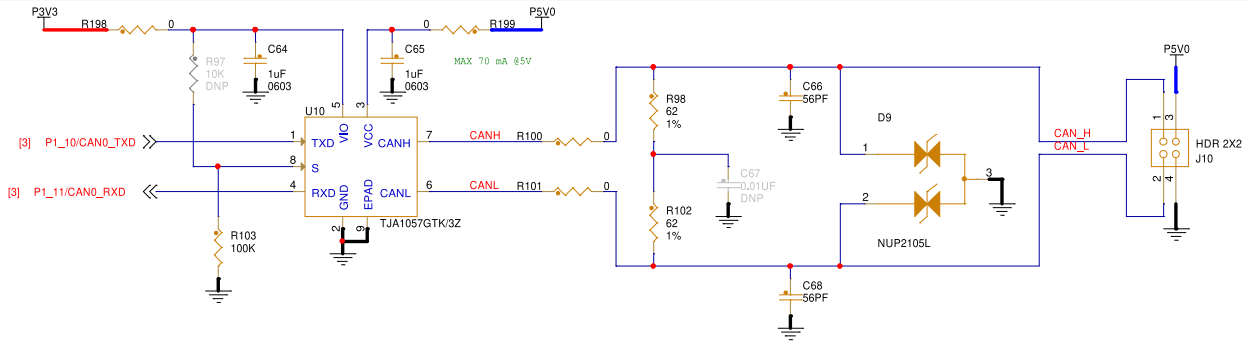

Figure 10. FlexCAN interface schematic

Table 14 describes the 4-pin J10 CAN header pinout.

Table 14. CAN header - pinout

Pin	Signal	Description
1	CAN1_H	CAN transceiver high signal
2	CAN1_L	CAN transceiver low signal
3	P5V0	5 V power supply
4	GND	Ground

2.6 I3C sensor interface

The FRDM-MCXN947 board includes one P3T1755DP digital temperature sensor to demonstrate the I3C capabilities of the target MCU. This sensor device allows for 32 I3C provisional IDs, supports the full operating voltage of the board, programmable overtemperature alerts, 12b resolution, and has an accuracy of ±0.5 °C (maximum) from –20 °C to +85 °C.

The 7-bit I2C address of the sensor device is 0b1001000 (0x48).

Figure 11 shows the I3C sensor schematic diagram.

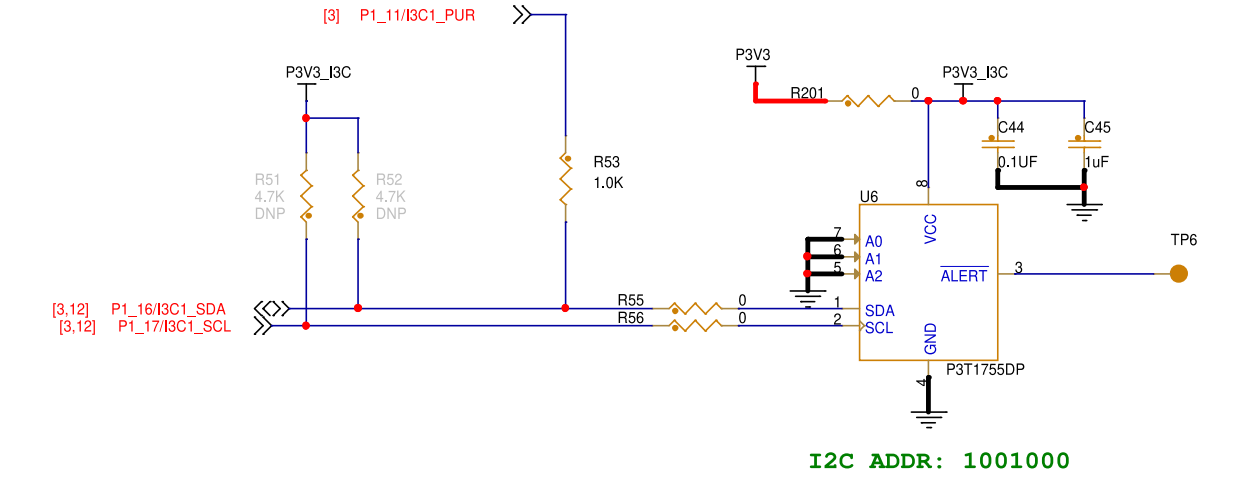

Figure 11. I3C sensor schematic diagram

The sensor device connect to the I3C1 controller of the device through P1_[16:17] Port. The data line is also connected to the GPIO (P1_11/I3C1_PUR) for the I3C Pull-Up Resistance (PUR) provision through the 1 kΩ R53 resistor.

2.7 SD card interface

The target MCU (MCX N94X) features one micro secure digital host controller (uSDHC) module.

On the FRDM-MCXN947 board, the uSDHC controller connects to the SD card connector (J12) (not populated by default).

- The SD card detect pin is an open switch that shorts with GND when the card is inserted.
- The SD card VDD (P3V3_SDHC) is supplied by the P3V3 rail through a zero-ohm resistor R202.
- All SD card bus pins connect to Port P2_[7:1] pins. Therefore, the Port P2 voltage must be set according to the type of card being used.

Figure 12 shows the SD card connections.

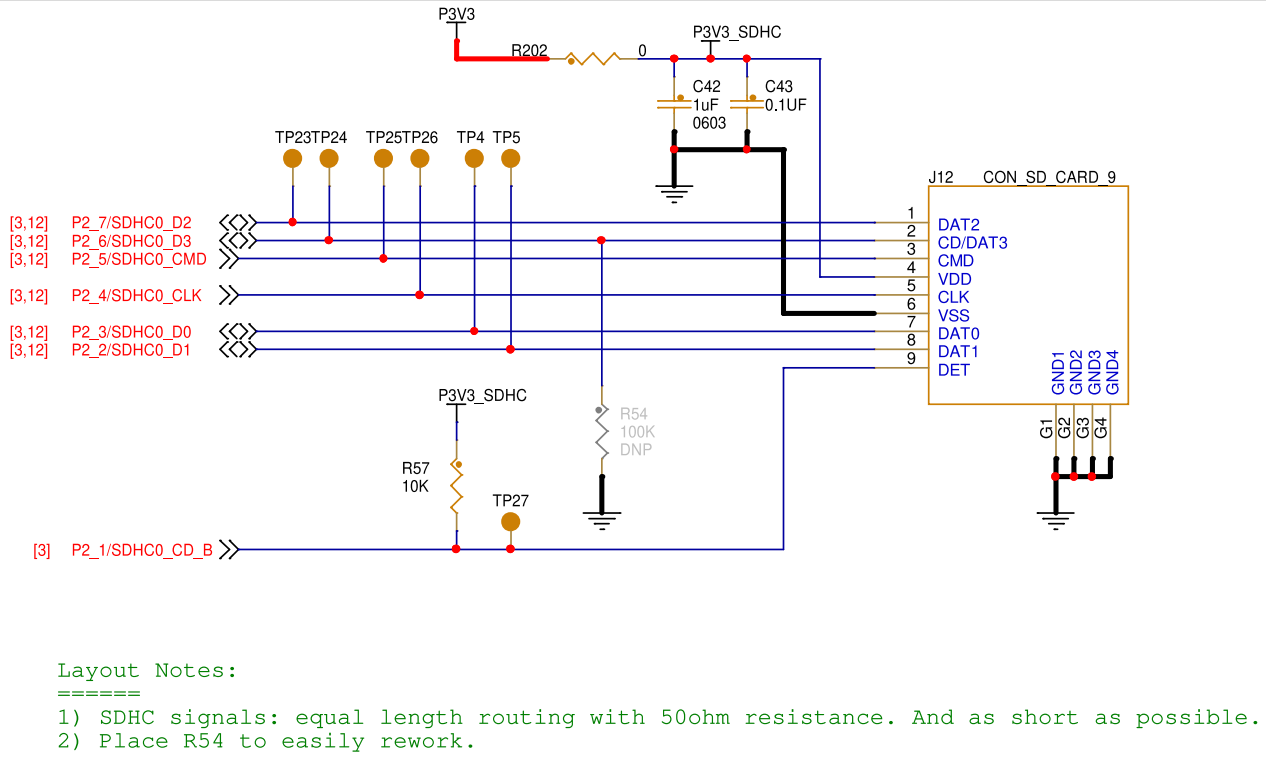

Figure 12. Micro SD card connector circuit

The Port P2_[7:2] lines that are used for the SDHC card slot are also shared with the PWM signals on the Arduino compatible headers (J1, J2, and J3). The Port P2_[2] line is shared with the camera connector and PWM signals on the Arduino compatible headers (J2 and J3). The zero-ohm onboard resistor (R168) is used to allow the selection of signals between the SDHC card slot or the camera connector. The Arduino-compatible header sockets are connected by default.

Figure 13 shows the zero-ohm resistors connection for uSDHC signals selection.

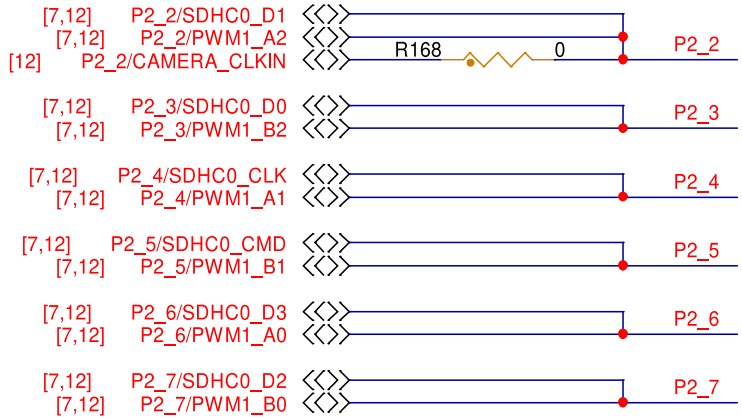

Figure 13. uSDHC card connection selection resistors

2.8 Flash memory interface

The target MCU (MCXN947) features one Flexible Serial Peripheral Interface (FlexSPI) controller, which can support an external memory.

On the FRDM-MCXN47 board, the MCU MCXN947 FlexSPI controller can connect either to an onboard QSPI flash memory (U8) or to an octal flash memory (U7). The footprint of U7 and U8 is overlapped, where U8 is populated by default and U7 is DNP.

The flash memory VCC is supplied by the P3V3 rail.

Table 15 provides the details of the flash memory used on the board.

Table 15. Flash memory

Part identifier	Manufacturer and part name	Description
U8	Winbond W25Q64JVSSIQ[1]	It is a 3 V 64 Mbit (8 MB) serial flash memory with dual and quad SPI, which is intended for demonstrating FlexSPI boot applications, and general FlexSPI operation. For main features, refer to device data sheet.
U7	Micron Technology MT35XL512 ABA1G12	512 Mbit, 2.7 V - 3.6 V, Octal I/O type, 64 MB T-PBGA NOR Flash memory

[1] The other option of QSPI Flash memory that can be used on the board is MT25QL128ABA1ESE-0SIT (MICRON).

Figure 14 shows the flash memory circuit diagram.

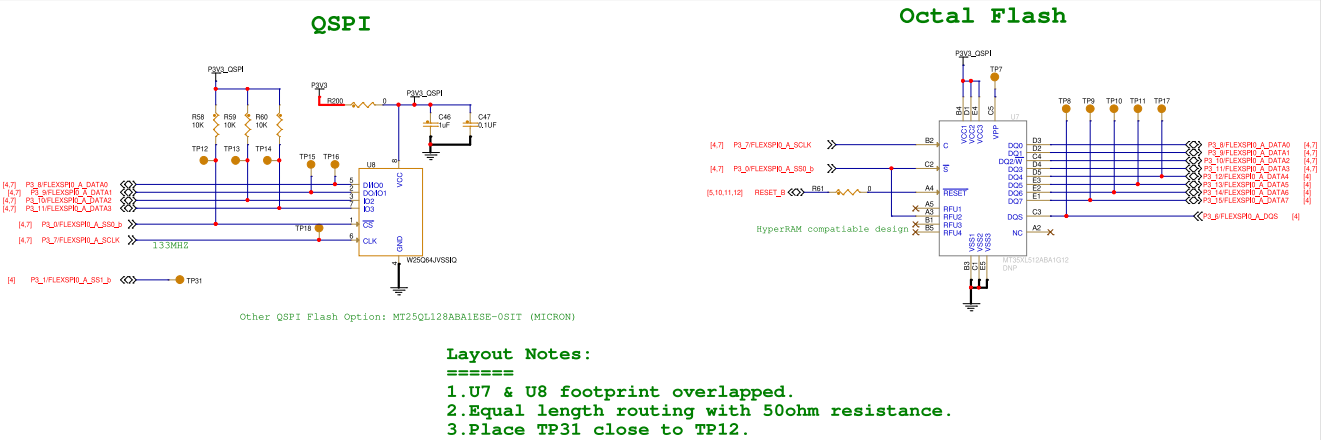

Figure 14. Flash memory circuit diagram

The FlexSPI data and clock signals for the flash memory interface are available on Port P3_[1:0] and P3_[15:6] pins.

2.9 Arduino compatible I/O headers

The FRDM-MCXN947 board provides Arduino Uno compatible headers to support the Arduino and FRDM ecosystem shield modules. These headers are dual-row headers with the outer rows supporting the Arduino compatible shields and the inner rows supporting the various FRDM shields. These headers are designed to support the following shields:

- Sensor: FRDM-STBC-AGM01, FRDM-STBC-AGM04, FRDM-FXS-MULT2-B
- NFC: OM5577, OM5578
- USB Type C: OM13790 (Host)
- Motor control: FRDM-MC-LVBLDC, FRDM-MC-LVPMSM
- Touch: FRDM-TOUCH

Table 16 describes the connectors of the Arduino socket.

Table 16. Arduino socket connectors

Part identifier	Connector type
J1	2x8 position receptacle
J2	2x10 position receptacle
J3	2x8 position receptacle
J4	2x6 position receptacle

Figure 15 shows the pinout of the Arduino socket connectors.

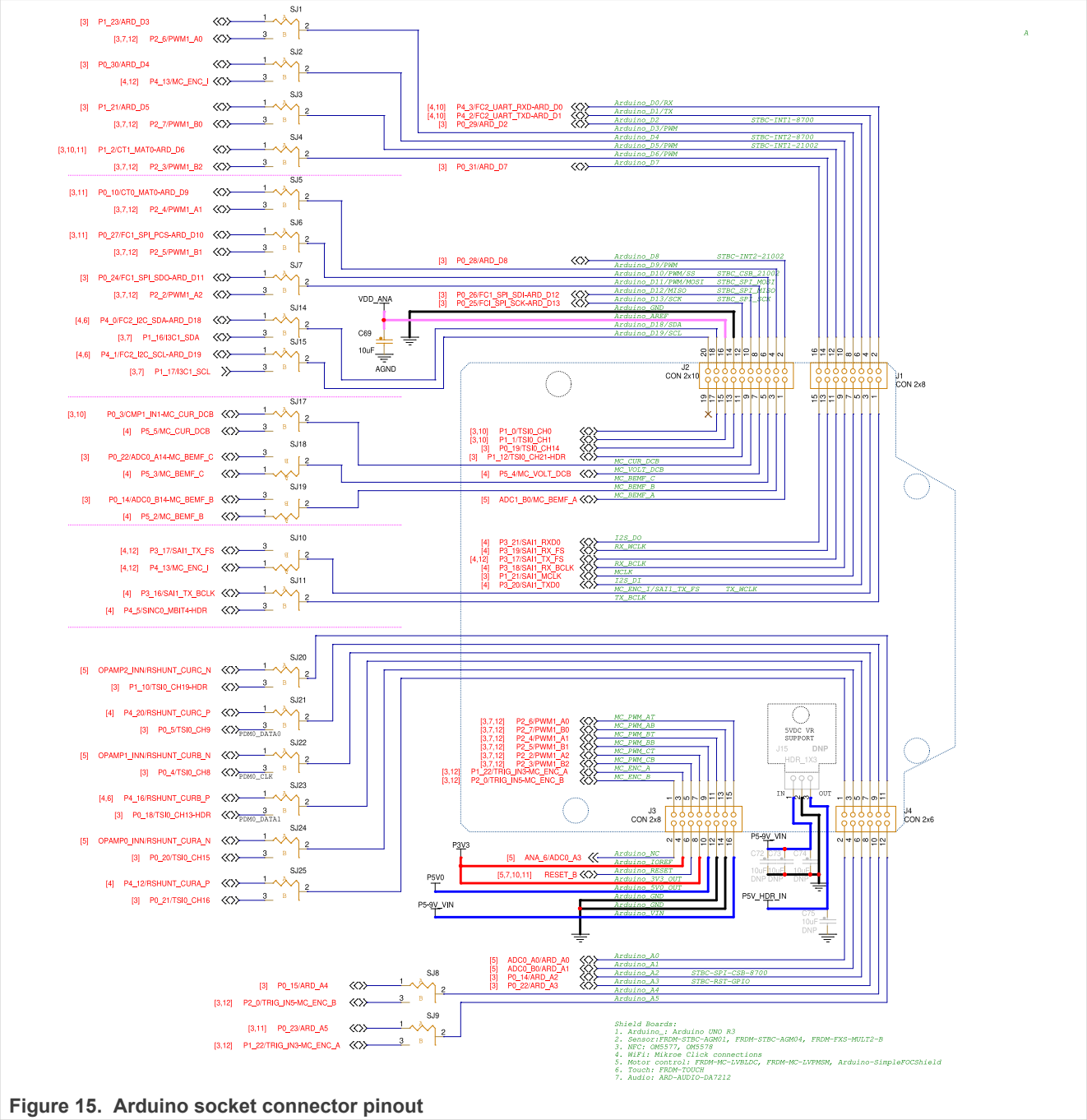

Figure 15. Arduino socket connector pinout

To allow for the flexibility in the design, some of the signals on the I/O headers can be swapped for other connections using zero-ohm resistors or jumpers. Table 16 describes such signals.

Table 17. Arduino compatible header J1 pinout

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
1	P3_16	SAI1_TX_BCLK	SJ11 Pin 1-2 selection (Default setting)	-
	P4_5	SINC0_MBIT4-HDR	SJ11 Pin 2-3 selection	FlexIO LCD connector (J8) pin 6

Table 17. Arduino compatible header J1 pinout*...continued*

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
2	P4_3	FC2_UART_RXD-ARD_D0	-	MCU-Link UART (P4_3/FC2_P3_UART_TXD_MCULINK)
3	P3_17	SAI1_TX_FS	SJ10 Pin 2-3 selection	-
	P4_13	MC_ENC_I	SJ11 Pin 1-2 selection (Default setting)	FlexIO LCD connector (J8) pin 18
4	P4_2	FC2_UART_TXD-ARD_D1	-	MCU-Link UART (P4_2/FC2_P2_UART_TXD_MCULINK)
5	P3_20	SAI1_TXD0	-	mikroBUS header J6 pin 6 (P3_20/ME_FC6_SPI_MOSI)
6	P0_29	ARD_D2	-	-
7	P1_21	SAI1_MCLK	-	• Arduino header J1 Pin 12 (P1_21/ARD_D5) • Ethernet PHY LAN8741 (P1_21/ENET_MDIO through the zero-ohm resistor R166)
8	P1_23	ARD_D3	SJ1 Pin 1-2 selection (Default setting)	Camera connector J9 pin 23 (P1_23/EZH_LCD_RD)
	P2_6	PWM1_A0	SJ1 Pin 2-3 selection	SD card connector J12 (P2_6/SDHC0_D3)
9	P3_18	SAI1_RX_BCLK	-	-
10	P0_30	ARD_D4	SJ2 Pin 1-2 selection (Default setting)	-
	P4_13	MC_ENC_I	SJ2 Pin 2-3 selection	FlexIO LCD connector (J8) pin 18
11	P3_17	SAI1_TX_FS	-	-
12	P1_21	ARD_D5	SJ3 Pin 1-2 selection (Default setting)	• Arduino header J1 Pin 7 (P1_21/SAI1_MCLK) • Ethernet PHY LAN8741 (P1_21/ENET_MDIO through the zero-ohm resistor R166)
	P2_7	PWM1_B0	SJ3 Pin 2-3 selection	SD card connector J12 (P2_7/SDHC0_D2)
13	P3_19	SAI1_RX_FS	-	mikroBUS header J5 pin 1 (P3_19/ME_PWM)
14	P1_2	CT1_MAT0-ARD_D6	SJ4 Pin 1-2 selection (Default setting)	• RGB LED (P1_2/LED_BLUE) • MCU-Link USB bridge (P1_2/FC1_SPI_SDI_MCULINK)
	P2_3	PWM1_B2	SJ4 Pin 2-3 selection	SD card connector J12 (P2_3/SDHC0_D0)
15	P3_21	SAI1_RXD0	-	mikroBUS header J6 pin 4 (P3_21/ME_FC6_SPI_CLK)
16	P0_31	ARD_D7	-	-

Table 18. Arduino compatible header J2 pinout

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
1	ADC1_B0	MC_BEMF_A	-	-
2	P0_28	ARD_D8		
3	P0_14	ADC0_B14-MC_BEMF_B	SJ19 Pin 2-3 selection	Arduino header J4 Pin 6 (P0_14/ ARD_A2)
	P5_2	MC_BEMF_B	SJ19 Pin 1-2 selection (default setting)	-
4	P0_10	CT0_MAT0-ARD_D9	SJ5 Pin 1-2 selection (default setting)	RGB LED (P0_10/LED_RED)
	P2_4	PWM1_A1	SJ5 Pin 2-3 selection	SD card connector (P2_4/ SDHC0_CLK)
5	P0_22	ADC0_A14-MC_BEMF_C	SJ18 Pin 2-3 selection	Arduino header J4 Pin 8 (P0_22/ ARD_A3)
	P5_3	MC_BEMF_C	SJ18 Pin 1-2 selection (default setting)	-
6	P0_27	FC1_SPI_PCS-ARD_D10	SJ6 Pin 1-2 selection (default setting)	RGB LED (P0_27/LED_GREEN)
	P2_5	PWM1_B1	SJ6 Pin 2-3 selection	SD card connector (P2_5/ SDHC0_CMD)
7	P5_4	MC_VOLT_DCB	-	-
8	P0_24	FC1_SPI_SDO-ARD_D11	SJ7 Pin 1-2 selection (default setting)	-
	P2_2	PWM1_A2	SJ7 Pin 2-3 selection	• Camera connector J9 pin 16 (P2_2/CAMERA_CLKIN through the zero-ohm resistor R168) • SD card connector (P2_2/ SDHC0_D1)
9	P0_3	CMP1_IN1-MC_CUR_DCB	SJ17 Pin 1-2 selection (default setting)	JTAG/SWD connector (P0_3/TDI)
	P5_5	MC_CUR_DCB	SJ17 Pin 2-3 selection	-
10	P0_26	FC1_SPI_SDI-ARD_D12	-	-
11	P1_12	TSI0_CH21-HDR	-	Camera connector J9 pin 28 (P1_ 12/EZH_LCD_D8)
12	P0_25	FCI_SPI_SCK-ARD_D13	-	-
13	P0_19	TSI0_CH14	-	Pmod connector J7 Pin 1 (P0_19/ PMOD_FC0_SPI_CS)
14	-	GND	-	-
15	P1_1	TSI0_CH1	-	• mikroBUS header J5 pin 5 (P1_ 1/FC3_I2C_SCL) • MCU-Link USB bridge (P1_1/ FCI_SPI_SCK_MCULINK)
16	-	VDD_ANA	-	-

Table 18. Arduino compatible header J2 pinout*...continued*

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
17	P1_0	TSI0_CH0	-	• mikroBUS header J5 pin 6 (P1_0/FC3_I2C_SDA) • MCU-Link USB bridge (P1_0/FC1_SPI_SDO_MCULINK)
18	P4_0	FC2_I2C_SDA-ARD_D18	SJ14 Pin 1-2 selection (default setting)	-
	P1_16	I3C1_SDA	SJ14 Pin 2-3 selection	• Camera connector J9 pin 4 (P1_16/EZH_LCD_D12) • mikroBUS header J5 pin 3 (P1_16/ME_FC5_RXD)
19	-	-	-	-
20	P4_1	FC2_I2C_SCL-ARD_D19	SJ15 Pin 1-2 selection (default setting)	-
	P1_17	I3C1_SCL	SJ15 Pin 2-3 selection	• Camera connector J9 pin 3 (P1_17/EZH_LCD_D13) • mikroBUS header J5 pin 4 (P1_17/ME_FC5_TXD)

Table 19. Arduino compatible header J3 pinout

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
1	P2_0	TRIG_IN5-MC_ENC_B	-	Camera connector J9 pin 25 (P2_0/EZH_LCD_WR through the zero-ohm resistor R167)
2	ANA_6	ADC0_A3	-	-
3	P1_22	TRIG_IN3-MC_ENC_A	-	Camera connector J9 pin 24 (P1_22/EZH_LCD_DC)
4	-	P3V3	-	-
5	P2_3	PWM1_B2	-	SD card connector (J12) pin 7 (SDHC0_D0)
6	RESET_b	RESET_B	-	-
7	P2_2	PWM1_A2	-	SD card connector (J12) pin 8 (SDHC0_D1)
8	-	P3V3	-	-
9	P2_5	PWM1_B1	-	SD card connector (J12) pin 3 (SDHC0_CMD)
10	P5V0	-	-	-
11	P2_4	PWM1_A1	-	SD card connector (J12) pin 5 (SDHC0_CLK)
12	GND	-	-	-
13	P2_7	PWM1_B0	-	SD card connector (J12) pin 1 (SDHC0_D2)
14	GND	-	-	-

Table 19. Arduino compatible header J3 pinout*...continued*

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
15	P2_6	PWM1_A0	-	SD card connector (J25) pin 2 (SDHC0_D3)
16	P5-9V_VIN	-	-	-

Table 20. Arduino compatible header J4 pinout

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
1	P4_12	RSHUNT_CURA_P	SJ25 Pin 1-2 selection (default setting)	FlexIO header J8, pin 17 (P4_12/ FXIO_D20)
	P0_21	TSI0_CH16	SJ25 Pin 2-3 selection	-
2	ADC0_A0	ARD_A0	-	-
3	OPAMP0_ INN	RSHUNT_CURA_N	SJ24 Pin 1-2 selection (default setting)	-
	P0_20	TSI0_CH15	SJ24 Pin 2-3 selection	Pmod connector J7 pin 2 (P0_ 20/PMOD_INT)
4	ADC0_B0	ARD_A1	-	-
5	P4_16	RSHUNT_CURB_P	SJ23 Pin 1-2 selection (default setting)	• USB VBUS power control (USB1_OTG_PWR) • FlexIO header J8 pin 21 (P4_ 16/FXIO_D24)
	P0_18	TSI0_CH13-HDR	SJ23 Pin 2-3 selection	Pmod connector J7 pin 5 (P0_ 18/PMOD_FC0_SPI_MISO)
6	P0_14	ARD_A2	-	Arduino J2 connector pin 3 (P0_ 14/ADC0_B14-MC_BEMF_B through the resistor SJ19 Pin 2-3 selection)
7	OPAMP1_ INN	RSHUNT_CURB_N	SJ22 Pin 1-2 selection (default setting)	-
	P0_4	TSI0_CH8	SJ22 Pin 2-3 selection	• Camera connector J9 pin 17 (P0_4/EZH_CAMERA_ VSYNC) • Pmod connector J7 pin 6 (P0_ 4/PDM0_CLK through the resistor SJ12 Pin 2-3 selection)
8	P0_22	ARD_A3	-	Arduino J2 connector pin 5 (P0_ 22/ADC0_A14-MC_BEMF_C through the resistor SJ18 Pin 2-3 selection)
9	P4_20	RSHUNT_CURC_P	SJ21 Pin 1-2 selection (default setting)	FlexIO header J8 pin 25 (P4_20/ FXIO_D28)
	P0_5	TSI0_CH9	SJ21 Pin 2-3 selection	• Camera connector J9 pin 15 (P0_5/EZH_CAMERA_PCLK)

Table 20. Arduino compatible header J4 pinout*...continued*

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
				• Pmod connector J7 pin 4 (P0_5/PDM0_DATA0 through the resistor SJ13 Pin 2-3 selection)
10	P0_15	ADC	SJ8 Pin 1-2 selection (default setting)	-
	P2_0	TRIG_IN5-MC_ENC_B	SJ8 Pin 2-3 selection	Camera connector J9 pin 25 (P2_0/EZH_LCD_WR through the zero-ohm resistor R167)
11	OPAMP2_INN	RSHUNT_CURC_N	SJ20 Pin 1-2 selection (default setting)	-
	P1_10	TSI0_CH19-HDR	SJ20 Pin 2-3 selection	• Camera connector J9 pin 14 (P1_10/EZH_LCD_D6_CAMERA_D6 through the resistor SJ16 pin 2-3 selection) • CAN PHY (P1_10/CAN0_TXD through the resistor SJ16 Pin 1-2 selection)
12	P0_23	ARD_A5	SJ9 Pin 1-2 selection (default setting)	Wakeup (P0_23/WAKEUP_B)
	P1_22	TRIG_IN3-MC_ENC_A	SJ20 Pin 2-3 selection	Camera connector J9 pin 24 (P1_22/EZH_LCD_DC)

2.10 FlexIO header

On the FRDM-MCXN947 board, one 28-pin FlexIO header (J8) is provided to support the LCD display and camera applications. Table 21 describes the pinout of the FlexIO header.

The FlexIO header is intended to support the Mikroe TFT Proto 5” Capacitive display and the LCD-PAR-S035 display. These displays support 3 V I/O only, so care must be taken to ensure that ALL pins used on this connector are configured for 3V3 operation (this includes Ports 0, 1, 2 and 4).

A few of the signals on the FlexIO connector are shared signals. Before using the FlexIO connector with other modules, be sure to check the schematics to determine if there are overlapping signals with the other modules intended to be used and that the necessary FlexIO signals are correctly selected via the zero-ohm resistor selections.

Table 21. FlexIO header J8 pinout

Pin number	Net name	GPIO	Function / Signal name	Potential conflict
1	P3V3	-	-	-
2	GND	-	-	-
3	I2C_SCL	P4_1	FC2_I2C_SCL-FXIO_HDR	-
4	I2C_SDA	P4_0	FC2_I2C_SDA-FXIO_HDR	-
5	INT	P4_6	FXIO_LCD_INT	-
6	BLK	P4_5	FXIO_LCD_GPIO	Arduino connector J1 pin 1 (P4_5/SINC0_MBIT4-HDR through

Table 21. FlexIO header J8 pinout*...continued*

Pin number	Net name	GPIO	Function / Signal name	Potential conflict
				the resistor SJ11 Pin 2-3 selection (DNP by default))
7	RST	P4_7	FXIO_LCD_RST	-
8	D/C	P0_7	FXIO_LCD_DC	-
9	CS	P0_12	FXIO_LCD_CS	-
10	WR	P0_9	FXIO_LCD_WR	-
11	RD	P0_8	FXIO_LCD_RD	-
12	TE	P0_13	FXIO_LCD_TE	-
13	D0	P2_8	FXIO_D16-FXIO_HDR	-
14	D1	P2_9	FXIO_D17-FXIO_HDR	-
15	D2	P2_10	FXIO_D18-FXIO_HDR	-
16	D3	P2_11	FXIO_D19	-
17	D4	P4_12	FXIO_D20	Arduino connector J4 pin 1 (P4_12/RSHUNT_CURA_P through the resistor SJ25 Pin 1-2 selection (default setting))
18	D5	P4_13	FXIO_D21	• Arduino connector J1 pin 10 (P4_3 /MC_ENC_I through the resistor SJ2 Pin 2-3 selection (DNP by default)) • Arduino connector J1 pin 3 (P4_13/MC_ENC_I through the resistor SJ10 Pin 1-2 selection (default setting))
19	D6	P4_14	FXIO_D22	-
20	D7	P4_15	FXIO_D23	-
21	D8	P4_16	FXIO_D24	• USB power control (P4_16/USB1_OTG_PWR through zero-ohm resistor R34 (DNP by default)) • Arduino connector J4 pin 5 (P4_16/RSHUNT_CURB_P through the resistor SJ23 Pin 1-2 selection (default setting))
22	D9	P4_17	FXIO_D25	USB power control (P4_17/USB1_OTG_OC through zero-ohm resistor R28 (DNP by default))
23	D10	P4_18	FXIO_D26	-
24	D11	P4_19	FXIO_D27	-
25	D12	P4_20	FXIO_D28	Arduino connector J4 pin 9 (P4_20/RSHUNT_CURC_P through the resistor SJ21 Pin 1-2 selection (default setting))
26	D13	P4_21	FXIO_D29	-

Table 21. FlexIO header J8 pinout*...continued*

Pin number	Net name	GPIO	Function / Signal name	Potential conflict
27	D14	P4_22	FXIO_D30	-
28	D15	P4_23	FXIO_D31	-

2.11 mikroBUS headers

Table 23 and Table 22 describe the pinout of the mikroBUS headers (J6 and J5).

Table 22. J6 header pinout

Pin number	Net name	GPIO	Function / Signal name	Potential conflict
1	AN	ANA_4	ADC1_A0	-
2	RST	P1_3	ME_RESET	• Touch pad (P1_3/TOUCH_PAD through zero-ohm resistor R156) • MCU-Link USB-to-SPI bridge (P1_3/FC1_SPI_PCS_MCULINK)
3	CS	P3_23	ME_FC6_SPI_CS	-
4	SCK	P3_21	ME_FC6_SPI_CLK	Arduino connector (J1) pin 15 (P3_21/SAI1_RXD0)
5	MISO	P3_22	ME_FC6_SPI_MISO	-
6	MOSI	P3_20	ME_FC6_SPI_MOSI	Arduino connector (J1) pin 5 (P3_20/SAI1_TXD0)
7	3V3	P3V3	3.3 V power line	-
8	GND	GND	Ground	-

Table 23. J5 header pinout

Pin number	Net name	GPIO	Function / Signal name	Potential conflict
1	PWM	P3_19	ME_PWM	Arduino connector (J1) pin 13 (P3_19/SAI1_RX_FS)
2	INT	P5_7	ME_INT	-
3	RX	P1_16	ME_FC5_RXD	• I3C sensor device (P1_16/I3C1_SDA) • Arduino connector J2 pin 18 (P1_16/I3C1_SDA through the resistor SJ14 Pin 2-3 selection (DNP by default)) • Camera connector pin 4 (P1_16/EZH_LCD_D12)
4	TX	P1_17	ME_FC5_TXD	• I3C sensor device (P1_17/I3C1_SCL) • Arduino connector J2 pin 20 (P1_17/I3C1_SCL through the resistor SJ15 Pin 2-3 selection (DNP by default))

Table 23. J5 header pinout*...continued*

Pin number	Net name	GPIO	Function / Signal name	Potential conflict
				• Camera connector pin 3 (P1_17/ EZH_LCD_D13)
5	SCL	P1_1	FC3_I2C_SCL	Arduino connector (J2) pin 15 (P1_1/ TSI0_CH1)
6	SDA	P1_0	FC3_I2C_SDA	Arduino connector (J2) pin 17 (P1_1/ TSI0_CH0)
7	5V0	P5V0	5 V power line	-
8	GND	GND	Ground	-

2.12 Camera header

The FRDM-MCXN947 provides a header for the camera connection. This is to demonstrate the camera interface features of the MCXN947 device.

Note: *The FRDM-MCXN947 board is tested with the OV7670 camera.*

Note:

To use the camera, three solder jumpers SJ16, SJ26, and SJ27 must be moved over from their default positions. This change avoids issues with reading in camera data correctly. However, this change disables Ethernet functionality. If Ethernet functionality is still desired while using the camera, an alternative workaround is to move the SJ16 and SJ26 jumpers from their default position. Then, connect a wire from P1_4 (on J9 pin 8) to the left side of the R58 resistor.

Table 24 describes the camera header (J9) pinout.

Table 24. Camera header connections

Pin number	GPIO	Function	Potential conflict
1	P1_15	EZH_LCD_D11	LAN8741 Ethernet PHY (P1_15/ENET_RXD1/ MODE1 through R165 zero-ohm resistor)
2	P1_14	EZH_LCD_D10	LAN8741 Ethernet PHY (P1_14/ENET_RXD0/ MODE0 through R164 zero-ohm resistor)
3	P1_17	EZH_LCD_D13	• I3C sensor device (P1_17/I3C1_SCL) • mikroBUS connector pin 4 (P1_17/ME_ FC5_TXD) • Arduino header J2 pin 20 (P1_17/I3C1_SCL through the resistor SJ15 Pin 2-3 selection (DNP by default))
4	P1_16	EZH_LCD_D12	• I3C sensor device (P1_16/I3C1_SDA) • mikroBUS connector pin 3 (P1_16/ME_ FC5_RXD) • Arduino header J2 pin 18 (P1_16/I3C1_ SDA through the resistor SJ14 Pin 2-3 selection (DNP by default))
5	P1_19	EZH_LCD_D15_CAMERA_RST	-
6	P1_18	EZH_LCD_D14_CAMERA_PDOWN	-

Table 24. Camera header connections*...continued*

Pin number	GPIO	Function	Potential conflict
7	P1_5	EZH_LCD_D1_CAMERA_D1	LAN8741 Ethernet PHY (P1_5/ENET_TXEN through R160 zero-ohm resistor)
8	P1_4	EZH_LCD_D0_CAMERA_D0	LAN8741 Ethernet PHY (P1_4/ENET_TXCLK through the resistor SJ27 Pin 1-2 selection (default setting))
9	P1_7	EZH_LCD_D3_CAMERA_D3	LAN8741 Ethernet PHY (P1_7/ENET_TXD1 through R162 zero-ohm resistor)
10	P1_6	EZH_LCD_D2_CAMERA_D2	LAN8741 Ethernet PHY (P1_6/ENET_TXD0 through R161 zero-ohm resistor)
11	P3_5	EZH_LCD_D5_CAMERA_D5	-
12	P3_4	EZH_LCD_D4_CAMERA_D4	-
13	P1_11	EZH_LCD_D7_CAMERA_D7	• I3C sensor device (P1_11/I3C1_PUR through R169 zero-ohm resistor) • CAN PHY (P1_11/CAN0_RXD through the resistor SJ26 Pin 1-2 selection (default setting))
14	P1_10	EZH_LCD_D6_CAMERA_D6	• CAN PHY (P1_10/CAN0_TXD through the resistor SJ16 Pin 1-2 selection (default setting)) • Arduino connector J4 pin 11 (P1_10/TSI0_CH19-HDR through the resistor SJ20 Pin 2-3 selection (DNP by default))
15	P0_5	EZH_CAMERA_PCLK	• Pmod connector pin 4 (P0_5/PDM0_DATA0 through the resistor SJ13 Pin 2-3 selection (DNP by default)) • Arduino connector J4 pin 9 (P0_5/TSI0_CH9 through the resistor SJ21 Pin 2-3 selection (DNP by default))
16	P2_2	CAMERA_CLKIN	SDHC connector (P2_2/SDHC0_D1)
17	P0_4	EZH_CAMERA_VSYNC	• Pmod connector pin 6 (P0_4/PDM0_CLK through the resistor SJ12 Pin 2-3 selection (DNP by default)) • Arduino connector J4 pin 5 (P0_4/TSI0_CH8 through the resistor SJ22 Pin 2-3 selection (DNP by default))
18	P0_11	EZH_CAMERA_HSYNC	-
19	P3_3	FC7_I2C_SCL	-
20	P3_2	FC7_I2C_SDA	-
21	-	P3V3	-
22	-	GND	-
23	P1_23	EZH_LCD_RD	Arduino connector J1 pin 8 (P1_23/ARD_D3 through the resistor SJ1 Pin 1-2 selection (default setting))
24	P1_22	EZH_LCD_DC	• Arduino connector J3 pin 3 (P1_22/TRIG_IN3-MC_ENC_A)

Table 24. Camera header connections...*continued*

Pin number	GPIO	Function	Potential conflict
			• Arduino connector J4 pin 12 (P1_22/TRIG_IN3-MC_ENC_A through the resistor SJ9 Pin 2-3 selection (DNP by default))
25	P2_0	EZH_LCD_WR	• Arduino connector J3 pin 1 (P2_0/TRIG_IN5-MC_ENC_B) • Arduino connector J4 pin 10 (P2_0/TRIG_IN5-MC_ENC_B through the resistor SJ8 Pin 2-3 selection (DNP by default))
26	P4_4	EZH_LCD_CS	-
27	P1_13	EZH_LCD_D9	LAN8741 Ethernet PHY (P1_13/ENET_RXDV/MODE2 through R163 zero-ohm resistor)
28	P1_12	EZH_LCD_D8	Arduino connector J2 pin 11 (P1_12/TSI0_CH21-HDR)
29	P5_9	-	-
30	P1_9	FC4_P1_UART_TXD_MCULINK	MCU-Link UART (P1_9/FC4_P1_UART_TXD_MCULINK)
31	P5_8	-	-
32	P1_8	FC4_P0_UART_RXD_MCULINK	MCU-Link UART (P1_8/FC4_P0_UART_RXD_MCULINK)

2.13 Pmod header

The FRDM-MCXN947 board supports a Pmod header J7 (Digilent PPPC062LJBN-RC) for connecting peripheral modules. Table 25 describes the pinout of the Pmod header.

Table 25. Pmod header pinout

Pin number	GPIO	Function name / Signal name	Resistor setting	Potential conflict
1	P0_19	PMOD_FC0_SPI_CS	-	Arduino header J2 Pin 13 (P0_19/TSI0_CH14)
2	P0_20	PMOD_INT	-	Arduino header J4 Pin 3 (P0_20/TSI0_CH15 through the resistor SJ24 Pin 2-3 selection)
3	P0_16	PMOD_FC0_SPI_MOSI		MCU-Link USB-to-I2C bridge (P0_16/FC2_I2C_SDA-MCULINK through the resistor R142 (DNP))
4	P5_6	PMOD_RESET	SJ13 Pin 1-2 selection (default setting)	-
	P0_5	PDM0_DATA0	SJ13 Pin 2-3 selection	• Camera connector J9 pin 15 (P0_5/EZH_CAMERA_PCLK) • Arduino header J4 Pin 9 (P0_5/TSI0_CH9 through the resistor SJ21 Pin 2-3 selection)

Table 25. Pmod header pinout*...continued*

Pin number	GPIO	Function name / Signal name	Resistor setting	Potential conflict
5	P0_18	PMOD_FC0_SPI_MISO	-	Arduino header J4 Pin 5 (P0_18/TSI0_CH13-HDR through the resistor SJ23 Pin 2-3 selection)
6	P3_3	FC7_I2C_SCL	SJ12 Pin 1-2 selection (default setting)	-
	P0_4	PDM0_CLK	SJ12 Pin 2-3 selection	• Camera connector J9 pin 17 (P0_4/EZH_CAMERA_VSYNC) • Arduino header J4 Pin 7 (P0_4/TSI0_CH8 through the resistor SJ22 Pin 2-3 selection)
7	P0_17	PMOD_FC0_SPI_CLK	-	MCU-Link USB-to-I2C bridge (P0_17/FC2_I2C_SCL-MCULINK through the resistor R143 (DNP))
8	P3_2	FC7_I2C_SDA	-	-
9	-	GND	-	-
10	-	GND	-	-
11	-	P3V3	-	-
12	-	P3V3	-	-

2.14 Board errata

- Incorrect device type - Boards with devices marked "PMCXN947" may report an incorrect device type in the SYSCON->DEVICE_TYPE field.
- Erroneous HVD assertion - Boards with devices marked "PMCXN947" may assert HVD events if VDD is greater than 3.5 V. This applies only to configurations that supply VDD externally.
- EdgeLock 2GO service not provisioned - Boards with devices marked "PMCXN947" do not have the proper provisioning for the EdgeLock 2GO service. If this service is required, contact your local field applications engineer (FAE) or sales representative for assistance.
- Wake-up times may be faster than expected - Boards with devices marked "PMCXN947" may exhibit faster wake-up times than qualified "MCXN947" devices.
- Incorrect LDO_SYS output capacitance - Boards with devices marked "PMCXN947" do not have the required 0.9 µF - 2.1 µF of capacitance. As a result, a false HVD / LVD event may occur. No other functional impacts are known.

2.15 Board operating conditions

The operating temperature range for the FRDM-MCXN947 board is -40 ℃ to +105 ℃. The MCX N94x device supports up to 105 ℃. See *MCX N94x, N54x Product Family Data Sheet* for more details on device operating conditions.

3 MCU-Link OB debug probe

This section describes the MCU-Link onboard (OB) debug probe, its features, how to install software support for it, and how to update its firmware.

3.1 MCU-Link overview

MCU-Link is a debug probe architecture jointly developed by NXP and Embedded Artists. The MCU-Link architecture is based on the LPC55S69 MCU, which is based on the Arm Cortex-M33 core.

The MCU-Link architecture is configurable to support different debug feature options, and to support both standalone probes (such as MCU-Link Pro) and for use on-board evaluation boards such as FRDM-MCXN947. These on-board implementations are referred to as MCU-Link OB.

The FRDM-MCXN947 board implements a subset of the MCU-Link architecture features, as described in Section 3.2. For more information on MCU-Link visit MCU-Link Debug Probe Architecture.

The MCU-Link OB on the FRDM-MCXN947 board is factory programmed with the firmware based on the NXP CMSIS-DAP protocol. The firmware also supports all other features supported in the hardware. A custom version of the J-Link firmware to make MCU-Link OB compatible with J-Link LITE is also available. However, this firmware version supports only limited features, including debug/SWO and VCOM. For information on how to update the firmware, see Section 3.5.

3.2 Supported MCU-Link features

MCU-Link includes several mandatory and optional features. Table 26 summarizes the MCU-Link features supported on the FRDM-MCXN947 board.

Table 26. Supported MCU-Link features

Feature	Description
Serial wire debug (SWD) / serial wire debug trace output (SWO)	Allows SWD-based debugging with SWO for profiling and/or low overhead debug standard I/O communication
Virtual communication (VCOM) serial port	Adds a serial COM port on the host computer, and connects it to the target MCU by using MCU-Link as a USB-to-UART bridge
USB serial input/output (USBSIO)[1]	Adds a USB serial I/O port on the host computer, and connects it to the target MCU by using MCU-Link as a USB-to-SPI bridge or USB-to-I2C bridge
External debug probe support	Allows debugging the target MCU (MCXN947) using an external debug probe, instead of MCU-Link. Support for an external debug probe is enabled by disabling the SWD feature. While using an external debug probe, the VCOM and USBSIO features can be used.
External target support[1]	Allows debugging an external target MCU using MCU-Link

[1] J-Link firmware does not support this feature.

3.3 Supported debug scenarios

In the FRDM-MCXN947 board, the MCU-Link debug probe target can be either the MCXN947 MCU or an external target compliant with MCU-Link. The board also allows to use an external debugger for debugging the MCXN947 MCU, in place of the MCU-Link debug probe.

Table 27 describes the debug scenarios supported on the FRDM-MCXN947 board.

Table 27. Supported debug scenarios

Debug scenario	Feature support	Jumper / connector settings
Use MCU-Link as a debugger for the target MCU (MCXN947)	• SWD is enabled • VCOM is enabled • USBSIO is enabled	• J19 must be open • J18 must be open

Table 27. Supported debug scenarios*...continued*

Debug scenario	Feature support	Jumper / connector settings
Use an external debugger to debug the target MCU (MCX N94X)	• SWD is disabled • VCOM is enabled • USBSIO is enabled	• J19 must be shorted • J18 must be open • Connect an external debugger to the target MCU SWD connector J23
Use MCU-Link as a debugger for an external target MCU	• SWD is enabled • VCOM is not supported • USBSIO is not supported	• J19 must be open • J18 must be shorted • Connect an external target MCU to the target MCU SWD connector J23

3.4 MCU-Link host driver and utility installation

The MCU debug probe is supported on Windows 10/11, MacOS X, and Ubuntu Linux platforms. The probe uses standard OS drivers. For Windows, the installation program also includes information files to provide user-friendly device names.

MCU-Link is supported by the Linkserver utility. Running the Linkserver installer also installs all the drivers and a firmware update utility required for MCU-Link. The Linkserver utility is a GDB server and flash utility from NXP with support for many NXP debug probes. You are recommended to use the Linkserver installer unless you are using MCUXpresso IDE version 11.6.1 or earlier. For more details on this utility, refer https://nxp.com/linkserver.

Note: *Installing the LinkserServer utility (using the Linkserver installer) only installs the required device drivers. LinkServer does not update the firmware, but the LinkerServer installation package includes the utilities that are used to update the firmware. In case you are using MCUXpresso IDE version 11.6.1 or earlier, you must install the firmware update utility version 2.263, which is not included in the LinkServer installation.*

If you do not use the Linkserver utility, follow the steps below to install the firmware MCU-Link update utility and information files:

1. Visit the board page on the NXP website.
2. Go to the Design Resources > Software section. Under the Development Software category, MCU-Link installation packages for Windows, MacOS, and Linux platforms are available.
3. Download the MCU-Link installation package applicable to your host OS.
4. Run the installer program (for Windows) or install the firmware package (for MacOS or Linux). It is recommended to update to the latest version of the firmware as it might have changed since your MCU-Link was manufactured.

Before updating the firmware by using the steps listed in Section 3.5, check the compatibility between the MCU-Link firmware and the MCUXpresso IDE. Table 28 describes the compatibility between the MCU-Link firmware and the MCUXpresso IDE.

Table 28. Compatibility between MCU-Link firmware and MCUXpresso IDE

MCU-Link firmware version	USB driver type	CMSIS-SWO support	FreeMASTER support via		Supported MCUXpresso IDE versions
			SWD / JTAG	USB bridge	
V1.xxx and V2.xxx	HID	No	Yes	Yes	MCUXpresso 11.3 or later
V3.xxx (up to and including V3.108)	WinUSB	No	Yes	FreeMASTER V3.2.2 or later	MCUXpresso 11.7.0 or later

Table 28. Compatibility between MCU-Link firmware and MCUXpresso IDE*...continued*

MCU-Link firmware version	USB driver type	CMSIS-SWO support	FreeMASTER support via		Supported MCUXpresso IDE versions
			SWD / JTAG	USB bridge	
V3.117 and later	WinUSB	Yes	Yes	FreeMASTER V3.2.2 or later	MCUXpresso 11.7.1 or later

3.5 Updating MCU-Link firmware

When updating the firmware, MCU-Link must be powered up in ISP mode. Follow these steps to configure MCU-Link in ISP mode and update MCU-Link firmware.

1. Disconnect the board from the host computer, short jumper J21, and reconnect the board. The red MCU-Link status D11 LED lights up and stays on. For the D10 LED details, see Section 3.10.
2. Navigate to the `MCU-LINK_installer_Vx_xxx` directory, where `Vx_xxx` indicates the version number, for example, V3.108.
3. Follow the instructions in the `readme.txt` to find and run the firmware update utilities for CMSIS-DAP or J-Link versions.
4. Disconnect the board from the host computer, open jumper J21, and reconnect the board. The board enumerates on the host computer as a WinUSB or HID device (depending on the firmware version).

Note:

- *Starting version V3.xxx, the MCU-Link firmware uses WinUSB instead of HID for higher performance; however, it is not compatible with MCUXpresso IDE versions earlier than 11.7.0.*
- *To enable SWO-related features in non-NXP IDEs, CMSIS-SWO support was introduced in firmware version V3.117.*

3.6 Using MCU-Link with development tools

The MCU-Link debug probe can be used with IDEs supported within the MCUXpresso ecosystem, such as MCUXpresso IDE, MCUXpresso for Visual Studio Code, IAR Embedded Workbench, and Arm Keil MDK.

3.6.1 Using MCU-Link with MCUXpresso IDE

The MCUXpresso IDE recognizes any type of MCU-Link probe that uses either CMSIS-DAP or J-Link firmware. When you start a new debug session, the IDE checks for all the available debug probes. For all the probes it finds, the IDE displays the probe types and unique identifiers in the **Probes discovered** dialog box.

If a debug probe requires a firmware update, the probe is displayed with a warning in the **Probes discovered** dialog box. For each such probe, the latest firmware version is indicated and a link to download the latest firmware package is provided. To update the firmware for the MCU-Link debug probe, see the instructions provided in Section 3.5.

You are advised to use the latest MCU-Link firmware to take the benefit of the latest functionality. However, the MCU-Link firmware version you can use depends on the MCUXpresso IDE version you are using. Table 28 shows the compatibility between the MCU-Link firmware and the MCUXpresso IDE.

3.6.2 Using MCU-Link with MCUXpresso for Visual Studio Code

The MCU-Link debug probe can be used with the MCUXpresso for Visual Studio Code extension from NXP. This extension uses the Linkserver debug server. To work with MCUXpresso for Visual Studio Code, install the Linkserver utility using the MCUXpresso Installer tool or as described in Section 3.4. For more details on MCUXpresso for Visual Studio Code, visit the MCUXpresso for Visual Studio Code page.

3.6.3 Using MCU-Link with third-party IDEs

The MCU-Link debug probe can be used with IAR Embedded Workbench and Arm Keil MDK, and may also work with other third-party tools. Refer to the documentation for these products, covering the use of generic CMSIS-DAP probes or J-Link probes (depending on the firmware image you are using.)

3.7 MCU-Link USB connector

The FRDM-MCXN947 board has a universal serial bus (USB) 2.0 Type-C connector (J17). This USB connector is used to create MCU-Link high-speed USB connection with the host computer. The MCU-Link receives power when the USB connector (J17) is plugged into a USB host.

3.8 Connecting to a target through a USB-to-UART bridge

The MCU-Link supports the VCOM serial port feature, which adds a serial COM port on the host computer, and connects it to the target MCU by using MCU-Link as a USB-to-UART bridge.

On the FRDM-MCXN947 board, MCU-Link LPC55S69 is connected to the P1_8 and P1_9 pins of the target MCU through the R173 and R172 resistors, respectively.

Note: *The P1_8 and P1_9 pins are also the default UART ISP pins to allow for ISP connection through the MCU-Link VCOM.*

To use MCU-Link as a USB-to-UART bridge, ensure that the J18 jumper is open and connect the J17 connector on the board to the USB port of the host computer.

When you boot the FRDM-MCXN947 board, a VCOM port with the name MCU-Link Vcom Port (COMxx) is enumerated on the host computer, where “xx” may vary from one computer to another. Each MCU-Link based board has a unique VCOM number associated with it.

3.9 Connecting to a target through a USB-to-SPI or USB-to-I2C bridge

MCU-Link supports the USB serial input/output (USBSIO) port feature, which adds a USB serial I/O port on the host computer, and connects it to the target MCU by using MCU-Link as a USB-to-SPI bridge or USB-to-I2C bridge. Support for the USBSIO feature can be enabled on the host computer using the libusbsio library, which is a free host library from NXP for Windows/Linux/MacOS systems. For more details on the libusbsio library, see https://www.nxp.com/libusbsio.

In the FRDM-MCXN947 board, MCU-Link connects to the P1_[3:0] pins of the target MCU using the FC1 SPI interface connection, through zero-ohm resistors (DNP by default). Populating these resistors enables the communication between MCU-Link and the target MCU through the USB-to-SPI bridge.

The SPI interface connections for this functionality are shared with the SPI connections on the Arduino compatible connectors and Mikroe connector connections. To prevent contention with these connectors, zero-ohm resistors are used to isolate the connections from the MCU-Link circuit by default.

A USB-to-SPI bridge can be used to emulate the host system. To use MCU-Link as a USB-to-SPI bridge, the board must be connected to the host computer through a USB cable from its J17 connector. Also, ensure the following resistor configuration on the board to enable the USBSIO bridge feature for SPI:

- Resistors R136, R137, R138, and R139 are populated
- Resistor R121 is DNP (default setting)
- Resistor R156 is unpopulated

On the FRDM-MCXN947 board, MCU-Link is also connected to the P0_[17:16] pins of the target MCU using the FC2 I2C interface connection through zero-ohm resistors (DNP by default). Populating these resistors enables the communication between MCU-Link and the target MCU through the USB-to-I2C bridge.

A USB-to-I2C bridge can be used to emulate the host system / board peripherals. To use MCU-Link as a USB-to-I2C bridge, the board must be connected to the host computer through a USB cable from its J17 connector. Also, ensure the following resistor configuration on the board to enable the USBSIO bridge feature for I2C:

- Zero-ohm resistors R142 and R143 are populated
- Resistor R121 is DNP (default setting)
- 2.2 kΩ resistors R140 and R141 should be populated

3.10 MCU-Link status LEDs

The FRDM-MCXN947 board has three status indicator LEDs for MCU-Link. Table 29 lists these LEDs and describes how each LED behaves in different MCU-Link modes. These LEDs are shown in Figure 3.

Table 29. MCU-Link LEDs

Part identifier	LED name / color	MCU-Link mode		
		Normal mode (with CMSIS-DAP firmware)	Normal mode (with J-Link firmware)	Firmware update (ISP) mode
D10	USB COMM / green	Lights up after successful USB enumeration at startup. Afterward, the LED stays ON.	Remains OFF	Remains OFF
D11	Status / red	Indicates heartbeat (fades in/out repeatedly), with SWD activity overlaid. The LED blinks rapidly at startup, if an error occurs.	Remains OFF	Lights up when MCU-Link target (LPC55S69) boots in ISP mode
D12	VCOM ACT / green	Indicates if VCOM port is transmitting/receiving data	Lights up when MCU-Link boots, and blinks when debug activity happens	Remains OFF

4 Related documentation

Table 30 lists and explains the additional documents and resources that you can refer to for more information on the FRDM-MCXN947 board. Some of the documents listed below may be available only under a non-disclosure agreement (NDA). To request access to these documents, contact your local field applications engineer (FAE) or sales representative.

Table 30. Related documentation

Document	Description	Link / how to access
MCX N94x, N54x Product Family Data Sheet	It provides information about electrical characteristics, hardware design considerations, and ordering information	MCXNx4x.pdf
MCX Nx4x Reference Manual	It is intended for the board-level product designers and product software developers who want to develop products with MCX Nx4x MCU	MCXNx4xRM.pdf
MCX Nx4x Chip Errata (MCXNx4x_xP02G)	Lists the details of all known silicon errata for the MCX Nx4x device.	Contact NXP FAE or sales representative

Table 30. Related documentation*...continued*

Document	Description	Link / how to access
FRDM-MCXN947 design file	A zip file including *.DSN, ASY, Layout, schematic files, and so on	Contact NXP FAE or sales representative
LPC55S6x/LPC55S2x/LPC552x User manual (UM11126)	Intended for system software and hardware developers and application programmers who want to develop products with LPC55S6x/ LPC55S2x/LPC552x MCU	UM11126.pdf

5 Acronyms

Table 31 lists and defines the acronyms used in this document.

Table 31. Acronyms

Term	Description
ADC	Analog-to-digital converter
CAN	Controller area network
DNP	Do not populate
ESR	Equivalent series resistor
GPIO	General-purpose input/output
I2C	Inter-integrated circuit
I3C	Improved inter-integrated circuit
ISP	In-system programming
LPI2C	Low-power inter-integrated circuit
PCB	Printed-circuit board
PHY	Physical interface of the OSI model
PMIC	Power management integrated circuit
POR	Power-on reset
PWM	Pulse width modulation
QSPI	Quadruple serial peripheral interface
RGMII	Reduced gigabit media independent interface
RTC	Real-time clock
SDHC	Secured digital host controller
SPI	Serial peripheral interface
SWD	Serial wire debug
SWO	Serial wire debug trace output
UART	Universal asynchronous receiver/transmitter
USB	Universal serial bus
USBSIO	USB serial input/output
VCOM	Virtual communication
WUU	Wake-up unit

6 Revision history

Table 32 summarizes the revisions to this document.

Table 32. Revision history

Document ID	Release date	Description
UM12018 v.1	20 January 2024	Initial public release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Amazon Web Services, AWS, the Powered by AWS logo, and FreeRTOS — are trademarks of Amazon.com, Inc. or its affiliates.

Contents

For more information, please visit: https://www.nxp.com

Date of release: 20 January 2024
Document identifier: UM12018