UM12036

MCX-N9XX-EVK Board User Manual

Rev. 2 — 10 July 2024 User manual

Document information

Information	Content
Keywords	MCX-N9XX-EVK, UM12036, MCX N94x
Abstract	The NXP MCX N9XX Evaluation Kit (MCX-N9XX-EVK) board is a full-featured evaluation and development board for application prototyping and demonstration of the MCX N94x family of devices. This document describes the hardware for the MCX-N9XX-EVK development board.

1 MCX-N9XX-EVK overview

The NXP MCX N9XX Evaluation Kit (MCX-N9XX-EVK) board is a full-featured evaluation and development board for application prototyping and demonstration of the MCX N94x family of devices.

The MCX N94X integrates a dual Arm Cortex-M33 MCU and a neural processing unit (NPU) into a single package. NXP supports MCX N94X with tools and software that include hardware evaluation boards, software development IDE, example applications, and drivers.

The MCX-N9XX-EVK development board consists of the MCX N94X device with a 64-Mbit external serial flash (provided by Winbond), FXLS8964AF accelerometer, P3T1755DP I3C temperature sensor, visible light sensor, onboard CAN PHY, Ethernet PHY, SDHC circuit, general-purpose RGB LED, touch slider, FS and HS USB circuits, general-purpose push buttons, and onboard MCU-Link debug probe circuit with energy monitoring. The board is compatible with the Arduino and FRDM ecosystem shield modules and Mikroe click boards.

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.

The board is a standalone PCB and supports application development with the NXP SDK software package.

Note:

- *The MCX N94x represents the MCX N947 and MCX N946 devices in this document.*
- *For simplicity, the MCU-Link debug probe is referred to as "MCU-Link" and the MCX N94X MCU is referred to as "target MCU" in this document.*

1.1 Block diagram

Figure 1 shows the MCX-N9XX-EVK block diagram.

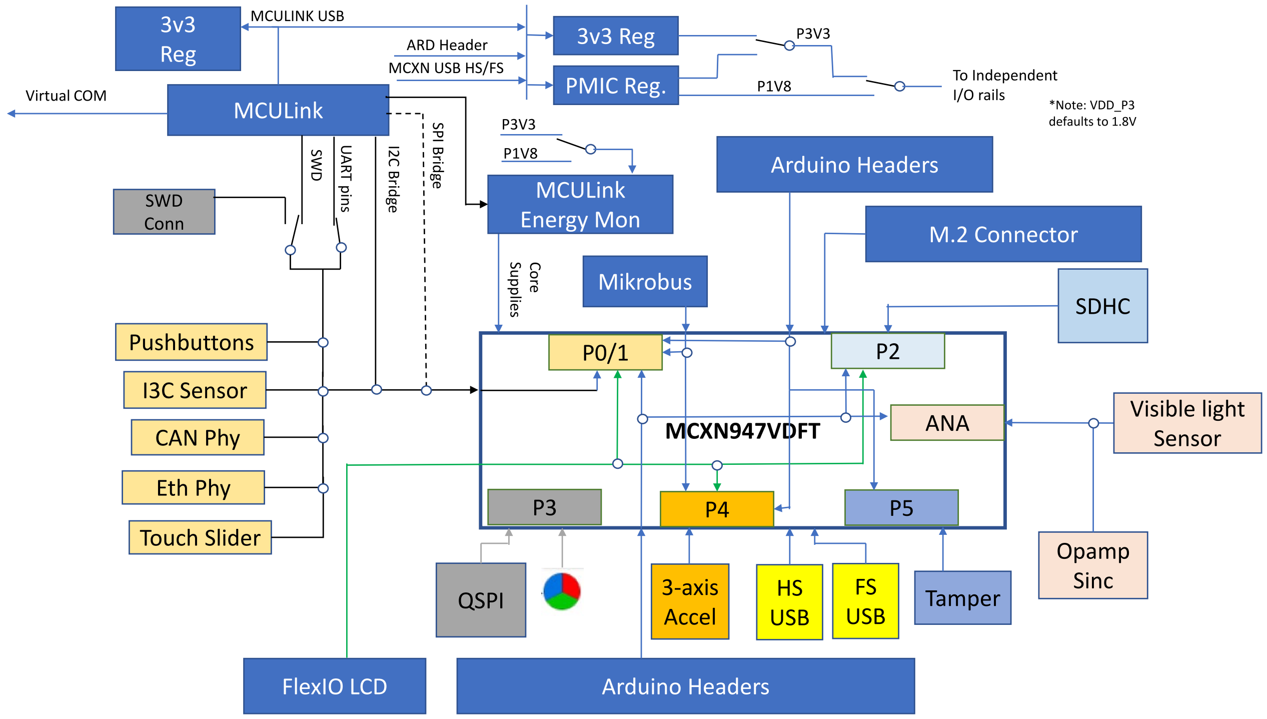


Figure 1. MCX-N9XX-EVK block diagram

1.2 Board features

Table 1 describes the features of the MCX-N9XX-EVK board.

Table 1. MCX-N9XX-EVK features

Board feature	Target MCU features used	Description
MCX N94x MCU (target MCU)		The MCX N94x is based on dual high-performance Arm Cortex-M33 cores running up to 150 MHz, with 2 MB of Flash with optional 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: – MCU-Link USB micro-B connector (default source) – Full-speed (FS) micro USB connector – High-speed (HS) micro USB connector – Arduino Shield compatible header – 5 V regulator populated at 3-pin jumper • A 10-pin jumper (J26) is used for the 5 V power source selection • PCA9420 PMIC for flexible voltage selections • Jumpers for different power sources selection
Clock		Two crystal oscillators for: • 24 MHz system reference clock • 32.768 kHz real-time clock (RTC)
USB	USB full-speed (FS) and high-speed (HS) modules	• One USB Micro-AB connector interfaced with USB HS host / device mode module • One USB Micro-AB connector interfaced with USB FS host / device mode module
Flash memory	FlexSPI controller	One 1.8 V 64-Mbit (8 MB) serial flash memory with dual and quad SPI
3D Accelerometer	Low-power inter-integrated circuit (LPI2C) module	Supports NXP FXLS8964AF accelerometer for motion sensing
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 • Mikroe click boards
Light sensor	ADC	One ambient light sensor, consisting of a phototransistor in a miniature SMD
Wi-Fi / Bluetooth interface	SDIO, I2S, UART, I2C, and GPIO	• One M.2 Key E mini card 75-pin connector supporting SDIO, I2S, UART, I2C, and Vendor-defined SPI interfaces • Compatible with Murata IW416 Wi-Fi card module
CAN	FlexCAN controller	One 4-pin CAN FD connector
Mass storage	uSDHC	MicroSD card connector
Touch slider	Touch sensing input (TSI)	One Touch Slider for touch sensing detection; connects to TSI input channels through the P1_10 and P1_12 pins

Table 1. MCX-N9XX-EVK features*...continued*

Board feature	Target MCU features used	Description
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
PCB		6” 3/4 x 4” 5/16 x 1/16”
Orderable part number		MCX-N9XX-EVK

1.3 Board kit contents

The MCX-N9XX-EVK board kit contains the following items:

- MCX-N9XX-EVK board hardware assembly
- A 3 ft micro USB A to micro USB B cable
- Nylon screws (4-40 x ¼”)
- Nylon standoffs (4-40 x 1.25, 3/16 Hex)

1.4 Board pictures

Figure 2 shows the top view of MCX-N9XX-EVK.

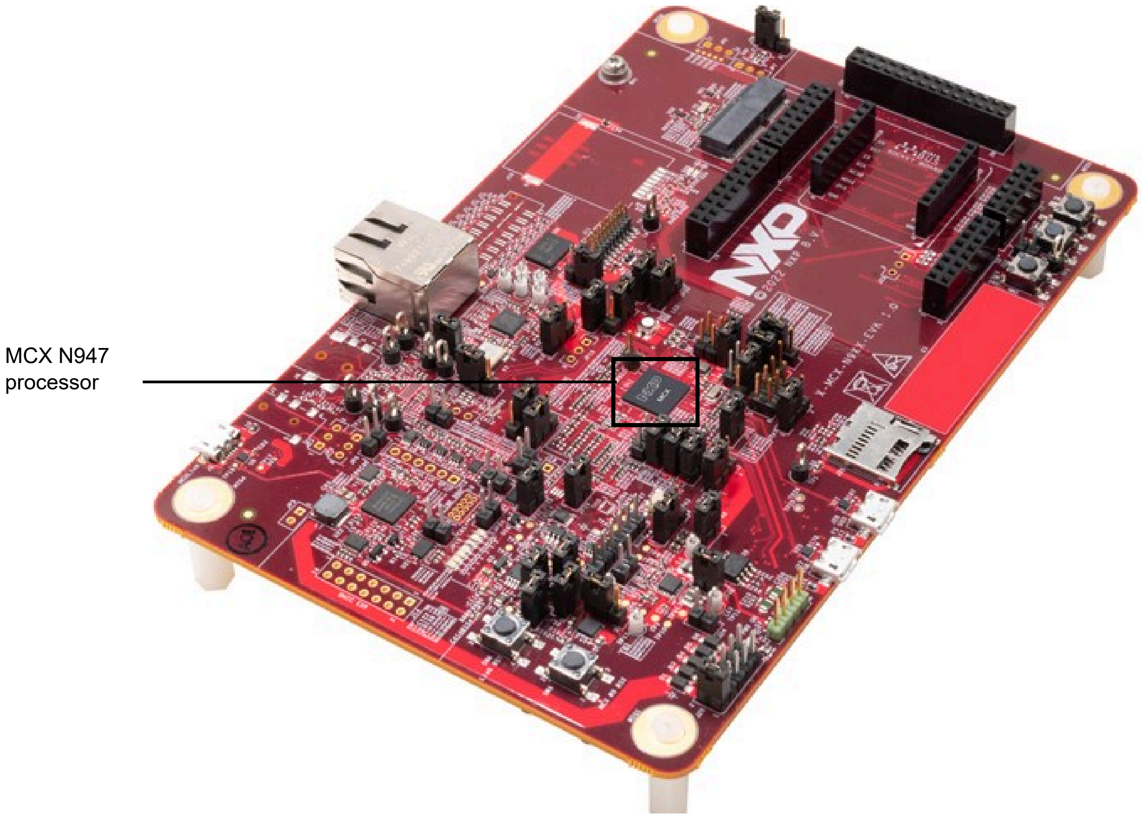


Figure 2. MCX-N9XX-EVK top view

Figure 3 shows the top-side view of the MCX-N9XX-EVK board, with connectors, push buttons, and LEDs highlighted.

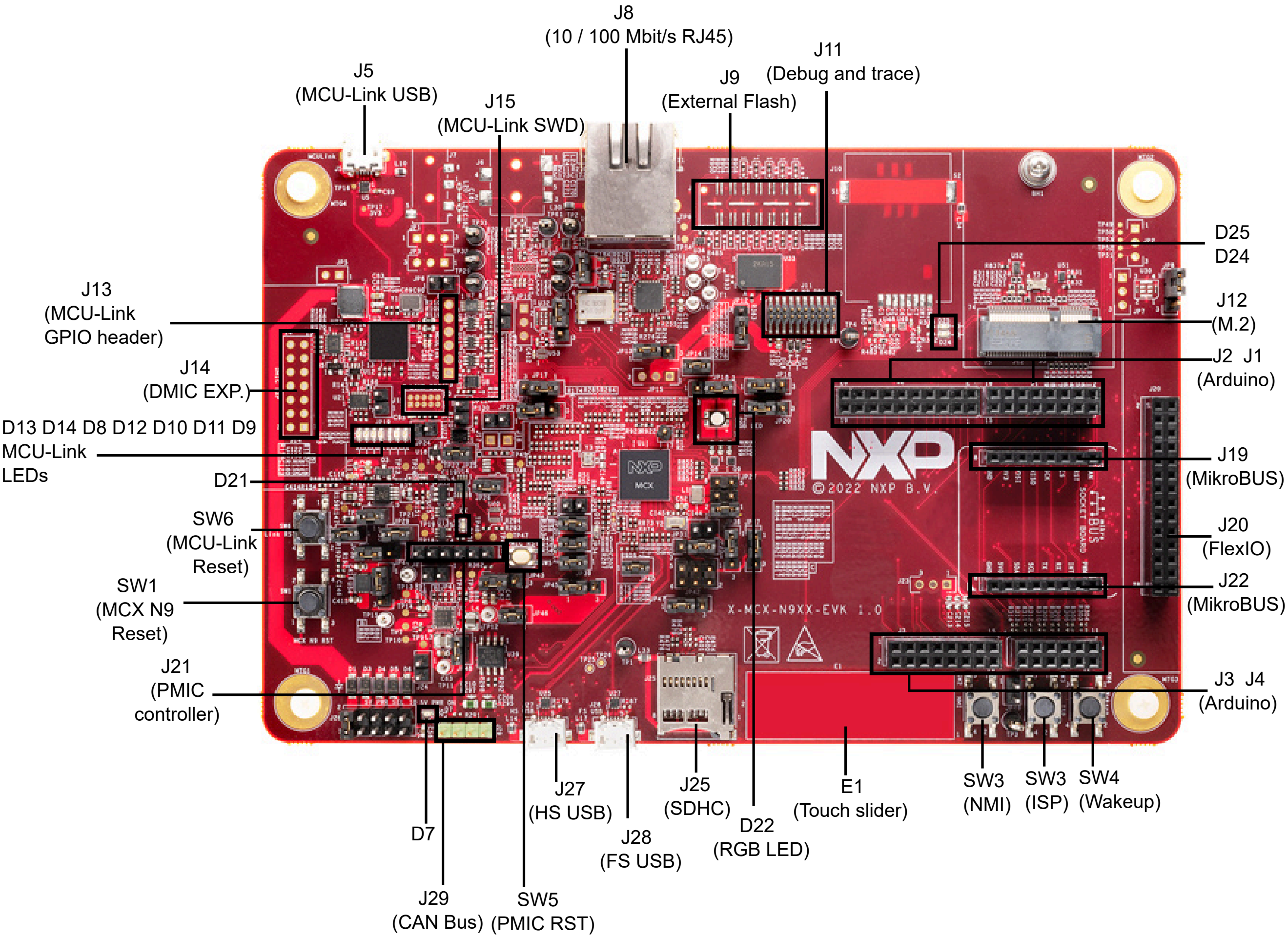


Figure 3. MCX-N9XX-EVK connectors, push buttons, and LEDs (top-side view)

Figure 4 shows the jumpers of the MCX-N9XX-EVK board.

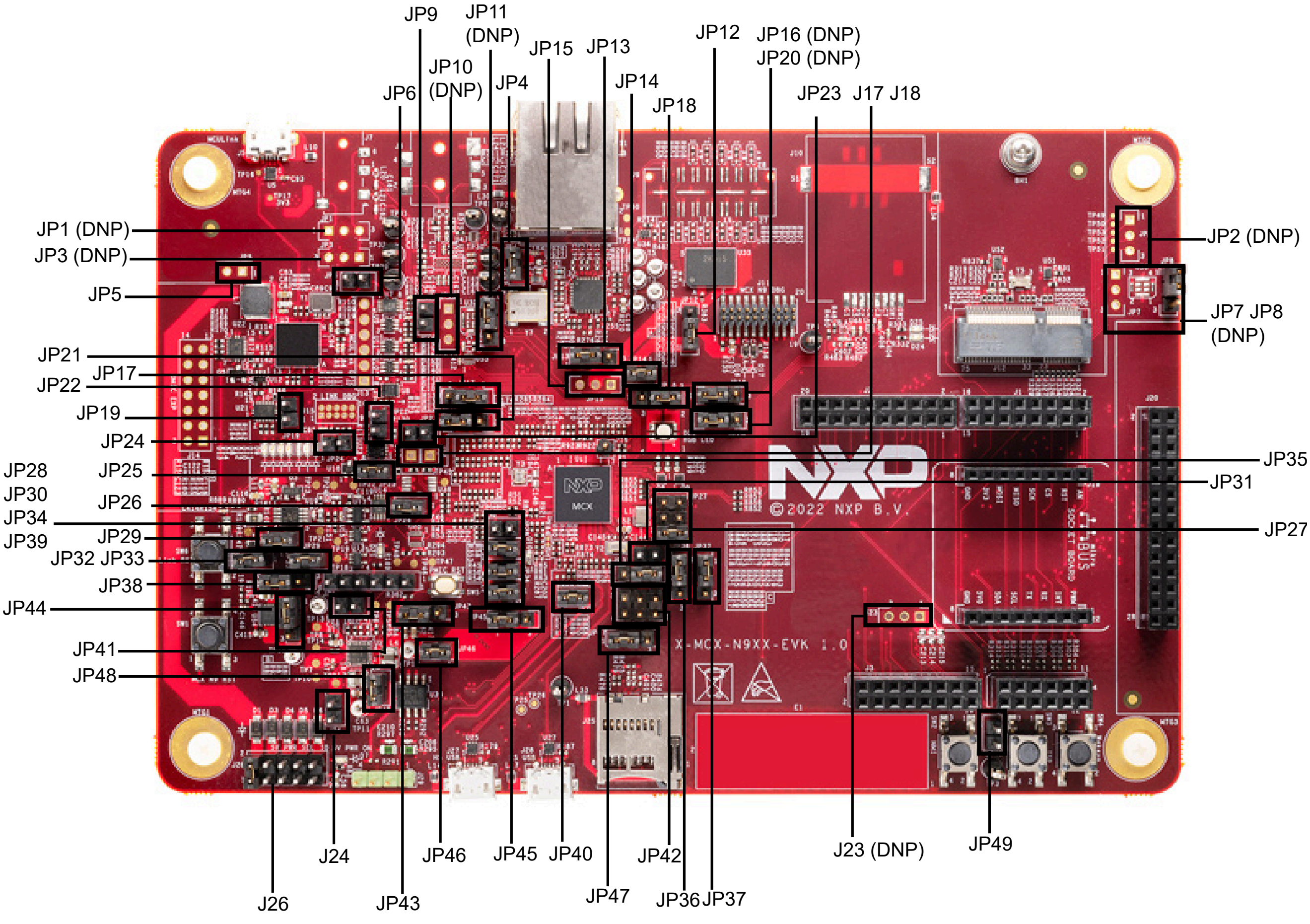


Figure 4. MCX-N9XX-EVK jumpers

1.5 Connectors

Table 2 describes the MCX-N9XX-EVK connectors. The connectors are shown in Figure 3.

Table 2. MCX-N9XX-EVK connectors

Part identifier	Connector type	Description	Reference section
BT1	Coincell battery holder	Holder for 3 V CR2025 or CR 2032 lithium coincell battery required for PMIC Battery backup (VBAT_BKUP) ***Note:*** *This holder is at the bottom of the board underneath the MCU RST button (SW1).*	For more information on this connector, see MCX-N9XX-EVK schematics
J24	1 x 2 pin header	Lithium-ion/external battery connector	
J1	2 x 8 pin header	Arduino compatible I/O header	Section 2.12
J2	2 x 10 pin header		
J3	2 x 8 pin header		
J4	2 x 6 pin header		
J5	USB connector	MCU-Link USB connector	Section 3.7

Table 2. MCX-N9XX-EVK connectors*...continued*

Part identifier	Connector type	Description	Reference section
J8	RJ45 connector	Shielded RJ45 connector jack with magnetic built-in	Section 2.6
J9 (DNP)	2 x 14 pin header	To connect external flash daughter card	Section 2.5
J11	2 x 10 pin header	Debug (JTAG / SWD) and trace connector to connect an external debug probe or external debug target	Section 3.8
J12	75-pin connector	M.2 Key E module connector	Section 2.11
J13	1 x 6 pin header	MCU-Link GPIO header	Section 3.12
J15 (DNP)	2 x 5 pin header	MCU-Link SWD connector	For more information on this connector, see MCX-N9XX-EVK schematics
J16	1 x 1 header	To monitor CLKOUT clock output on P0_6 ***Note:*** *The bypass capacitor, C150, affects the clock output (CLKOUT) performance. To use the clock output function, the C150 capacitor must be removed.*	For more information on this connector, see MCX-N9XX-EVK schematics
J19	1 x 8 position receptacles	mikroBUS socket connectors	Section 2.14
J20	2 x 14 pin header	FlexIO header for LCD screen connection	Section 2.13
J21	1 x 6 pin header	To connect a PMIC controller for regulating PMIC (U2) regulator outputs	For more information on this connector, see MCX-N9XX-EVK schematics
J22	1 x 8 position receptacles	mikroBUS socket connectors	Section 2.14
J25	SDHC connector	To connect an external SD card	Section 2.4
J27	USB micro-AB connector	USB1 high-speed connector	Section 2.3
J28	USB micro-AB connector	USB0 full-speed connector	Section 2.3
J29	1 x 4 pin header	Connects to the CAN bus and allows external connection with the bus	Section 2.9
JP42	2 x 3 pin header	Used for tamper different mode test	Section 2.15

1.6 Jumpers

Table 3 describes the MCX-N9XX-EVK jumpers. The jumpers are shown in Figure 4.

Table 3. MCX-N9XX-EVK jumpers

Part identifier	Jumper type	Description	Reference section
J26	2x5 pin header	P5V0 input power selection • Pin 1-2 shorted (**default setting**): P5V_MCU_LINK_USB supply from MCU-Link USB micro-B connector (J5) • Pin 3-4 shorted: P5V_HDR_IN from 5 V regulator populated at 3-pin jumper (J23) • Pin 7-8 shorted: VREG_IN_USB1 supply from HS micro USB connector (J27)	Section 2.1

Table 3. MCX-N9XX-EVK jumpers*...continued*

Part identifier	Jumper type	Description	Reference section
		• Pin 9-10 shorted: VREG_IN_USB0 supply from FS micro USB connector (J28)	
JP4	1x2 pin header	• Pin 1-2 open: Disconnects the P3V3 supply from the LAN8741 Ethernet PHY circuit • Pin 1-2 shorted (**default setting**): Connects the P3V3 supply to the VDD analog pins of the LAN8741 Ethernet PHY	Section 2.6
JP5	1x2 pin header	• Open (**default setting**): Enables the onboard MCU-Link USBSIO bridge feature for SPI • Shorted: Sends a low signal on HW_VER_2 to disable the onboard MCU-Link USBSIO bridge feature for SPI	Section 3.9
JP6	1x2 pin header	• Open (**default setting**): Enables the MCU-Link SWD feature • Shorted: Sends a low signal on 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
JP9	1x2 pin header	• Open (**default setting**): No JTAG connection • Shorted: Connects TDI signal to P0_3 port for JTAG connection	For more information on this jumper, see MCX-N9XX-EVK schematic
JP12	1x3 pin header	Target power selection jumper • 1-2 shorted (**default setting**): Onboard target MCU is used as a debug target or an external target MCU is used as a debug target but it uses board power • 2-3 shorted: An external target MCU is used as a debug target and it uses its own power	Section 3.3
JP13	1x3 pin header	• Pin 1-2 shorted (**default setting**): Port P1 pin 21 (P1_21) connects to the DA7212 audio codec (DNP) through dual-supply translating transceiver NTS0302 • Pin 2-3 shorted: Port P1 pin 21 connects to the MDIO interface of the LAN8741 Ethernet PHY	For more information on this jumper, see MCX-N9XX-EVK schematic
JP14	1x2 pin header	Pin 1-2 shorted (**default setting**): VDD_MCU is sourced from MCU_PWR supply	For more information on this jumper, see MCX-N9XX-EVK schematic
JP15 (DNP)	1x3 pin header	• Pin 1-2 shorted: VDD_P3 is sourced from the P3V3 power supply • Pin 2-3 shorted (**default setting**): VDD_P3 is sourced from P1V8 supply ***Note:*** *VDD_P3 must be 1.8 V for QSPI flash and SIM card interfaces.*	Section 2.1.1
JP17	1x3 pin header	• Pin 1-2 shorted (**default setting**): Port P0 pin 11 (P0_11) connects to SD connector power switch • Pin 2-3 shorted: Port P0 pin 11 connects to the VBUS controller (NX5P3090UK) of USB FS	Section 2.4
JP18	1x3 pin header	• Pin 1-2 shorted: VDD_LDO_CORE_IN is sourced from MCU_PWR power supply	Section 2.1.1

Table 3. MCX-N9XX-EVK jumpers*...continued*

Part identifier	Jumper type	Description	Reference section
		• Pin 2-3 shorted (**default setting**): VDD_LDO_CORE_IN is sourced from VDD_CORE supply ***Note:*** *Default setting for LDO_CORE disable case.*	
JP19	1x2 pin header	• Open (**default setting**): Enables serial port connection (VCOM) between MCU-Link and the target MCU (connected through pins P1_8 and P1_9) • Shorted: Sends a low signal on HW_VER_6 to disable serial port connection (VCOM) available between the MCU-Link LPC55S69 and the MCX N94X MCU	Section 3.8
JP21	1x3 pin header	• Pin 1-2 shorted (**default setting**): Port P1 pin 11 (P1_11) connects to P3T1755DP temperature sensor if jumper JP23 is shorted • Pin 2-3 shorted: Port P1 pin 11 (P1_11) connects to the FlexIO header	For more information on this jumper, see MCX-N9XX-EVK schematic
JP22	1x2 pin header	MCU-Link debug probe target selection jumper: • Open (**default setting**): The MCU-Link debug probe or an external debug probe can be connected to the onboard target MCU. Use this setting when the target MCU operates in Normal mode (its SWD interface is enabled). • Shorted: The MCU-Link debug probe can be connected to an external target MCU. Use this setting when using the MCU-Link debug probe for debugging an external target MCU.	Section 3.3
JP23	1x2 pin header	• Open (**default setting**): To disable external pull-up resistance on the I3C SDA • Shorted: To enable external pull-up resistance on I3C SDA	Section 2.8
JP24	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.*	Installing device drivers and updating MCU-Link firmware
JP25	1x2 pin header	USB-to-I2C bridge enable/disable jumper • Open: The USB-to-I2C bridge between MCU-Link and the target MCU is disabled • Shorted (**default setting**): MCU-Link is connected to the target MCU through a USB-to-I2C bridge	Section 3.9
JP26	1x2 pin header	Pin 1-2 shorted (**default setting**): VDD_BRD is sourced from MCU_PWR_SRC supply	For more information on this jumper, see MCX-N9XX-EVK schematic

Table 3. MCX-N9XX-EVK jumpers*...continued*

Part identifier	Jumper type	Description	Reference section
JP27	2x3 pin header	• Pin 1-2 shorted (**default setting**): VDD_DCDC is sourced from the MCU_PWR power supply • Pin 3-4 shorted: VDD_DCDC is pulled down to GND • Pin 5-6 shorted: VDD_DCDC is pulled down to GND ***Note:*** *In DC-DC bypass mode, open pin 1-2 and short pins 3-4 and 5-6.*	Section 2.1.1
JP28	1x2 pin header	• Pin 1-2 open (**default setting**): Voltage reference output (VREFO) does not connect to Pin 16 of Arduino connector J1 • Pin 1-2 shorted: VREFO connects to Pin 16 of Arduino connector J1	Section 2.12
JP30	1x2 pin header	Pin 1-2 shorted (**default setting**): VREFH is sourced from VDDA supply	For more information on this jumper, see MCX-N9XX-EVK schematic
JP31	1x2 pin header	Pin 1-2 shorted (**default setting**): VDD_SYS is sourced from VDD_LDO_SYS_IN supply ***Note:*** *This is shorted in LDO_BYPASS mode.*	Section 2.1.1
JP29	1x2 pin header	MCU-LINK current measurement circuit jumpers. Pins 1-2 shorted (**default setting**)	Section 3.10
JP32			
JP33			
JP34	1x2 pin header	• Pin 1-2 open: No input voltage to ambient light sensor • Pin 1-2 shorted (**default setting**): Input voltage VDDA connects to ambient light sensor	Section 2.10
JP35	1x2 pin header[1]	Pin 1-2 shorted (**default setting**): VDD_CORE is sourced from the DCDC_VDDCORE power supply ***Note:*** *VDD_CORE is powered by DCDC, LDO_core is disabled.*	Section 2.1.1
JP36	1x3 pin header	• Pin 1-2 shorted (**default setting**): VDD_LDO_SYS_IN is sourced from MCU_PWR power supply • Pin 2-3 shorted: VDD_LDO_SYS_IN supply is sourced from P1V8 supply	Section 2.1.1
JP37	1x3 pin header	• Pin 1-2 shorted (**default setting**): VDD_BAT is sourced from the MCU_PWR power supply • Pin 2-3 shorted: VDD_BAT is sourced from PMIC LDO1_OUT supply	Section 2.1.1
JP38	1x3 pin header	MCU-LINK current measurement circuit jumper • Pin 1-2 shorted (**default setting**): MCU_PWR and MCU_PWR_SRC are sourced from P3V3 supply • Pin 2-3 shorted: MCU_PWR and MCU_PWR_SRC are sourced from P1V8 supply	Section 3.10
JP39	1x2 pin header	Pin 1-2 shorted (**default setting**): VDDA is sourced from VDD_P4 supply	For more information on this jumper, see MCX-N9XX-EVK schematic

Table 3. MCX-N9XX-EVK jumpers*...continued*

Part identifier	Jumper type	Description	Reference section
JP40	1x2 pin header	Pin 1-2 shorted (**default setting**): VDD_USB is sourced from the P3V3 power supply	For more information on this jumper, see MCX-N9XX-EVK schematic
JP43	1x3 pin header	PMIC reset • Pin 1-2 shorted (**default setting**): PMIC reset is supported through the SW5 push button • Pin 2-3 shorted: PMIC reset is supported through the SW1 (MCU reset) button	Section 2.1
JP44	1x3 pin header	P3V3 source selection • Pin 1-2 shorted (**default setting**): ADM7172 LDO is selected for P3V3 output • Pin 2-3 shorted: PCA9420 PMIC LDO2 is selected for P3V3 output	Section 2.1
JP45	2x3 pin header	• Pin 1-2 shorted (**default setting**): VDD_P4 is sourced from a P3V3 power supply • Pin 2-3 shorted: VDD_P4 is sourced from P1V8 supply ***Note:*** *If VDD_P4 is sourced from P3V3 while MCU_PWR is sourced by P1V8, P3V3 MUST be sourced from PMIC LDO2_OUT.*	Section 2.1.1
JP46, JP41	1x2 pin header	P1V8 source selection • JP46 pin 1-2 shorted (**default setting**): PCA9420 PMIC SW2 is selected for P1V8 output • JP41 pin 1-2 shorted: PCA9420 PMIC LDO1 is selected for P1V8 output	Section 2.1
JP47	1x3 pin header	• Pin 1-2 shorted (**default setting**): VDD_P2 is sourced from a P3V3 power supply • Pin 2-3 shorted: VDD_P2 is sourced from P1V8 supply ***Note:*** *If VDD_P2 is sourced from P3V3 while MCU_PWR is sourced by P1V8, P3V3 MUST be sourced from PMIC LDO2_OUT.*	Section 2.1.1
JP48	1x2 pin header	P1V1 output • Pin 1-2 shorted (**default setting**): PCA9420 PMIC SW1 is selected for P1V1 output	Section 2.1
JP49	1x2 pin header	ISP mode enable/disable • Open (**default setting**): Normal operation • Shorted: ISP mode	Section 1.7
JP50	1x2 pin header	Ethernet PHY clock input selection • Pin 1-2 open: Enables crystal oscillator (Y7) to output 50 MHz clock to Ethernet PHY (LAN8741) • Pin 1-2 shorted (**default setting**): Disables the crystal oscillator (Y7) and allows the target MCU to provide a 50 MHz reference clock to Ethernet PHY (LAN8741) through P1_4/ENET0_TXCLK	Section 2.6
JP51	1x2 pin header	Controls the enabling/disabling of the USB power switch (U26) • Open: Disconnects pull-up resistor from P0_11/USB0_ID	Section 2.3

Table 3. MCX-N9XX-EVK jumpers*...continued*

Part identifier	Jumper type	Description	Reference section
		• Shorted (**default setting**): Connects pull-up resistor to P0_11/USB0_ID for enabling USB FS mode in case P0_11 is used for another function	

[1] JP35 is only a two-pin jumper (even though the footprint is a three pin footprint).

1.7 Push buttons

Tactile buttons are populated on the MCX-N9XX-EVK board for human machine interaction (HMI). Each of the SW[4:2] buttons has a 0.1 μF bypass capacitor for debouncing and pads for external pull-up resistors, if desired.

Table 4 describes the MCX-N9XX-EVK push buttons. The push buttons are shown in Figure 3.

Table 4. MCX-N9XX-EVK push buttons

Part identifier	Switch name	Description
SW1	Reset button (MCX N94X 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 D21 turns ON.
SW2	Non-maskable interrupt (NMI) button	SW2 is designated as the MCX N94X NMI pin (however, the MCX architecture allows for any of the pins to operate as the NMI pin). It also acts as a general-purpose input.
SW3	In-system programming (ISP) mode switch	SW3 is an ISP mode switch (it is a white-colored push button for easy identification) and can also act as a general-purpose input. It is shared with the clock output (clkout) header, J16. ***Note:*** *The bypass capacitor, C150, affects the clock output (clkout) performance. To use the clock output function, the C150 capacitor must be removed.* ***Note:*** *Apart from the SW3 button, ISP mode can also be enabled or disabled through the JP49 jumper. For details, see Section 1.6.*
SW4	Wakeup button	SW4 is a general-purpose input as well as a low-power wake-up unit (WUU) pin. This allows for demonstration of Power Down and Deep Power Down applications that use a push button wakeup.
SW5	PMIC reset	Pressing this button resets the PCA9420 PMIC if pins 1-2 are shorted (default configuration) for the JP43 jumper.
SW6	MCU-Link reset	Pressing this button resets the MCU-Link processor (LPC55S69).

Figure 5 shows the circuit diagrams of the MCX-N9XX-EVK push buttons.

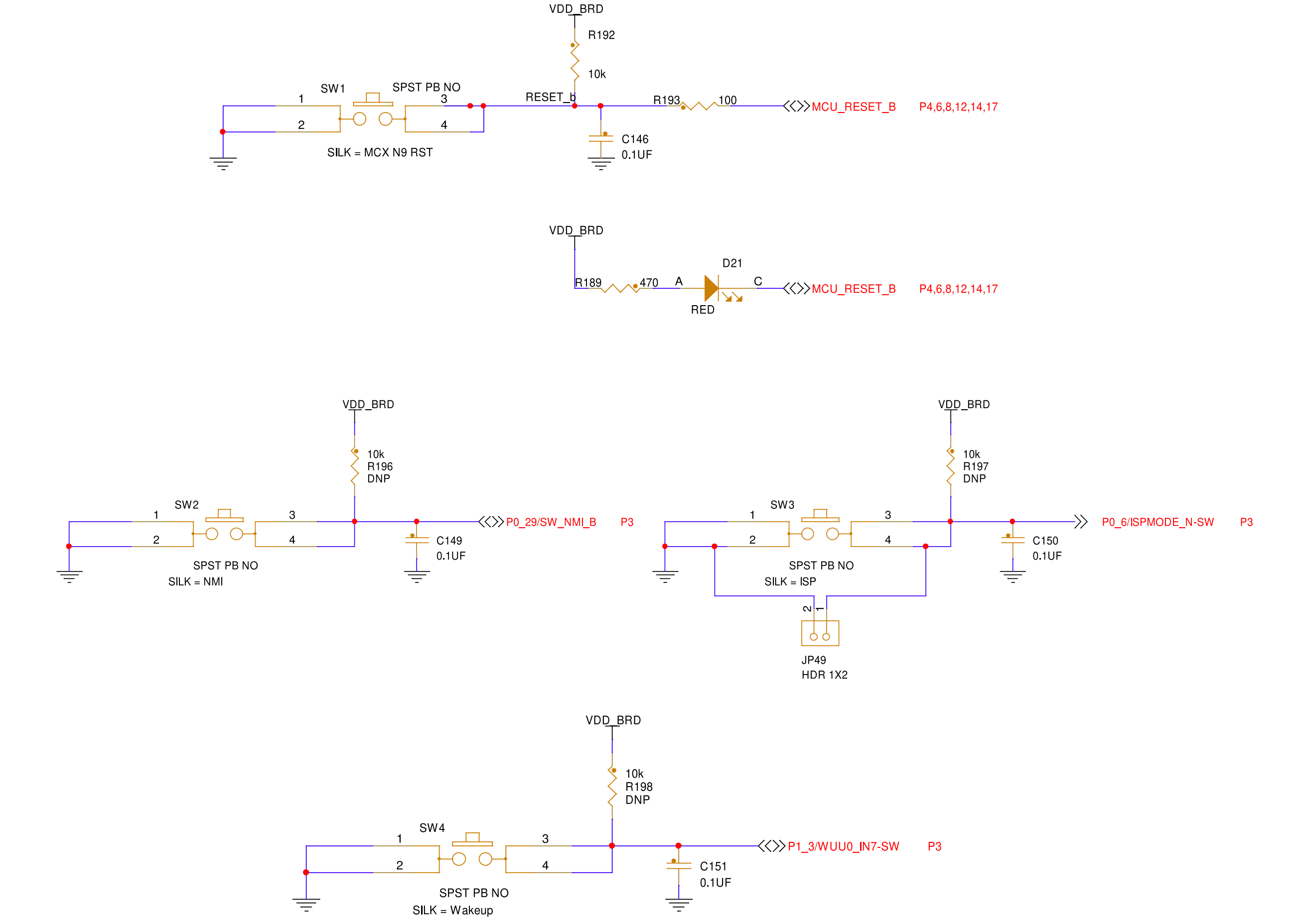


Figure 5. Push button circuit diagram

1.8 LEDs

Table 5 describes the MCX-N9XX-EVK light-emitting diodes (LEDs) that correspond to the target MCU. The board also has some MCU-Link-specific LEDs, which are described in Section 3.11. The LEDs are shown in Figure 3.

Table 5. MCX-N9XX-EVK LEDs

Part identifier	LED color	LED name / function	Description
D22	Red/green/blue	RGB LED	User application LEDs. Each of these LEDs can be controlled through a user application.
D24	RED	Wi-Fi LED	Indicates WLAN status of IW416 Wi-Fi/Bluetooth module
D25	RED	Bluetooth LED	Indicates Bluetooth status of IW416 Wi-Fi/Bluetooth module
D7	Green	5 V PWR ON	Indicates 5 V system power on status. When the board is powered up, D7 turns ON.
D21	Red	Reset LED	Indicates system reset activity. When board reset is initiated, for example, by pressing the SW1 reset button, the D21 LED turns ON.

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

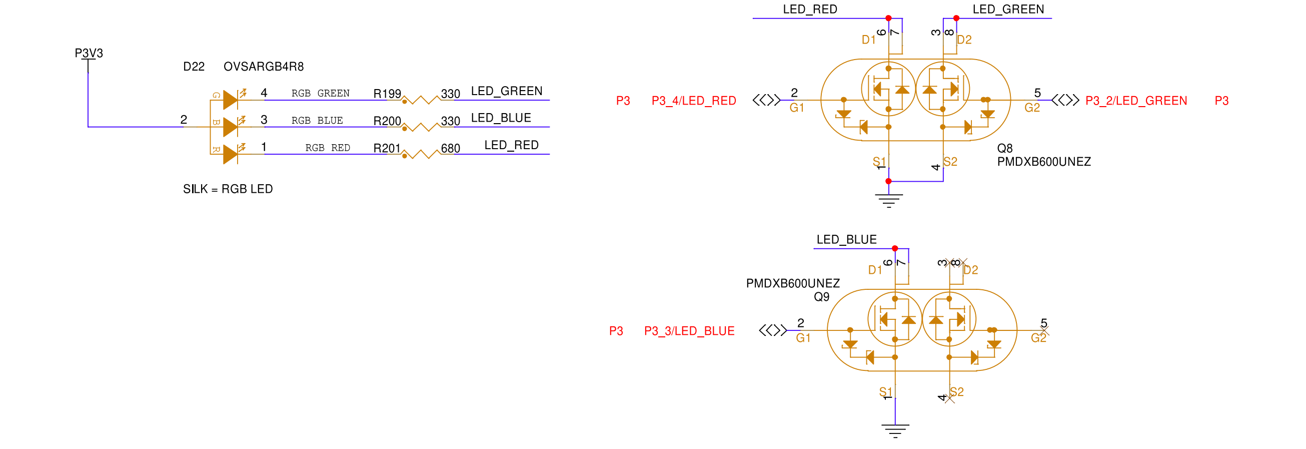


Figure 6. RGB LED circuit diagram

The RGB LEDs are powered by the P3V3 rail. However, these LEDs are not included in current measurements by the MCULink Energy Monitor circuit. These LEDs are controlled by transistors, Q8 and Q9, which allow the LEDs to operate while being controlled by GPIOs that are powered at a voltage less than P3V3.

2 MCX-N9XX-EVK functional description

This section describes the features and functions of the MCX-N9XX-EVK 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 MCX-N9XX MCU, see MCX N94x, N54x Product Family Data Sheet and MCX Nx4x Reference Manual.*

2.1 Power supplies

The MCX-N9XX-EVK board is powered with a P5V0 (5 V) power supply using one of the following source options:

- VREG_IN_USB0 supply from full-speed (FS) USB connector (J28)
- VREG_IN_USB1 supply from high-speed (HS) USB connector (J27)
- P5V_HDR_IN supply from 5 V regulator populated at 3-pin jumper (J23) (Not populated by default)
- P5V0 supply from Arduino Shield compatible header, J3 (pin 10)
- P5V_MCU_LINK_USB supply from MCU-Link USB micro-B connector (J5)

The P5V0 supply is an input power supply for the onboard PCA9420 PMIC (U2) and ADM172 3.3 V regulator (U54), which provides secondary power supplies. The P5V0 supply also powers up the energy monitoring analog circuits, FS / HS USB VBUS controllers, and CAN driver.

Many power supplies in the MCX-N9XX-EVK board are connected through jumpers, which can be used to configure their respective power supplies. This configuration includes enabling/disabling a power supply and changing the input power source for a power supply.

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
J5	MCU-Link USB micro-B connector	P5V_MCU_LINK_USB	• One of the sources of P5V0 (5 V) supply **(default option)**

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

Part identifier	Device / power source	Output power supply	Description
			• USB regulator input power supply for MCU-Link microcontroller LPC55S69
J28	FS USB connector	VREG_IN_USB0	One of the sources for P5V0 (5 V) supply
J27	HS USB connector	VREG_IN_USB1	One of the sources for P5V0 (5 V) supply
J23	5 V power regulator populated at J23	P5V_HDR_IN	One of the sources for P5V0 (5 V) supply
J3 (pin 10)	Arduino shield compatible header	P5V0	• Power supply for: – Onboard PCA9420 PMIC (U2) – ADM172 3.3 V regulator (U54) – mikroBUS connector (J22) – HS and FS USB connectors power switches (U24 and U26) – FlexCAN transceiver TJA1057 (U39) • Power source for: – P5VA analog power supply

A 10-pin jumper (J26) is provided on the board to make the 5 V power source selection. By default, the jumper is set to select a 5 V supply from the MCU-Link USB micro-B connector (J5). For the J26 jumper details, see Section 1.6. The D7 LED indicates that the 5 V power is available on the board. For the D7 LED details, see Section 1.8.

Figure 7 shows the 5 V power selection on the MCX-N9XX-EVK board.

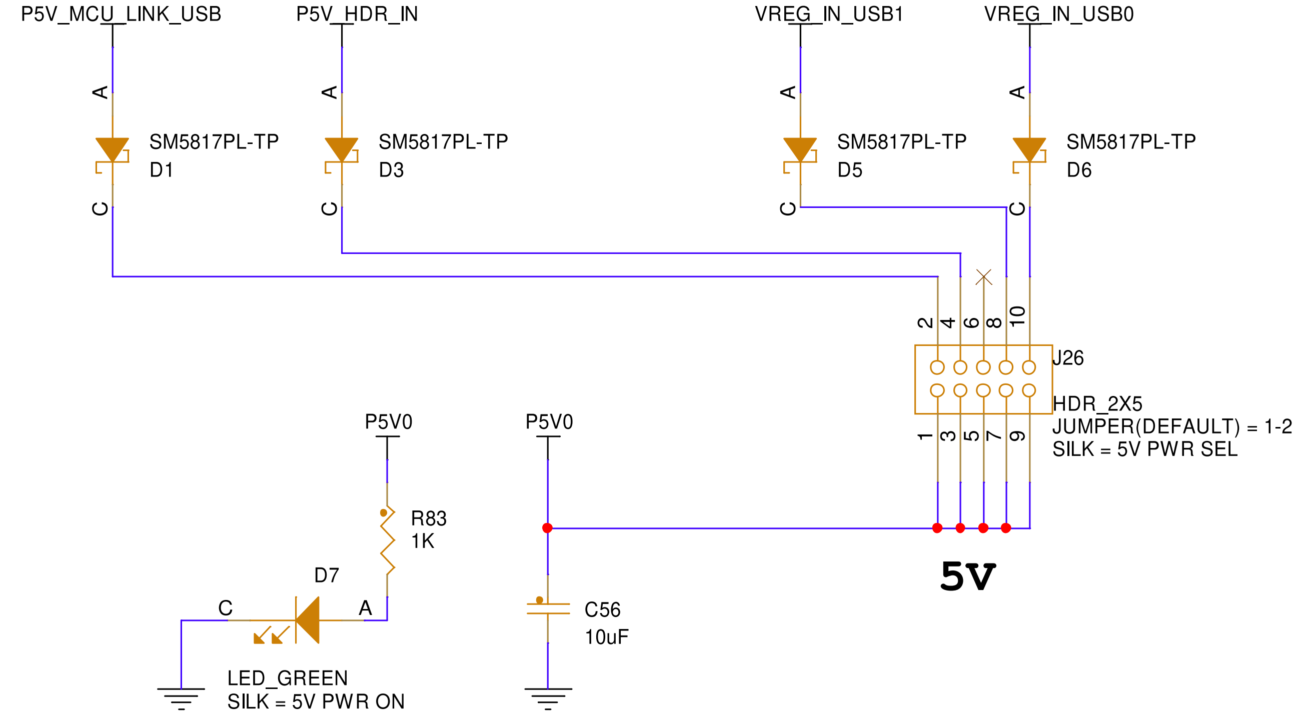


Figure 7. MCX-N9XX-EVK 5 V power source selection

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	PCA9420BSAZ (NXP Semiconductors)	LDO2_OUT: 3.3 V	One of the sources for P3V3 supply
U54	ADM7172 regulator (Analog Devices)	ADM7172_VOUT: 3.3 V	One of the sources for P3V3 supply **(default option)**
-	LDO2_OUT through JP44	P3V3	• Power supply for: – EMV SIM load switch (U48) – SD card load switch (U56) – Ethernet transceiver LAN8741 (U37) through JP4 jumper – FlexCAN transceiver TJA1057 (U39) – Dual supply translating transceivers NTS0302 (U50), NTB0102 (U52), NTB0101 (U51) – mikroBUS connector (J19) – Flex IO header (J20) – M.2 Key E card connector (J12) – Crystal oscillator (Y5) – RGB LED • One of the sources for VDD_P2 supply through the JP47 jumper. For details, see Section 1.6 • One of the sources for VDD_P3 supply through the JP15 jumper. For details, see Section 1.6 • One of the sources for VDD_P4 supply through the JP45 jumper. For details, see Section 1.6 • One of the sources for VDD_USB supply through the JP40 jumper. For details, see Section 1.6 • One of the power sources for VDD_AUDIO • Power source for the MCU_PWR_SRC supply through the JP38 jumper. For details, see Section 1.6
-	ADM7172_VOUT through JP44	P3V3	
J3 (pin 8)	Arduino shield compatible header	P3V3	

The 3.3 V power source for the board is selected by jumper JP44. By default, the ADM172 regulator (U54) output is selected. For the JP44 jumper details, see Section 1.6. Figure 8 shows the P3V3 power selection on the MCX-N9XX-EVK board.

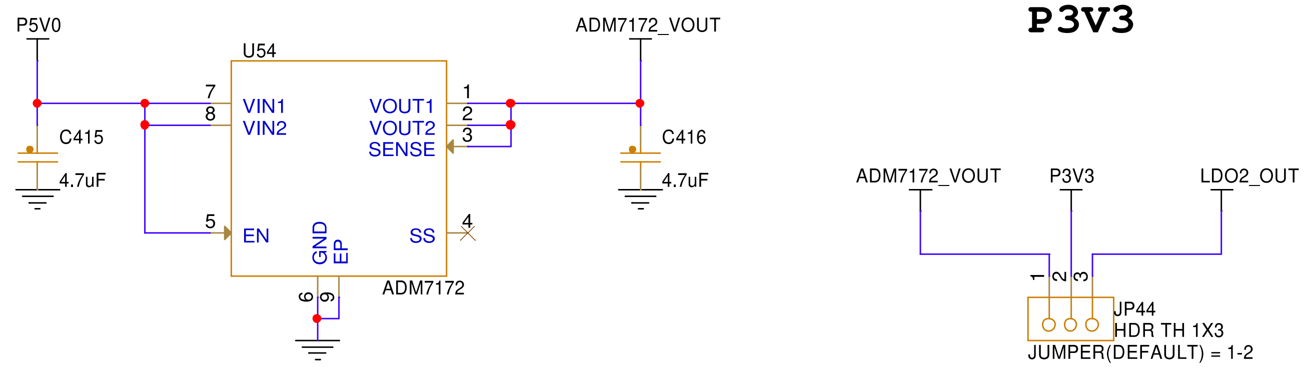


Figure 8. P3V3 source selection

1.8 V power sources and selection

Table 8 describes the 1.8 V input power sources and their output power supplies.

Table 8. P1V8 power sources

Part identifier	Device / power source	Output power supply	Description
U2	PCA9420BSAZ (NXP Semiconductors)	LDO1_OUT: 1.8 V	One of the sources for P1V8 supply
		SW2_OUT: 1.8 V	One of the sources for P1V8 supply **(default option)**
-	LDO1_OUT through JP46 and JP41	P1V8	• Power supply for: – Audio codec DA7212 (Not connected by default) – EMV SIM load switch (U48) (Not a default option) • One of the sources for the VDD_LDO_SYS_IN supply through the JP36 jumper. For details, see Section 1.6 • One of the sources for the VDD_P2 supply through the JP47 jumper. For details, see Section 1.6 • One of the sources for th eVDD_P3 supply through the JP15 jumper. For details, see Section 1.6 • One of the sources for the VDD_P4 supply through the JP45 jumper. For details, see Section 1.6 • One of the power sources for VDD_AUDIO • Power source for the MCU_PWR_SRC supply through the JP38 jumper. For details, see Section 1.6
-	SW2_OUT through JP46 and JP41		

The 1.8 V power source for the board is selected by jumpers JP46 and JP41. By default, SW2_OUT is selected. For the JP46 and JP41 jumpers details, see Section 1.6. Figure 9 shows the P1V8 power selection on the MCX-N9XX-EVK board.

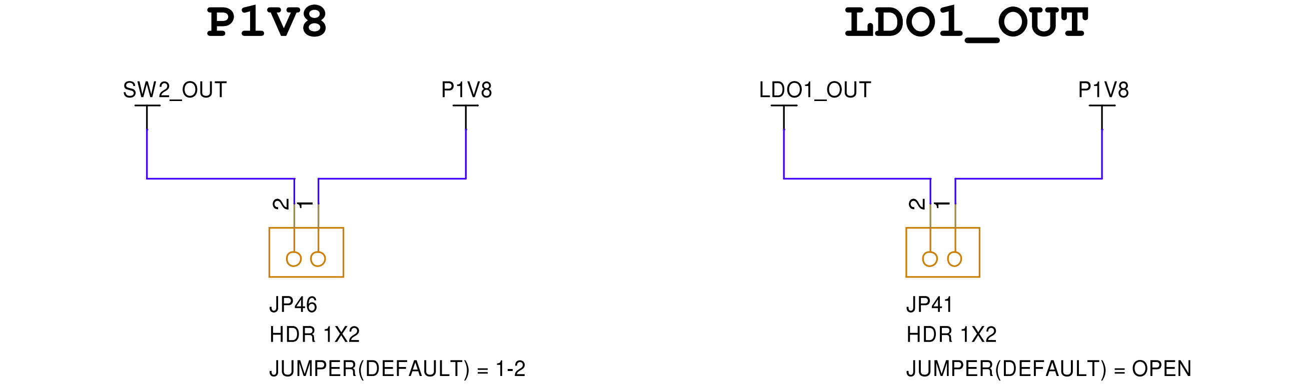


Figure 9. P1V8 source selection

1.1 V power sources and selection

Figure 10 describes the 1.1 V input power sources and their output power supplies.

Table 9. P1V1 power sources

Part identifier	Device / power source	Output power supply	Description
U2	PCA9420BSAZ (NXP Semiconductors)	SW1_OUT	Source for P1V1 supply **(default option)**
-	SW1_OUT through JP48 jumper	P1V1	Used to demonstrate the 1.2 V GPIO capability of P2 / P3 port

The P1V1 power source for the board is selected by jumper JP48. By default, SW1_OUT is selected. For the JP48 jumper details, see Section 1.6. Figure 10 shows the P1V1 power selection on the MCX-N9XX-EVK board.

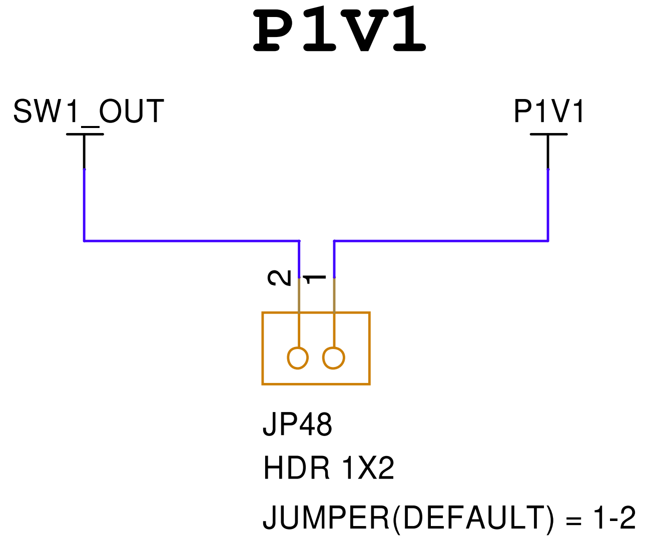


Figure 10. P1V1 source selection

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 as shown in Figure 11.

These jumpers provide access to insert ammeters in all the supplies connecting to the MCX N94X device. They also provide a means of connecting external supplies to any of the MCX power pins.

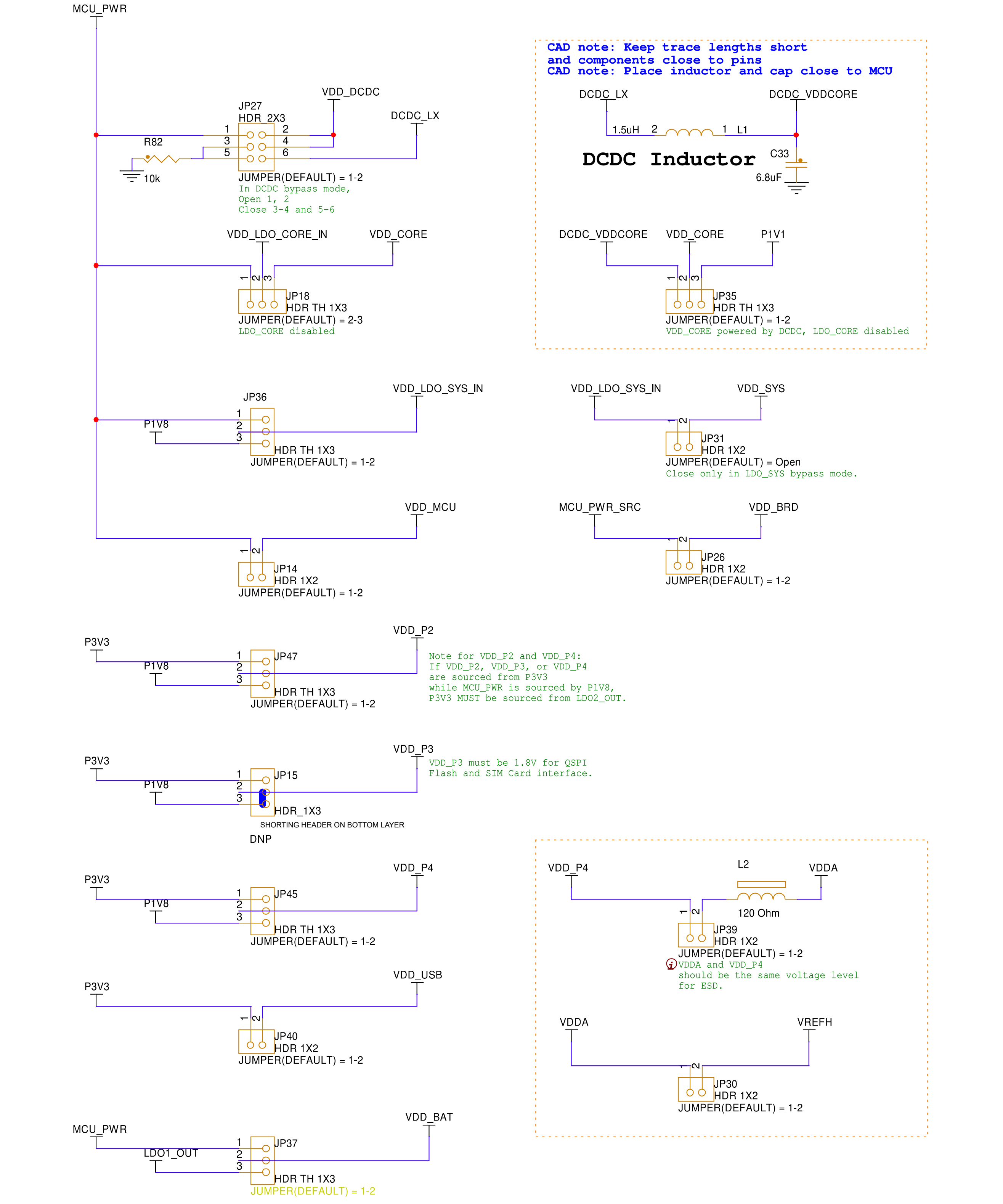


Figure 11. Power configuration jumpers

.

The MCX-N9XX-EVK board supports current measurement for board power supplies through the onboard MCU-Link debug probe. For more details, see Section 3.10.

2.1.1 Power supply configuration

- Table 10 describes the default power jumper configuration for the MCX-N9XX-EVK board.

Table 10. MCX-N9XX-EVK default power supply configuration

Description	JP38	JP27	JP35	JP18	JP36	JP31	JP37	JP47	JP15[1]	JP45
Dual IO, 3.3 V and 1.8 V	1-2	1-2	1-2	2-3	1-2	1-2	1-2	1-2	2-3	1-2

[1] Changing JP15 to 3.3 V requires cutting a cut trace at the bottom of the board and installing a header. If this setting is done, care must be taken to disconnect or remove the onboard flash device, as this device is a 1.8 V only device.

This configuration is selected to demonstrate the 3.3 V capability and to allow for demonstration of a high-speed low-power (1.8 V) QSPI flash device.

Table 11 describes the typical power supply configurations for DCDC operation, while Table 12 describes the typical power supply configurations for LDO operation.

- MCX-N9XX-EVK power supply configurations for DCDC operation

Table 11. Power supply configurations for DCDC operation

Description	JP38	JP27	JP35	JP18	JP36	JP31	JP37	JP47	JP15	JP45
Single supply operation, IO @ 3.3 V	1-2	1-2	1-2	2-3	1-2	1-2	1-2	1-2	1-2	1-2
Single supply operation, IO @ 1.8 V	2-3	1-2	1-2	2-3	2-3	Open	2-3	2-3	2-3	2-3
Dual IO, 3.3 V and 1.8 V	1-2	1-2	1-2	2-3	2-3	2-3	1-2 or 2-3	1-2 or 2-3	1-2 or 2-3	1-2 or 2-3

- MCX-N9XX-EVK power supply configurations for LDO operation

Table 12. Power supply configurations for LDO operation

Description	JP38	JP27	JP35	JP18	JP36	JP31	JP37	JP47	JP15	JP45
Single supply operation, IO @ 3.3 V	1-2	3-4 5-6	Open	1-2	1-2	1-2	1-2	1-2	1-2	1-2
Single supply operation, IO @ 1.8 V	2-3	3-4 5-6	Open	1-2	2-3	Open	2-3	2-3	2-3	2-3
Dual IO, 3.3 V and 1.8 V	1-2	3-4 5-6	Open	1-2	1-2	2-3	1-2 or 2-3	1-2 or 2-3	1-2 or 2-3	1-2 or 2-3

2.1.2 DC-DC inductor

The MCX-N9XX-EVK board uses a 1.5 µH DC-DC inductor L1 (Taiyo Yuden LSBHB2520MKT1R5M). The inductor is enabled when the board is configured in DC-DC Buck mode. Figure 12 shows the DC-DC inductor circuit diagram of the MCX-N9XX-EVK board.

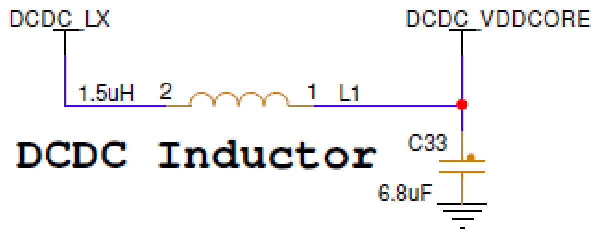


Figure 12. DC-DC inductor circuit diagram

Choosing the right DC-DC inductor for your target board is important. While selecting a DC-DC inductor, look for the following specifications:

- Inductor value: 1.5 µH
- ESR: < 0.3 Ω
- Saturation current (Isat): > 300 mA
- Self-resonant frequency: > 50 MHz

2.2 Clocks

The MCX-N9XX-EVK board provides crystal oscillators to provide accurate time bases for the device and different components on the board.

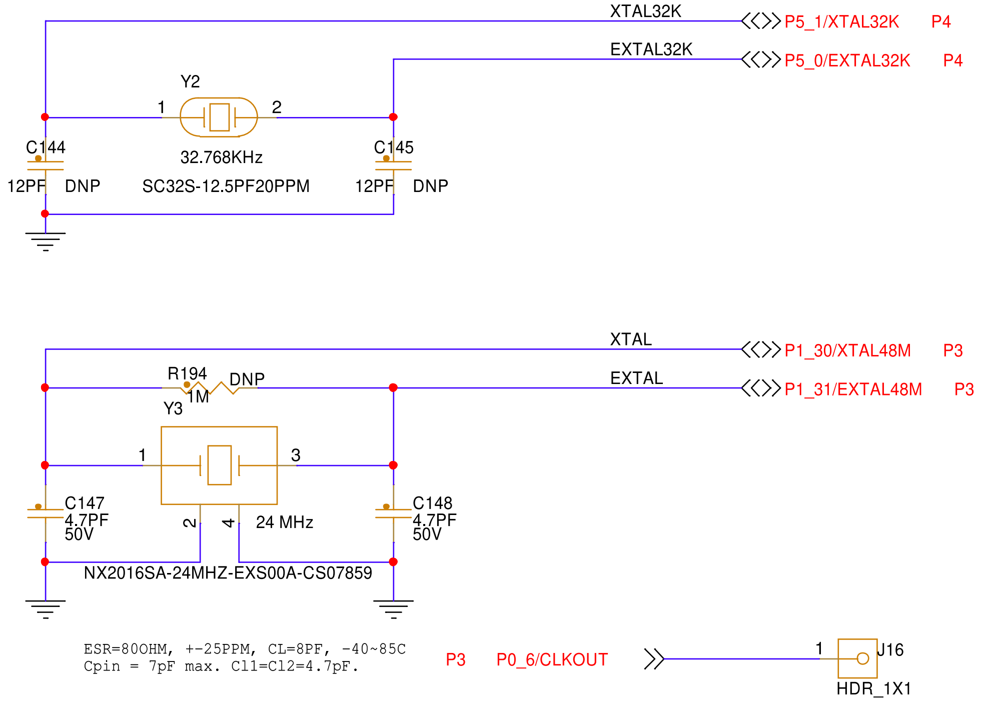


Figure 13. MCX-N9XX-EVK onboard system clocks

Table 13 describes the clock sources available on the MCX-N9XX-EVK board.

Table 13. MCX-N9XX-EVK clocks

Clock generator	Clock frequency	Destination	Description
Crystal oscillator, Y2	32.768 kHz	• Port 5 pin 1 (XTAL32K) of target MCU MCX N94X • Port 5 pin 0 (EXTAL32K) of target MCU MCX N94X	For an 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 TAMPER1 pin. It can be observed at J2_5.
Crystal oscillator, Y3	24 MHz	• Port 1 pin 30 (XTAL48M) of target MCU MCX N94X • Port 1 pin 31 (EXTAL48M) of target MCU MCX N94X	For high-frequency accurate timebase: • Required external load capacitors are provided. • Small package size (2.0 mm x 1.6 mm) • Low-ESR (80 Ω) crystal.
Crystal oscillator, Y5	32.7680 kHz	Pin 50 (SUSCLK) of the M.2 Key E card connector (J12)	Crystal oscillator VDD is supplied by the P3V3 power rail.
Crystal oscillator, Y7	50 MHz	RMII 10/100 Mbit/s Ethernet transceiver (U37)	Provides ETHPHY_CLK to LAN8741 Ethernet PHY depending upon the JP50 jumper setting. It also provides a provision to feed the clock back into the target MCU (MCX N94X). For JP50 details, see Section 1.6.

2.3 USB interface

The target MCU (MCX N94X) features two USB modules (FS USB and HS USB), each with device and host capabilities and a built-in transceiver.

- FS USB is a dual-role USB controller that supports full-speed (FS) device or FS / low-speed (LS) embedded host operation
- HS USB can operate in either device or host mode. In device mode, it can connect to the USB host systems at either HS USB rate of 480 Mbit/s or FS rate of 12 Mbit/s. In host mode, it can connect to peripheral devices operating at HS, FS, or the USB 2.0 LS rate of 1.5 Mbit/s.

Table 14 describes the USB ports available on the MCX-N9XX-EVK board.

Table 14. USB Ports

Part identifier	Connector type	Description
J28	USB2.0 Micro-AB connector	Connects to FS USB controller of the target MCU. The J28 USB port can connect in both Host and Device mode. In Device mode, this port can provide the 5 V power supply source to the board depending upon the J26 jumper setting (see Section 1.6).
J27	USB2.0 Micro-AB connector	Connects to HS USB controller of the target MCU. The J27 USB port can connect in both Host and Device mode. In Device mode, this port can provide the 5 V power supply source to the board depending upon the J26 jumper setting (see Section 1.6).

FS USB interface

On the MCX-N9XX-EVK board, the USB0_DM and USB0_DP signals from the target MCU (MCX N94X) connect to the onboard USB connector (J28) directly through the required 33 ohm resistors and a common mode choke. The common mode choke is included for noise suppression on the DM / DP signals, however, pads are included to install bypass resistors for testing choke-less configurations.

Figure 14 shows the FS USB circuit diagram.

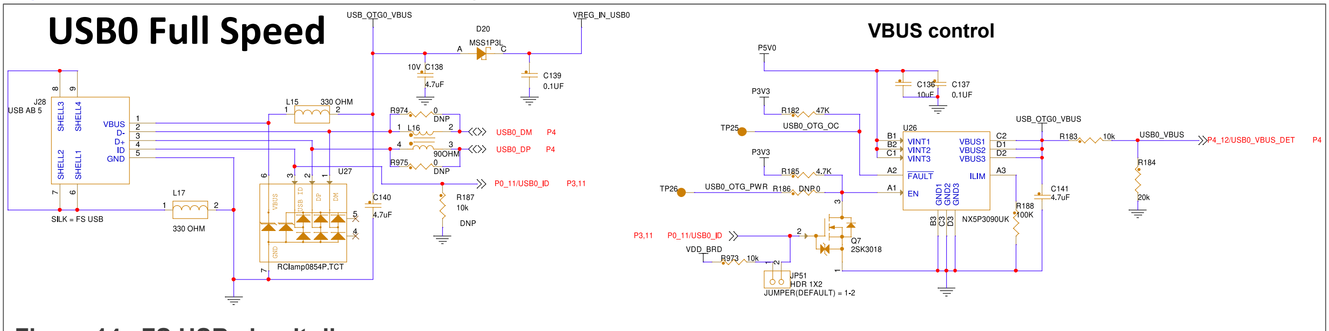


Figure 14. FS USB circuit diagram

HS USB interface

On the MCX-N9XX-EVK board, the USB1_DM and USB1_DP signals from the MCX N94X MCU connect to the onboard USB connector (J27) directly through a common mode choke. The common mode choke is included for noise suppression on the DM / DP signals, however, pads are included to install bypass resistors for testing choke-less configurations.

Figure 15 shows the HS USB circuit diagram.

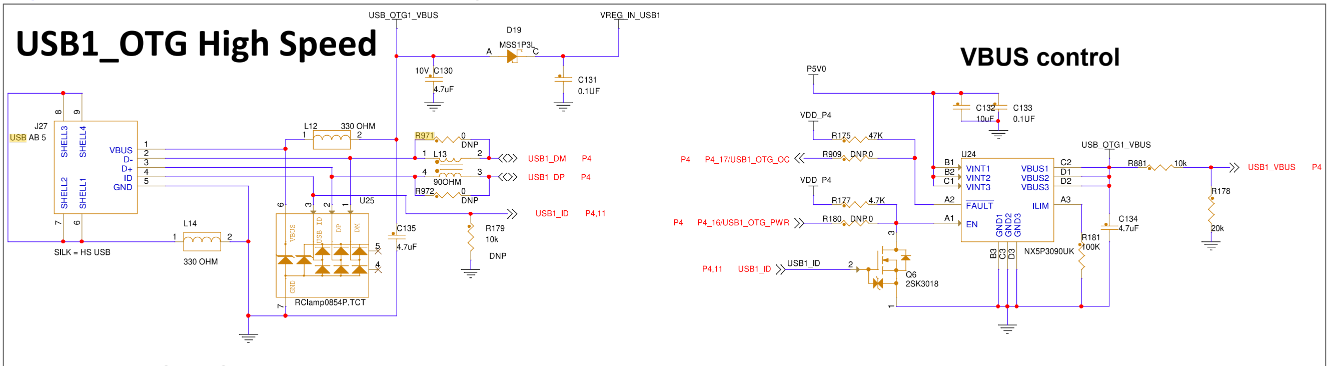


Figure 15. USB HS circuit diagram

2.4 SD card interface

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

On the MCX-N9XX-EVK board, the uSDHC controller connects to the SD card connector (J25).

- The SD card detect pin is an open switch that shorts with GND when the card is inserted.
- The SD card VDD (VDD_SD) is supplied by the P3V3 rail and is controlled by the protected power switch (U56). The Port P0 pin 11 is used to control the ON/OFF of the U56 power switch depending upon the 3-pin jumper (JP17) configuration. For the JP17 Jumper details, see Section 1.6.
- The SD card bus pins are Port P2_[7:1] pins. Therefore, the Port P2 voltage must be set according to the type of card being used.

Figure 16 shows the SD card connections.

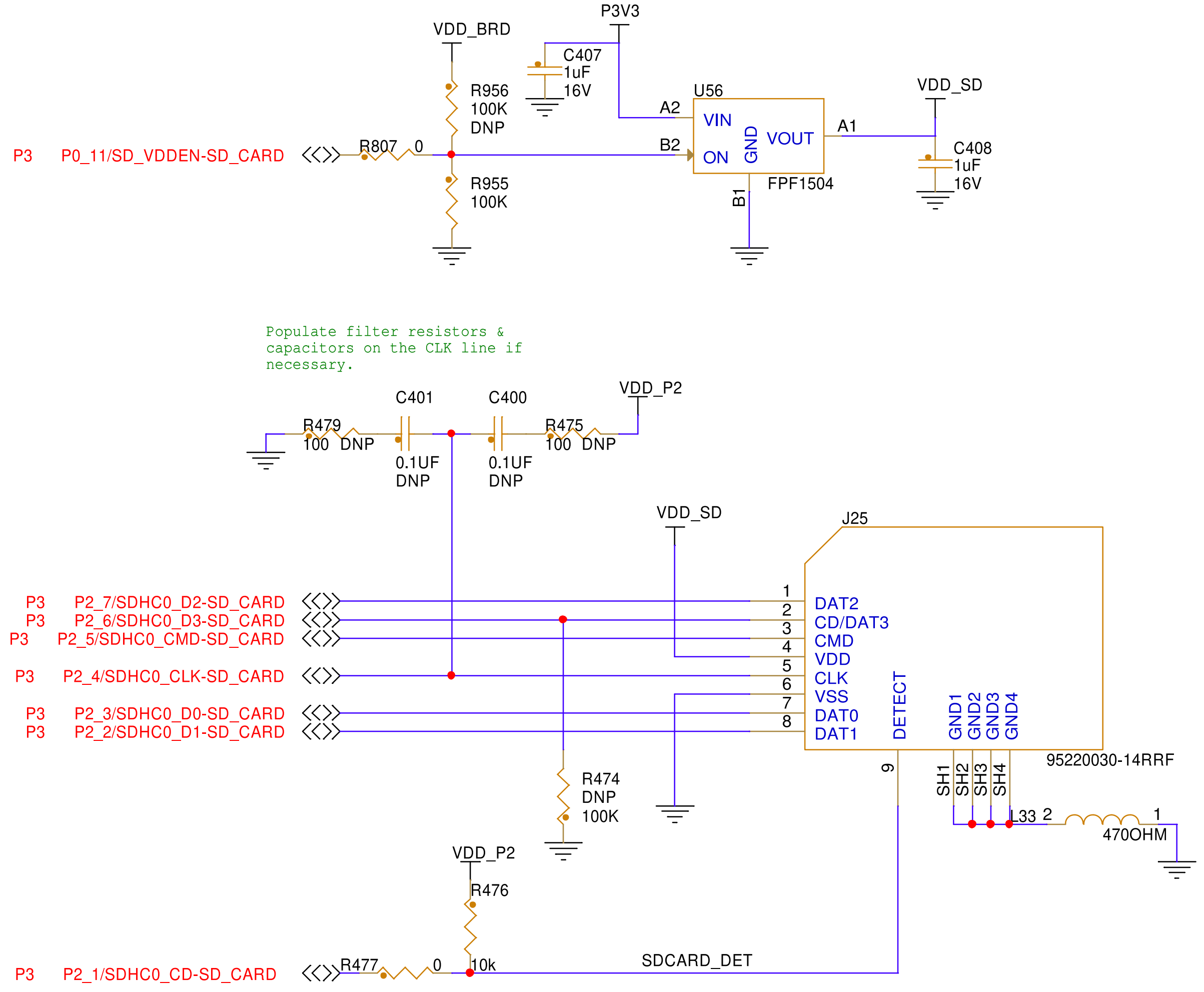


Figure 16. Micro SD card connector circuit

The Port P2_[7:1] lines that are used for SDHC interface signals are also shared with the M.2 connector (J12) signals and the PWM signals on the Arduino compatible header (J3). Zero-ohm onboard resistors are used to allow the selection of signals between the SDHC card slot or the M.2 connector. The Arduino-compatible header sockets are connected by default, however, can also be disconnected easily by removing the appropriate zero-ohm resistors.

Figure 17 shows the zero-ohm resistor connections for uSDHC signals selection.

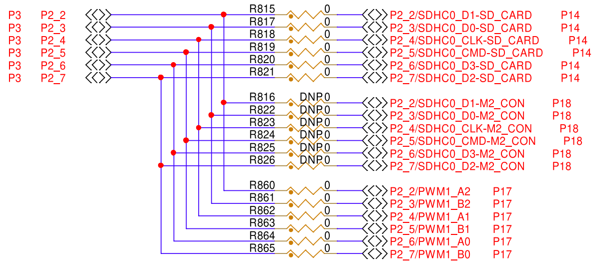

P3 P2_2
P3 P2_3
P3 P2_4
P3 P2_5
P3 P2_6
P3 P2_7
R815 0 P2_2/SDHC0_D1-SD_CARD P14
R817 0 P2_3/SDHC0_D0-SD_CARD P14
R818 0 P2_4/SDHC0_CLK-SD_CARD P14
R819 0 P2_5/SDHC0_CMD-SD_CARD P14
R820 0 P2_6/SDHC0_D3-SD_CARD P14
R821 0 P2_7/SDHC0_D2-SD_CARD P14
R816 DNP 0 P2_2/SDHC0_D1-M2_CON P18
R822 DNP 0 P2_3/SDHC0_D0-M2_CON P18
R823 DNP 0 P2_4/SDHC0_CLK-M2_CON P18
R824 DNP 0 P2_5/SDHC0_CMD-M2_CON P18
R825 DNP 0 P2_6/SDHC0_D3-M2_CON P18
R826 DNP 0 P2_7/SDHC0_D2-M2_CON P18
R860 0 P2_2/PWM1_A2 P17
R861 0 P2_3/PWM1_B2 P17
R862 0 P2_4/PWM1_A1 P17
R863 0 P2_5/PWM1_B1 P17
R864 0 P2_6/PWM1_A0 P17
R865 0 P2_7/PWM1_B0 P17

Figure 17. uSDHC card connection selection resistors

2.5 Flash memory interface

The target MCU (MCX N94X) features one Flexible Serial Peripheral Interface (FlexSPI) controller, which can support an external memory.

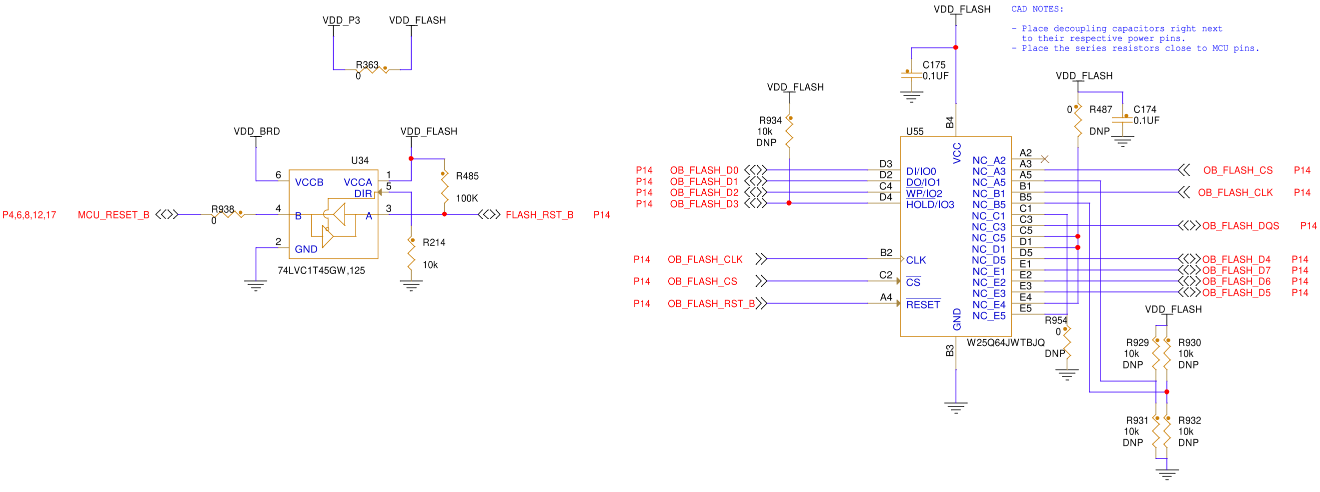
On the MCX-N9XX-EVK board, one Quad SPI memory is provided. The flash memory VCC (VDD_FLASH) is supplied by the P1V8 rail through the JP15 jumper. For JP15 jumper details, see Section 1.6.

Table 15 provides the detail of the Quad SPI flash memory on the board.

Table 15. Quad SPI flash memory

Part identifier	Manufacturer and part name	Description
U55	Winbond W25Q64JWTBJQ	It is a 1.8 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.

Figure 18 shows the flash memory circuit diagram.



CAD NOTES:

- Place decoupling capacitors right next to their respective power pins.
- Place the series resistors close to MCU pins.

Figure 18. FlexSPI 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.

The MCX N94X FlexSPI controller can connect either to an onboard QSPI flash memory (U55) or to an external flash memory connected through the Flash daughter card connector (J9). The recommended connector for the flash daughter card connector is the Samtec QSE-014-01-F-D-DP-A.

Some of the FlexSPI signals are also multiplexed with the SAI signals for the M.2 connector. Zero-ohm onboard resistors are used to allow the selection of signals either for the QSPI flash memory or for the other connectors (flash data card connector or SAI interface of the M.2 connector). By default, signals for the QSPI flash memory are selected.

Onboard zero ohm resistors are used to connect or disconnect the flash interface lines from the onboard QSPI flash memory chip and connect to the flash data card connector or SAI interface of the M.2 connector.

Figure 19 displays the resistors that make these connections.

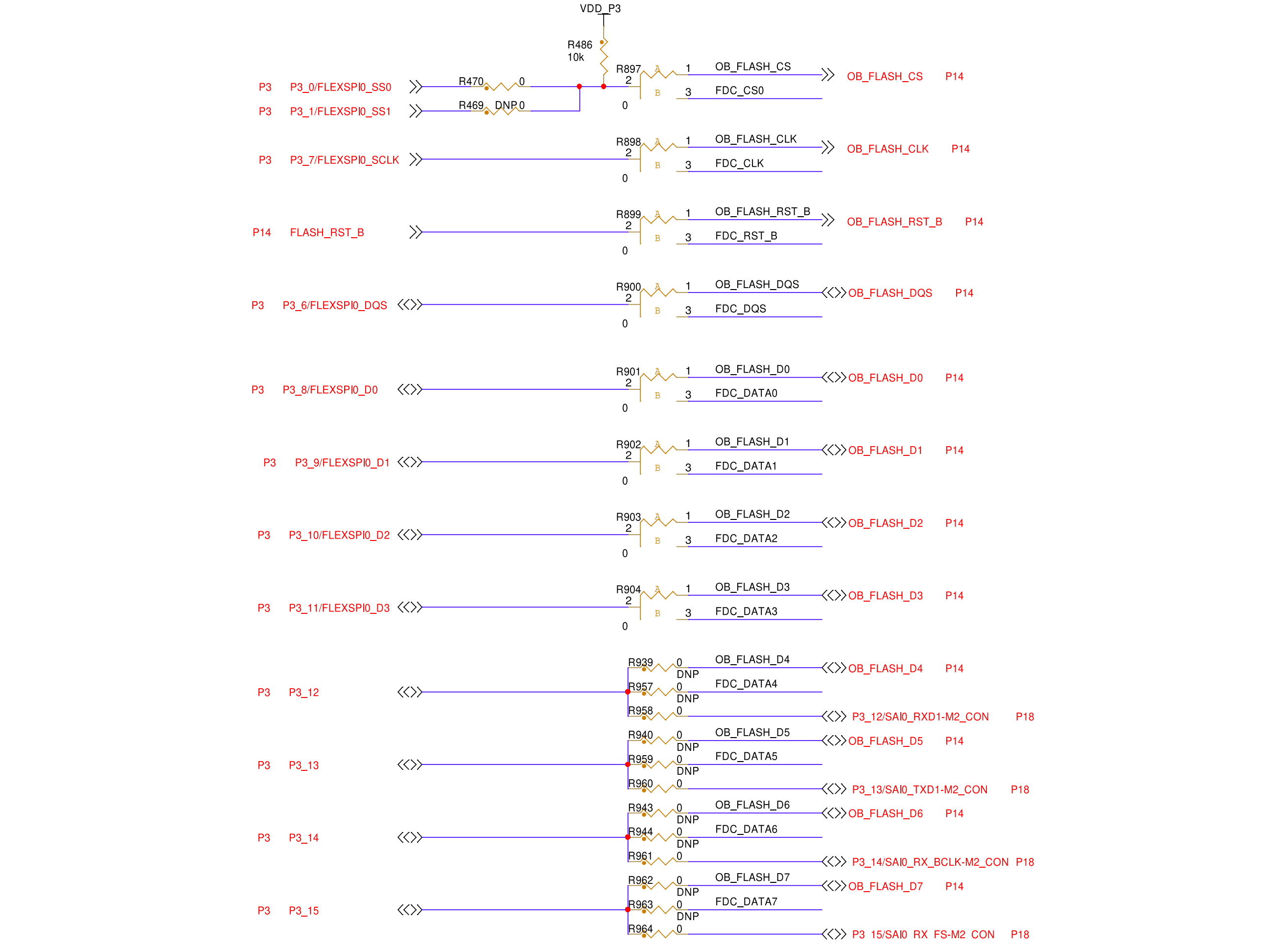


Figure 19. Resistors configuration

Table 16 describes the zero-ohm resistors configuration to select onboard flash memory or other board components on the flash interface lines.

Table 16. Zero-ohm resistors configuration

Port[pin] / Signal	Resistor	Resistor configuration	Description
P3_0	R897	• Selection A (Pin 1-2) **(default setting)**: Flash interface line connects to onboard flash memory • Selection B (Pin 2-3): Flash interface line connects to flash daughter card connector	Connects FLEXSPI0_SS0 to the flash chip select signal (through resistor R470)
P3_1	R897		Connects FLEXSPI0_SS1 to the flash chip select signal (through resistor R469, which is DNP by default) ***Note:*** *Both resistors, R470 and R469 should not be installed simultaneously.*
P3_7	R898		For flash clock signal
FLASH_RST_B[1]	R899		For flash reset
P3_6	R900		For flash data strobe signal

Table 16. Zero-ohm resistors configuration*...continued*

Port[pin] / Signal	Resistor	Resistor configuration	Description
P3_8	R901		For flash data signals
P3_9	R902		
P3_10	R903		
P3_11	R904		
P3_12	R939, R957, R958	• R939 Populated (default setting): Connects the signals from the P3_12 pin with the onboard flash memory • R957 Populated: Connects the signals from the P3_12 pin with the flash daughter card connector. DNP by default. • R958 Populated: Connects the signals from the P3_12 pin with the SAI0 interface of M.2 connector. DNP by default.	For flash data signal / SAI0 RXD
P3_13	R940, R959, R960	• R940 Populated (default setting): Connects the signals from the P3_13 pin with the onboard flash memory • R959 Populated: Connects the signals from the P3_13 pin with the flash daughter card connector. DNP by default. • R960 Populated: Connects the signals from the P3_13 pin with the SAI0 interface of the M.2 connector. DNP by default.	For flash data signal / SAI0 TXD
P3_14	R943, R944, R961	• R943 Populated (default setting): Connects the signals from the P3_14 pin with the onboard flash memory • R944 Populated: Connects the signals from the P3_14 pin with the flash daughter card connector. DNP by default. • R961 Populated: Connects the signals from the P3_14 pin with the SAI0 interface of the M.2 connector. DNP by default.	For flash data signal / SAI RX BCLK
P3_15	R962, R963, R964	• R962 Populated (default setting): Connects the signals from the P3_15 pin with the onboard flash memory • R963 Populated: Connects the signals from the P3_15 pin with the flash daughter card connector. DNP by default. • R964 Populated: Connects the signals from the P3_15 pin with the SAI0 interface of the M.2 connector. DNP by default.	For flash data signal / SAI RX FS

[1] The FLASH_RST_B is driven by the MCU_RESET_B signal.

2.6 Ethernet interface

The target MCU (MCX N94X) features one Ethernet controller (ENET0) module.

On the MCX-N9XX-EVK 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 MCX-N9XX-EVK only supports RMII configuration. For this reason, the TXD3 and TXD2 pins have been grounded through resistors R260 and R261.

Table 17 describes the onboard devices supporting the Ethernet interface.

Table 17. Ethernet interface devices

Part identifier	Part name and Manufacturer	Description
J8	Würth Elektronik 7499211121A	Shielded RJ45 connector jack with magnetic built-in to connect to an Ethernet cable
U37	Microchip Technology LAN8741	Single-chip 10 /100 Mbit/s RMII Ethernet PHY is compliant with IEEE802.3/802.3u (Fast Ethernet), ISO 802-3/IEEE 802.3 (10 BASE-T), and Energy-Efficient Ethernet IEEE 802.3az

Input to the XTAL1/CLKIN pin of the Ethernet PHY can be either of the following:

- ENET0_TXCLK clock from the target MCU (MCX N94X) received through the P1_4 port. This is the default configuration.
- OSC_50M clock from an external 50 MHz crystal oscillator (Y7). This oscillator can be enabled/disabled through the JP50 jumper. The jumper is disabled by default to allow the target MCU to provide the PHY clock (and should always be disabled when the target MCU has to provide the PHY clock). For details, see Section 1.6.

The Ethernet circuit supports both 3.3 V and 1.8 V I/O operation. However, the jumper JP4 is provided to allow for complete depowering of the Ethernet circuit. This may be necessary when connecting large loads to the P3V3 net. For the JP4 jumper details, see Section 1.6.

2.7 Accelerometer sensor interface

On the MCX-N9XX-EVK board, an accelerometer sensor is used to sense motion, a feature required in the IoT application space.

The main features of the Accelerometer sensor interface are as follows:

- 3-Axis Low-G MEMS accelerometer sensor device FXLS8964AF (U40) is used.
- The sensor device is powered by the VDD_FXL supply, which is tied to the VDD_P4 supply through a zero ohm resistor (R893).
- Discrete pull-up resistors for the I2C bus lines are provided
- The default I2C address for the device is configured as 0x19. Address can be changed by pull-up / pull-down resistors on the SA0 line.
  - With a pull-up resistor (R299), a high signal on SA0, and the 7-bit I2C address is 0x19.
  - With a pull-down resistor (R896), a low signal on SA0, and the 7-bit I2C address is 0x18.
- Two interrupt signals routed to test pads for prototyping, if required.
- The I2C uses shared lines for the I2C interface.
- Series zero ohm resistors (R42 and R43) are provided to isolate the sensor from the MCX N94X device.

Figure 20 shows the FXLS8964AF sensor circuit diagram.

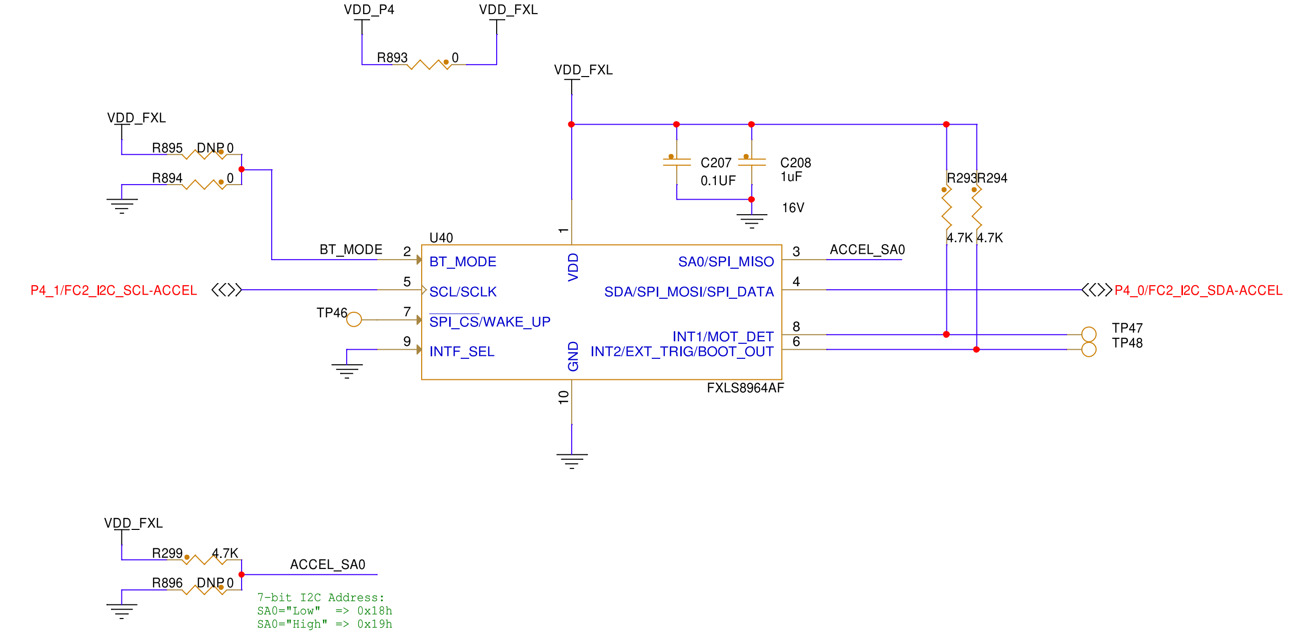


Figure 20. FXLS8964AF sensor circuit

2.8 I3C interface

The MCX-N9XX-EVK includes one P3T1755 digital temperature sensor to demonstrate the I3C capabilities of the target MCU. This device allows for 32 I3C provisional IDs, supports the fully operating voltage of the board (1.71 V - 3.6 V), programmable overtemperature alerts, 12b resolution, and has an accuracy of ± 1 °C.

Jumper JP23 is used for the I3C Pull-Up Resistance (PUR) provision on the I3C1 SDA signal. For JP23 details, see Section 1.6

Figure 21 shows the I3C sensor schematic diagram.

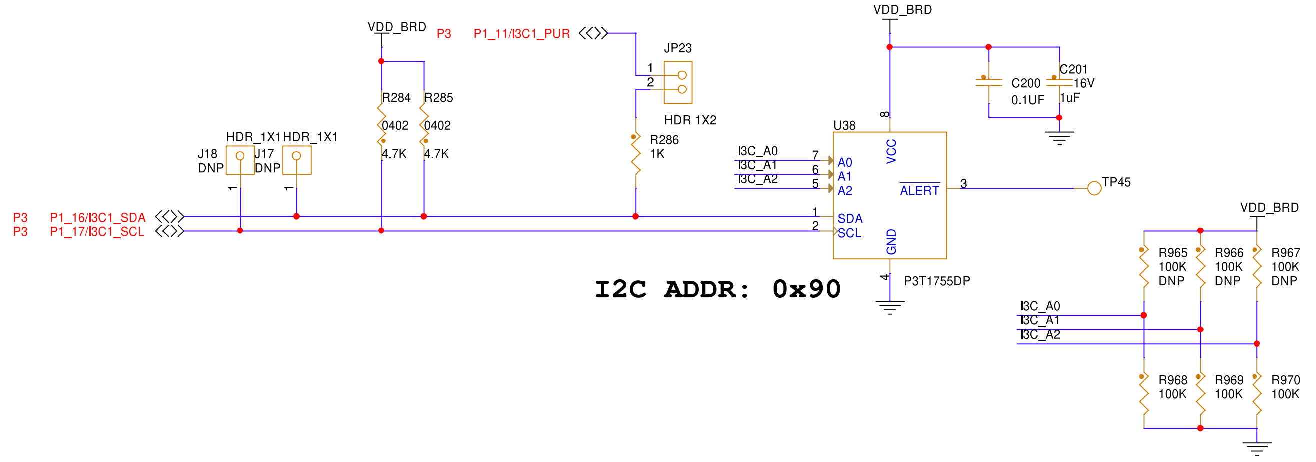


Figure 21. I3C sensor schematic diagram

2.9 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 (MCX N94X) supports two CAN (w/wo FD) controllers (CAN0 to CAN1).

On MCX-N9XX-EVK, only the CAN0 controller is used. The CAN0 controller connects to a 4-pin CAN header through a CAN transceiver. The CAN0_TX and CAN0_RX signals are through ports P1_18 and P1_19, respectively. As the TJA1057 VIO pin allows for direct interfacing with 3.3 V and 5 V (and not 1.8 V), a 2-bit dual-supply translating transceiver (NTS0302) is provided if the target MCU P1 pins are powered with 1.8 V supply. The NTS0302 (U50) transceiver provides bidirectional voltage translations from VDD_BRD to P3V3 and vice versa.

Table 18 describes the HS CAN transceiver and 4-pin CAN header used on the board.

Table 18. High-speed CAN transceiver and header

Part identifier	Manufacturing part number	Description
U39	TJA1057GT/3	High-speed CAN transceiver. Provides an interface between the CAN0 controller and the physical two-wire CAN0 bus. s
J29	-	4-pin CAN header. It is connected to the CAN0 bus and allows external connection with the bus.

Figure 22 shows the FlexCAN interface schematic.

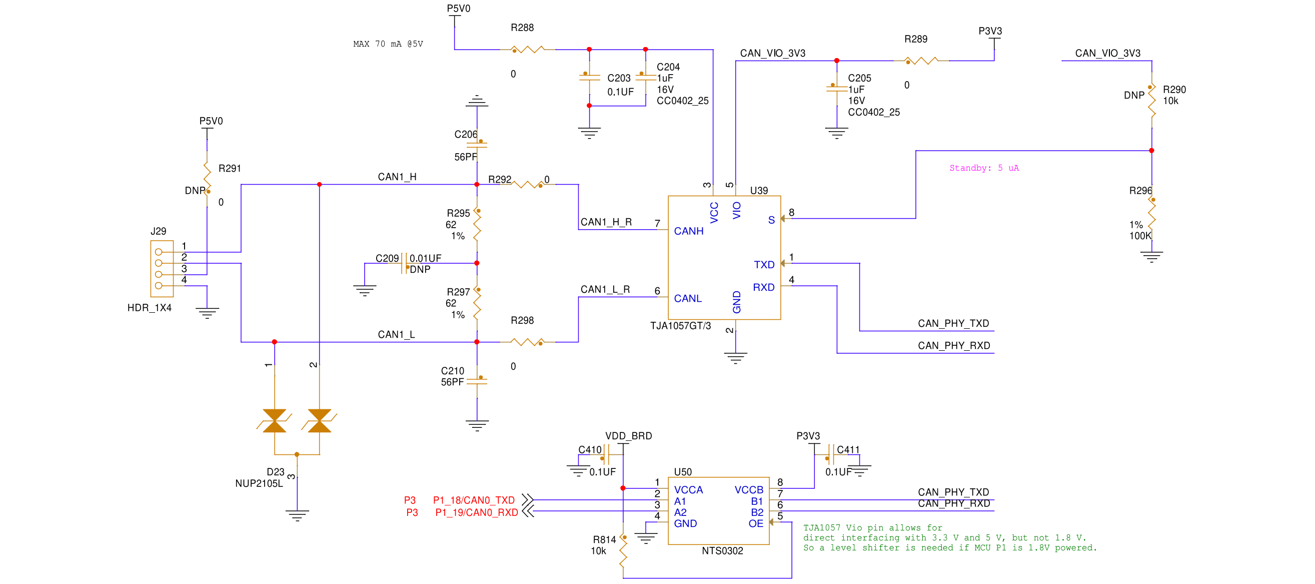


Figure 22. FlexCAN interface schematic

Table 19 describes the 4-pin J19 CAN header pinout.

Table 19. 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

2.10 Visible light sensor interface

On the MCX-N9XX-EVK board, one phototransistor is provided, which connects to the ADC input channel ANA_4 of the target device (MCX N94X) for evaluating the ADC module.

Table 20 provides the detail of the light sensor device on the board.

Table 20. Light sensor device

Part identifier	Manufacturer and part name	Description
Q10	Everlight ALS-PT19-315C/L177/TR8	It is a low-cost ambient light sensor, consisting of phototransistor in miniature SMD.

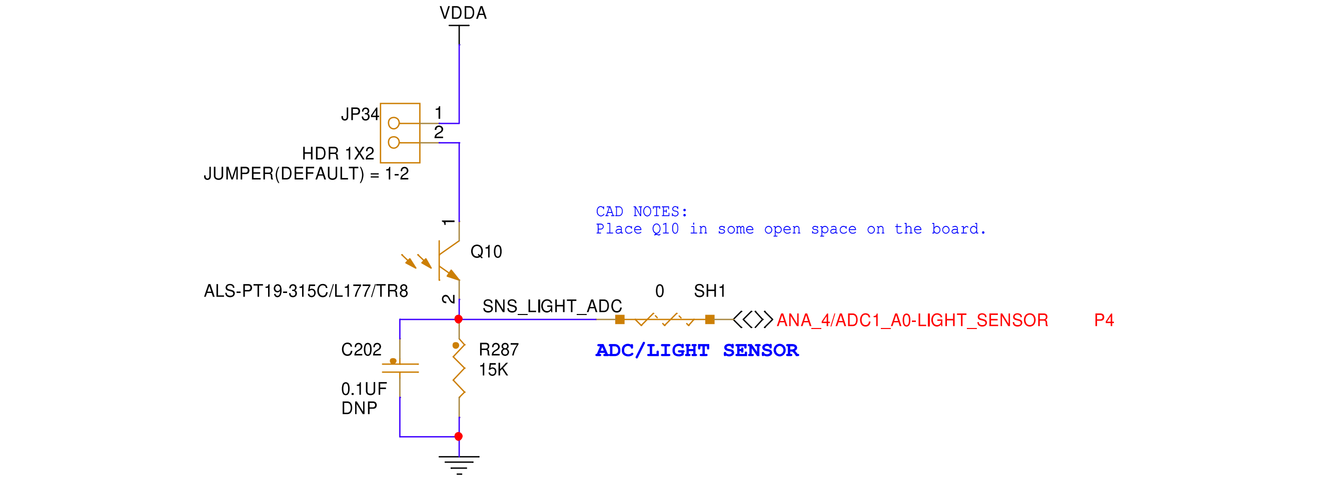


Figure 23. Light sensor circuit diagram

The light sensor output is not shared with other GPIO or devices and is connected via a high-speed analog channel. The light sensor can be isolated from the MCX N94X device by cutting the shorting link SH1.

The input voltage to the light sensor is VDDA through jumper connection JP34 to allow for removing the light sensor from any current measurements being made. For the JP34 jumper details, see Section 1.6. If VDDA is more than the configured VREFH, the maximum voltage the ADC can convert is that of VREFH.

If no light reaches the light sensor, a small current is drawn from VDDA.

2.11 M.2 interface

The MCX-N9XX-EVK board features a M.2 Key E mini card 75-pin connector (J12) for Wi-Fi support. The M.2 mini card connector supports USB, SDHC, I2S, UART, I2C, and GPIO connections. The connector can be used for Wi-Fi/Bluetooth card, 802.15.4 Radio, or 3G/4G card.

This card connector supports the Murata Wi-Fi card module, which is based on an NXP IW416 chip supporting dual-band 1x1 Wi-Fi 4 and Bluetooth 5.2.

Figure 24 shows the M.2 connector interface circuit diagram.

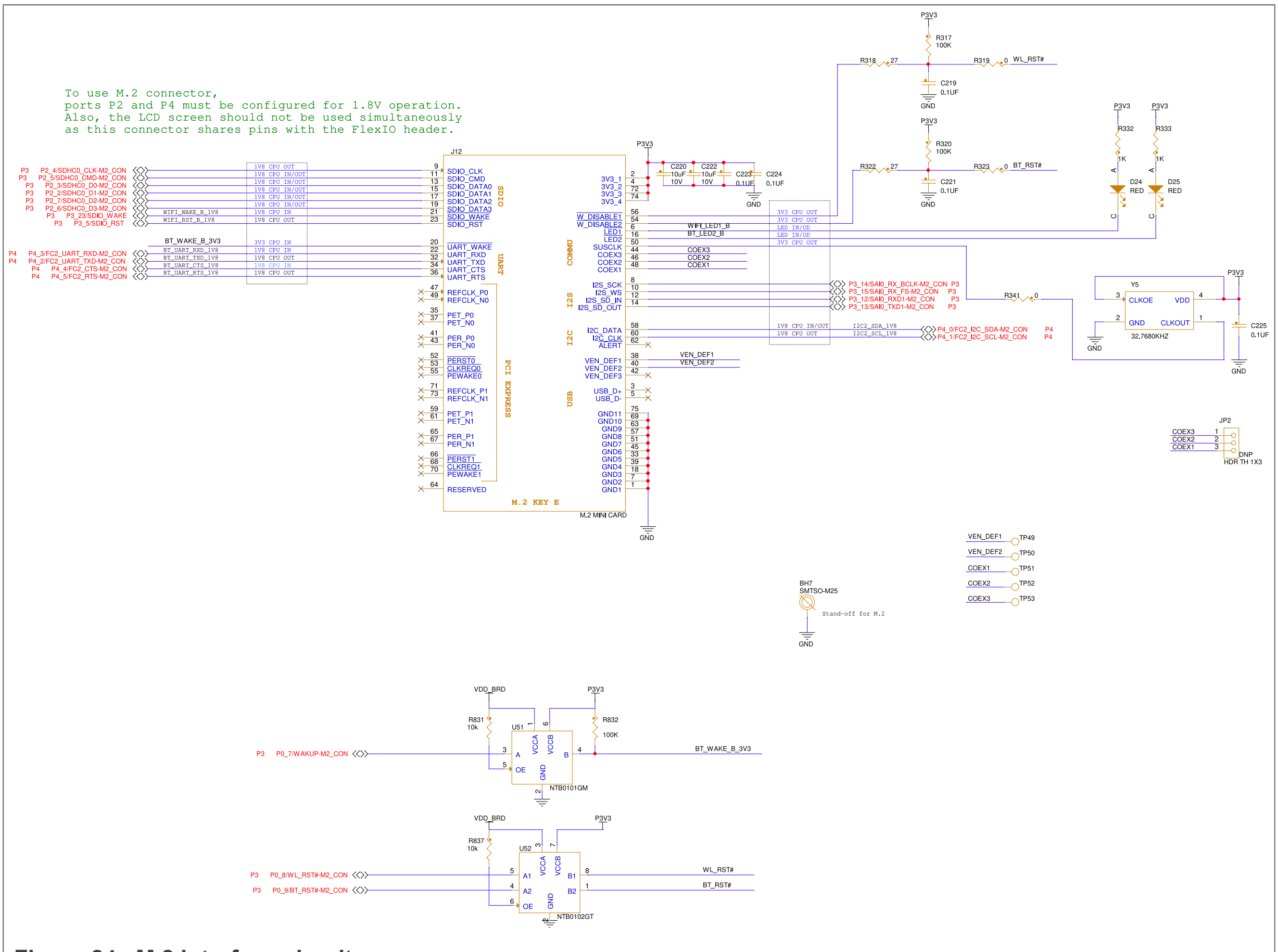


Figure 24. M.2 interface circuit

Note:

- *Many of these connections are not the default connections and are shared with other features of the MCX-N9XX-EVK. Be sure to check that the signals you intend to use are properly connected to the M.2 connector circuit and do not overlap with other desired peripherals.*
- *Ports P2 and P4 must be configured for 1.8 V operation to use this card connector.*

Table 21 describes the pinout of the M.2 connector (J12).

Table 21. M.2 connector (J12) pinout

Pin number	Net name	GPIO	Potential conflict
1	GND	-	-
2	3V3	-	-
3	USB D+	-	-
4	3V3	-	-
5	USB D-	-	-
6	LED1	-	-
7	GND	-	-
8	I2S_SCK	P3_14	• Flash Daughter card connector (J9 pin 25)

Table 21. M.2 connector (J12) pinout*...continued*

Pin number	Net name	GPIO	Potential conflict
			• Onboard flash memory (U55 pin E2)
9	SDIO_CLK	P2_4	• SDHC card connector (J25 pin 5) • Arduino compatible header (J3 pin 11)
10	I2S_WS	P3_15	• Flash Daughter card connector (J9 pin 26) • Onboard flash memory (U55 pin E1)
11	SDIO_CMD	P2_5	• SDHC card connector (J25 pin 3) • Arduino compatible header (J3 pin 9)
12	I2S_SD_IN	P3_12	• Flash Daughter card connector (J9 pin 21) • Onboard flash memory (U55 pin D5)
13	SDIO_DATA0	P2_3	SDHC card connector (J25 pin 7) Arduino compatible header (J3 pin 5)
14	I2S_SD_OUT	P3_13	• Flash Daughter card connector (J9 pin 22) • Onboard flash memory (U55 pin E3)
15	SDIO_DATA1	P2_2	• SDHC card connector (J25 pin 8) • Arduino compatible header (J3 pin 7)
16	LED2	-	-
17	SDIO_DATA2	P2_7	• SDHC card connector (J25 pin 1) • Arduino compatible header (J3 pin 13)
18	GND	-	-
19	SDIO_DATA3	P2_6	• SDHC card connector (J25 pin 2) • Arduino header (J3 pin 15)
20	UART_WAKE	P0_7	FLEXIO header (J20 pin 8)
21	SDIO_WAKE	P3_23	-
22	UART_RXD	P4_3	• Arduino header (J1 pin 2) • mikroBUS connector (J22 pin 3)
23	SDIO_RST	P3_5	-
24	-	-	-
25	-	-	-
26	-	-	-
27	-	-	-
28	-	-	-
29	-	-	-
30	-	-	-
31	-	-	-
32	UART_TXD	P4_2	• Arduino header (J1 pin 4) • mikroBUS connector (J22 pin 4)
33	GND	-	-
34	UART_CTS	P4_4	-
35	PET_P0	-	-

Table 21. M.2 connector (J12) pinout*...continued*

Pin number	Net name	GPIO	Potential conflict
36	UART_RTS	P4_5	Arduino header (J1 pin 1)
37	PET_N0	-	-
38	VEN_DEF1	-	-
39	GND	-	-
40	VEN_DEF2	-	-
41	PER_P0	-	-
42	VEN_DEF3	-	-
43	PER_N0	-	-
44	COEX3	-	-
45	GND	-	-
46	COEX2	-	-
47	REFCLK_P0	-	-
48	COEX1	-	-
49	REFCLK_N0	-	-
50	SUSCLK	-	-
51	GND	-	-
52	PERST0	-	-
53	CLKREQ0	-	-
54	W_DISABLE2	P0_9	FLEXIO header (J20 pin 10)
55	PEWAKE0	-	
56	W_DISABLE1	P0_8	FLEXIO header (J20 pin 11)
57	GND	-	-
58	I2C_DATA	P4_0	-
59	PET_P1	-	-
60	I2C_CLK	P4_1	-
61	PET_N1	-	-
62	ALERT		
63	GND	-	-
64	RESERVED	-	-
65	PER_P1	-	-
66	PERST1	-	-
67	PER_N1	-	-
68	CLKREQ1	-	-
69	GND	-	-
70	PEWAKE1	-	-
71	REFCLK_P1	-	-

Table 21. M.2 connector (J12) pinout*...continued*

Pin number	Net name	GPIO	Potential conflict
72	3V3	-	-
73	REFCLK_N1	-	-
74	3V3	-	-
75	GND	-	-

2.12 Arduino compatible I/O headers

The MCX-N9XX-EVK 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 22 describes the connectors of the Arduino socket.

Table 22. Arduino socket connectors

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

Figure 25 shows the pinout of the Arduino socket connectors.

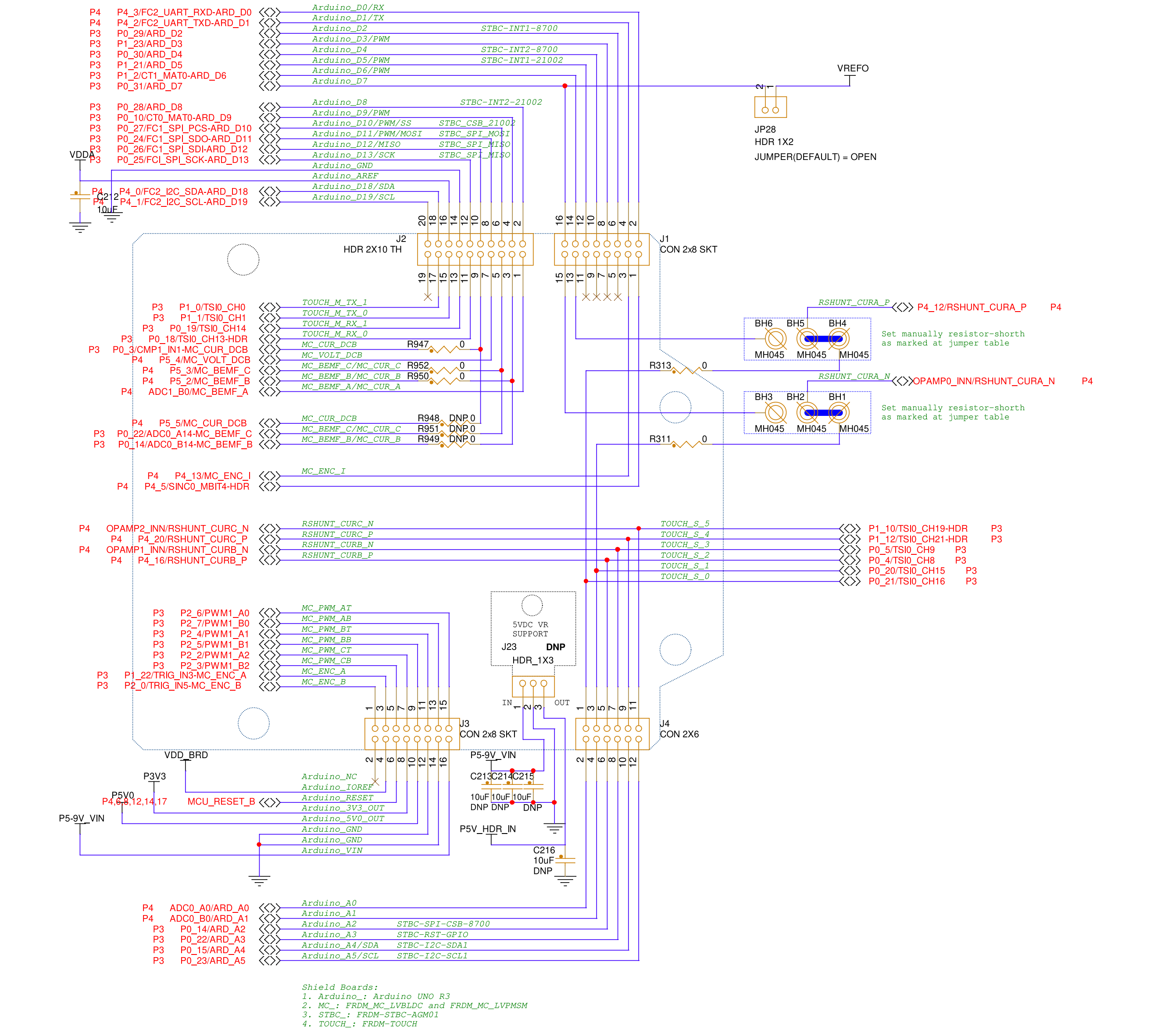

P4 P4_3/FC2_UART_RXD-ARD_D0 Arduino_D0/RX
P4 P4_2/FC2_UART_TXD-ARD_D1 Arduino_D1/TX
P3 P0_29/ARD_D2 Arduino_D2 STBC-INT1-8700
P3 P1_23/ARD_D3 Arduino_D3/PWM
P3 P0_30/ARD_D4 Arduino_D4 STBC-INT2-8700
P3 P1_21/ARD_D5 Arduino_D5/PWM STBC-INT1-21002
P3 P1_2/CT1_MAT0-ARD_D6 Arduino_D6/PWM
P3 P0_31/ARD_D7 Arduino_D7
P3 P0_28/ARD_D8 Arduino_D8 STBC-INT2-21002
P3 P0_10/CT0_MAT0-ARD_D9 Arduino_D9/PWM
P3 P0_27/FC1_SPI_PCS-ARD_D10 Arduino_D10/PWM/SS STBC_CSB_21002
P3 P0_24/FC1_SPI_SDO-ARD_D11 Arduino_D11/PWM/MOSI STBC_SPI_MOSI
P3 P0_26/FC1_SPI_SDI-ARD_D12 Arduino_D12/MISO STBC_SPI_MISO
P3 P0_25/FC1_SPI_SCK-ARD_D13 Arduino_D13/SCK STBC_SPI_MISO
Arduino_GND
Arduino_AREF
VDDA
P4 P4_0/FC2_I2C_SDA-ARD_D18 Arduino_D18/SDA
P4 P4_1/FC2_I2C_SCL-ARD_D19 Arduino_D19/SCL
C212 10uF
VREFO
JP28
HDR 1X2
JUMPER(DEFAULT) = OPEN
J2
HDR 2X10 TH
J1
CON 2x8 SKT
P3 P1_0/TSI0_CH0 TOUCH_M_TX_1
P3 P1_1/TSI0_CH1 TOUCH_M_TX_0
P3 P0_19/TSI0_CH14 TOUCH_M_RX_1
P3 P0_18/TSI0_CH13-HDR TOUCH_M_RX_0
P3 P0_3/CMP1_IN1-MC_CUR_DCB MC_CUR_DCB R947 0
P4 P5_4/MC_VOLT_DCB MC_VOLT_DCB
P4 P5_3/MC_BEMF_C MC_BEMF_C/MC_CUR_C R952 0
P4 P5_2/MC_BEMF_B MC_BEMF_B/MC_CUR_B R950 0
P4 ADC1_B0/MC_BEMF_A MC_BEMF_A/MC_CUR_A
P4 P5_5/MC_CUR_DCB MC_CUR_DCB R948 DNP 0
P3 P0_22/ADC0_A14-MC_BEMF_C MC_BEMF_C/MC_CUR_C R951 DNP 0
P3 P0_14/ADC0_B14-MC_BEMF_B MC_BEMF_B/MC_CUR_B R949 DNP 0
P4 P4_13/MC_ENC_I MC_ENC_I
P4 P4_5/SINC0_MBIT4-HDR
RSHUNT_CURA_P P4_12/RSHUNT_CURA_P P4
BH6 BH5 BH4
MH045 MH045 MH045
Set manually resistor-shorth as marked at jumper table
R313 0
RSHUNT_CURA_N OPAMP0_INN/RSHUNT_CURA_N P4
BH3 BH2 BH1
MH045 MH045 MH045
Set manually resistor-shorth as marked at jumper table
R311 0
P4 OPAMP2_INN/RSHUNT_CURC_N RSHUNT_CURC_N
P4 P4_20/RSHUNT_CURC_P RSHUNT_CURC_P
P4 OPAMP1_INN/RSHUNT_CURB_N RSHUNT_CURB_N
P4 P4_16/RSHUNT_CURB_P RSHUNT_CURB_P
TOUCH_S_5 P1_10/TSI0_CH19-HDR P3
TOUCH_S_4 P1_12/TSI0_CH21-HDR P3
TOUCH_S_3 P0_5/TSI0_CH9 P3
TOUCH_S_2 P0_4/TSI0_CH8 P3
TOUCH_S_1 P0_20/TSI0_CH15 P3
TOUCH_S_0 P0_21/TSI0_CH16 P3
P3 P2_6/PWM1_A0 MC_PWM_AT
P3 P2_7/PWM1_B0 MC_PWM_AB
P3 P2_4/PWM1_A1 MC_PWM_BT
P3 P2_5/PWM1_B1 MC_PWM_BB
P3 P2_2/PWM1_A2 MC_PWM_CT
P3 P2_3/PWM1_B2 MC_PWM_CB
P3 P1_22/TRIG_IN3-MC_ENC_A MC_ENC_A
P3 P2_0/TRIG_IN5-MC_ENC_B MC_ENC_B
5VDC VR SUPPORT
J23 DNP
HDR_1X3
IN OUT
J3
CON 2x8 SKT
J4
CON 2X6
VDD_BRD
P3V3
P5V0
P4,6,8,12,14,17
MCU_RESET_B
P5-9V_VIN
Arduino_NC
Arduino_IOREF
Arduino_RESET
Arduino_3V3_OUT
Arduino_5V0_OUT
Arduino_GND
Arduino_GND
Arduino_VIN
P5-9V_VIN
C213 C214 C215
10uF 10uF 10uF
DNP DNP DNP
P5V_HDR_IN
C216
10uF
DNP
P4 ADC0_A0/ARD_A0 Arduino_A0
P4 ADC0_B0/ARD_A1 Arduino_A1
P3 P0_14/ARD_A2 Arduino_A2 STBC-SPI-CSB-8700
P3 P0_22/ARD_A3 Arduino_A3 STBC-RST-GPIO
P3 P0_15/ARD_A4 Arduino_A4/SDA STBC-I2C-SDA1
P3 P0_23/ARD_A5 Arduino_A5/SCL STBC-I2C-SCL1
Shield Boards:
1. Arduino_: Arduino UNO R3
2. MC_: FRDM_MC_LVBLDC and FRDM_MC_LVPMSM
3. STBC_: FRDM-STBC-AGM01
4. TOUCH_: FRDM-TOUCH

Figure 25. 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 22 describes such signals.

Table 23. Arduino compatible header J1 pinout

Pin number	Device pin / GPIO	Default function	Secondary function	Tertiary function	Potential conflict
1	P4_5	GPIO	FC2_RTS	-	M.2 connector (J12) pin 36
2	P4_3	FC2_UART_RXD	DAC1_OUT	-	• MikroBUS socket (J22) pin 3 • M.2 connector (J12) pin 20
3	P4_13	GPIO	Motor control (Opamp / SINC / ENC)	FlexIO	FlexIO header (J20) pin 18
4	P4_2	FC2_UART_TXD	DAC0_OUT	-	• MikroBUS socket (J22) pin 4 • M.2 connector (J12) pin 32

Table 23. Arduino compatible header J1 pinout*...continued*

Pin number	Device pin / GPIO	Default function	Secondary function	Tertiary function	Potential conflict
5	NC	-	-	-	-
6	P0_29	GPIO / PWM	-	-	• 6-pin connector (J21) for PMIC MODESEL0 • NMI push-button (SW2)
7	NC	-	-	-	-
8	P1_23	GPIO / PWM	-	-	MikroBUS socket (J22) pin 1
9	NC	-	-	-	-
10	P0_30	GPIO / PWM	-	-	-
11	NC	-	-	-	-
12	P1_21	GPIO	ENET0_MDIO	SAI1_MCLK	• SAI1_MCLK • Ethernet PHY (ENET0_MDIO)
13	P4_12	-	-	-	• USB0 VBUS DET • FlexIO header (J20) pin 17
14	P1_2	GPIO	-	-	-
15	ANA_14	-	-	-	-
16	P0_31	-	-	-	VREFO

Table 24. Arduino compatible header J2 pinout

Pin number	Device pin / GPIO	Default function	Secondary function	Tertiary function	Potential conflict
1	ADC1_B0	ADC	-	-	-
2	P0_28	GPIO	USB0_ID	-	-
3	P5_2	ADC	-	-	P0_14 (R949)
4	P0_10	GPIO / PWM	-	-	MikroBUS socket (J22) pin 2
5	P5-3	ADC	-	-	P0_22 (R951)
6	P0_27	GPIO	FC1_SPI_PCS		• MikroBUS socket (J19) pin 3 • MCU-Link SPI PCS
7	P5_4	ADC	-	-	-
8	P0_24	GPIO	FC1_SPI_SDO		• MikroBUS socket (J19) pin 6 • MCU-Link SPI SDO
9	P0_3	CMP1_IN1	JTAG_TDI	-	• P5_5 (R948) • IF_TDI (JP9)
10	P0_26	GPIO	FC1_SPI_SDI	-	• MikroBUS socket (J19) pin 5 • MCU-Link SPI SDI
11	P0_18	TSI	PDM		External DMIC header (J14) pin 6, R925 resistor
12	P0_25	GPIO	FC1_SPI_SCK		• MikroBUS socket (J19) pin 4 • MCU-Link SPI SCK
13	P0_19	TSI	-	-	-

Table 24. Arduino compatible header J2 pinout*...continued*

Pin number	Device pin / GPIO	Default function	Secondary function	Tertiary function	Potential conflict
14	GND	-	-	-	-
15	P1_1	TSI	-	-	-
16	VDDA	-	-	-	-
17	P1_0	TSI	-	-	-
18	P4_0	FC2_I2C_SDA	-	-	-
19	NC	-	-	-	-
20	P4_1	FC2_I2C_SCL	-	-	-

Table 25. Arduino compatible header J3 pinout

Pin number	Device pin / GPIO	Default function	Secondary function	Tertiary function	Potential conflict
1	P2_0	MC_ENC_B	-	-	-
2	NC	-	-	-	-
3	P1_22	MC_ENC_A	-	-	-
4	VDD_BRD	-	-	-	-
5	P2_3	PWM1_B2	SDHC0_D0	-	• M.2 connector (J12) pin 13 • SD card connector (J25) pin 7
6	Reset	Reset	-	-	-
7	P2_2	PWM1_A2	SDHC0_D1		• M.2 connector (J12) pin 15 • SD card connector (J25) pin 8
8	P3V3	-	-	-	-
9	P2_5	PWM1_B1	SDHC0_CMD	-	• M.2 connector (J12) pin 11 • SD card connector (J25) pin 3
10	P5V0	-	-	-	-
11	P2_4	PWM1_A1	SDHC0_CLK	-	• M.2 connector (J12) pin 9 • SD card connector (J25) pin 5
12	GND	-	-	-	-
13	P2_7	PWM1_B0	SDHC0_D2		• M.2 connector (J12) pin 17 • SD card connector (J25) pin 1
14	GND	-	-	-	-
15	P2_6	PWM1_A0	SDHC0_D3		• M.2 connector (J12) pin 19 • SD card connector (J25) pin 2
16	P5-9V_VIN	-	-	-	-

Table 26. Arduino compatible header J4 pinout

Pin number	Device pin / GPIO	Default function	Secondary function	Tertiary function	Potential conflict
1	P4_12	CURA.P	FlexIO	GPIO	• USB0_VBUS_DET

Table 26. Arduino compatible header J4 pinout*...continued*

Pin number	Device pin / GPIO	Default function	Secondary function	Tertiary function	Potential conflict
					• FlexIO header J20, pin 17
2	ANA_0 / ADC0_A0	-	-	-	-
3	OPAMP0_INN	CURA.N	-	-	-
4	ANA_1 / ADC0_B0	-	-	-	-
5	P4_16	CURB.P	FlexIO	GPIO	• USB1_OTG_PWR • FlexIO header J20, pin 21
6	P0_14	ADC	-	-	Arduino J2 connector pin 3 (R949)
7	OPAMP1_INN	CURB.N	-	-	-
8	P0_22	ADC	-	-	Arduino J2 connector pin 5 (R951)
9	P4_20	CURC.P	FlexIO	-	FlexIO header J20, pin 25
10	P0_15	ADC	-	-	-
11	OPAMP2 INN	CURC.N	-	-	-
12	P0_23	ADC	-	-	-

2.13 FlexIO header

On the MCX-N9XX-EVK board, one 28-pin FlexIO header is provided to support the LCD display and camera applications. Table 27 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, therefore, 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).

Many signals on the FlexIO connector are shared signals (most notably with the M.2 connector). 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. Note that the FlexIO connections are connected by default.

Table 27. FlexIO header J20 pinout

Pin number	Net name	GPIO	Function	Potential conflict
1	P3V3	-	-	-
2	GND	-	-	-
3	I2C2_SCL / SIOC	P4_1	I2C SCL	-
4	I2C2_SDA / SIOD	P4_0	I2C SDA	-
5	LCD_INT	P4_6	Interrupt pin	-
6	LCD_GPIO	P1_11	General-purpose I/O pin	• Accelerometer sensor (I3C1_ PUR)

Table 27. FlexIO header J20 pinout*...continued*

Pin number	Net name	GPIO	Function	Potential conflict
				• Debug trace connector (J11, TRACED3)
7	LCD_RST	P4_7	LCD reset pin	-
8	LCD_DC	P0_7	LCD clock pin	M.2 connector (pin 20)
9	LCD_CS	P0_12	LCD chip select pin	-
10	FXIO0_D1/LCD_WR	P0_9	Data line / LCD write pin	M.2 connector (pin 54)
11	FXIO0_D0/LCD_RD	P0_8	Data line / LCD read pin	M.2 connector (pin 56)
12	LCD_TE	P0_13	LCD control pin	-
13	FXIO0_D16/LCD_D0	P2_8	Data line	-
14	FXIO0_D17/LCD_D1	P2_9	Data line	-
15	FXIO0_D18/LCD_D2	P2_10	Data line	-
16	FXIO0_D19/LCD_D3	P2_11	Data line	-
17	FXIO0_D20/LCD_D4	P4_12	Data line	• Arduino connector J4 pin 1 / J1 pin 13 / P0_21 • USB0_VBUS_DET
18	FXIO0_D21/LCD_D5	P4_13	Data line	Arduino connector J1 pin 3
19	FXIO0_D22/LCD_D6	P4_14	Data line	-
20	FXIO0_D23/LCD_D7	P4_15	Data line	-
21	FXIO0_D24/LCD_D8	P4_16	Data line	• Arduino connector J4 pin 5 / P0_4 • USB1_OTG_PWR
22	FXIO0_D25/LCD_D9	P4_17	Data line	USB1_OTG_OC
23	FXIO0_D26/LCD_D10	P4_18	Data line	-
24	FXIO0_D27/LCD_D11	P4_19	Data line	-
25	FXIO0_D28/LCD_D12	P4_20	Data line	Arduino connector J4 pin 9 / P1_12
26	FXIO0_D29/LCD_D13	P4_21	Data line	-
27	FXIO0_D30/LCD_D14	P4_22	Data line	-
28	FXIO0_D31/LCD_D15	P4_23	Data line	-

2.14 mikroBUS headers

Table 29 and Table 28 describe the pinout of the mikroBUS headers (J19 and J22).

Table 28. J19 header pinout

Pin number	Net Name	GPIO	Functions	Potential conflict
1	AN	ANA_6	Analog pin	-
2	RST	MCU_RESET_B	Reset pin	-
3	CS	P0_27	SPI chip select line	• Arduino connector (J2) pin 6

Table 28. J19 header pinout*...continued*

Pin number	Net Name	GPIO	Functions	Potential conflict
				• MCU-Link SPI PCS
4	SCK	P0_25	SPI clock line	• Arduino connector (J2) pin 12 • MCU-Link SPI SCK
5	MISO	P0_26	SPI slave output line	• Arduino connector (J2) pin 10 • MCU-Link SPI SDI
6	MOSI	P0_24	SPI slave input line	• Arduino connector (J2) pin 8 • MCU-Link SPI SDO
7	3V3	P3V3	3.3 V power line	-
8	GND	GND	Ground	-

Table 29. J22 header pinout

Pin No.	Net name	GPIO	Functions	Potential conflict
1	PWM	P1_23	PWM output line	Arduino connector (J1) pin 8
2	INT	P0_10	Hardware interrupt line	Arduino connector (J2) pin 4
3	RX	P4_3	UART receive line	• M.2 connector (J12) pin 20 • Arduino connector (J1) pin 2
4	TX	P4_2	UART transmit line	• M.2 connector (J12) pin 22 • Arduino connector (J1) pin 4
5	SCL	P4_1	I2C clock line	-
6	SDA	P4_0	I2C data line	-
7	5V0	P5V0	5 V power line	-
8	GND	GND	Ground	-

2.15 Tamper I/O header

The MCX-N9XX-EVK provides a header for the Tamper I/O. This is to demonstrate the security features of the MCX N94x device.

Table 30 describes the tamper header (JP42) pinout.

Table 30. Tamper I/O header connections

Pin No.	GPIO	Function	Potential conflict
1	-	VDD_BAT	-
2	-	GND	-
3	P5_6	Tamper 4	-
4	P5_7	Tamper 5	-
5	P5_8	Tamper 6	-
6	P5_9	Tamper 7	-

2.16 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.
- Wakeup times may be faster than expected - Boards with devices marked "PMCXN947" may exhibit faster wakeup 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.

2.17 Board operating conditions

The operating temperature range for the MCX-N9XX-EVK 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 MCX-N9XX-EVK. These on-board implementations are referred to as MCU-Link OB.

The MCX-N9XX-EVK 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 MCX-N9XX-EVK 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 31 summarizes the MCU-Link features supported on the MCX-N9XX-EVK board.

Table 31. 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

Table 31. Supported MCU-Link features*...continued*

Feature	Description
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 (MCX N94X) 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
Energy/power/current/voltage consumption measurement[1] (MCUXpresso IDE only)	Allows onboard measurement of current drawn by the target MCU

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

3.3 Supported debug scenarios

In the MCX-N9XX-EVK board, the MCU-Link debug probe target can be either the MCX N94X MCU or an external target compliant with MCU-Link. The board also allows to use an external debugger for debugging the MCX N94X MCU, in place of the MCU-Link debug probe.

Table 32 describes the debug scenarios supported on the MCX-N9XX-EVK board.

Table 32. Supported debug scenarios

Debug scenario	Feature support	Jumper / connector settings
Use MCU-Link as a debugger for the target MCU (MCX N94X)	SWD is enabled	JP6 must be open
	VCOM is enabled	JP19 must be open
	USBSIO is enabled	JP5 must be open
	Target selection supported	JP22 must be open
	Target power selection supported	JP12 Pin 1-2 must be shorted
Use an external debugger to debug the target MCU (MCX N94X)	SWD is disabled	• JP6 must be shorted • Connect an external debugger to the target MCU SWD connector J11
	VCOM is enabled	JP19 must be open
	USBSIO is enabled	JP5 must be open
	Target selection supported	JP22 must be open
	Target power selection supported	JP12 Pin 1-2 must be shorted
Use MCU-Link as a debugger for an external target MCU	SWD is enabled	• JP6 must be open • Connect an external target MCU to the target MCU SWD connector J11
	VCOM is not supported	JP19 must be shorted
	USBSIO is not supported	JP5 must be shorted

Table 32. Supported debug scenarios...*continued*

Debug scenario	Feature support	Jumper / connector settings
	Target selection supported	JP22 must be shorted
	Target power selection supported	• JP12 Pin 1-2 shorted: external target MCU gets power from the board • JP12 Pin 2-3 shorted: external target MCU uses its own power

3.4 MCU-Link firmware update utility installation

The MCU-Link debug probe is supported on a host computer running a Windows 10/11, MacOS X, or Ubuntu Linux operating system (OS). The debug probe works with standard OS drivers. For Windows, the MCU-Link firmware installation program also includes information files to provide user-friendly device names.

Support for MCU-Link can be enabled using the LinkServer utility, which is an NXP GDB server and flash utility that supports many NXP debug probes. For more details on this utility, visit the https://nxp.com/linkserver page.

Running the LinkServer installer also installs a firmware update utility and the drivers (information files) required for MCU-Link. NXP recommends using the LinkServer installer for installing the MCU-Link firmware update utility, unless you are using MCUXpresso IDE version 11.6.1 or earlier.

Note: *To use MCU-Link with MCUXpresso IDE version 11.6.1 or earlier, you need MCU-Link firmware update utility version 2.263 (not included in the LinkServer installer). For Linux OS, MCU-Link installation package 2.263 is available for download at the following link:*

https://www.nxp.com/design/design-center/software/development-software/mcuxpresso-software-and-tools-/mcu-link-debug-probe:MCU-LINK#design-resources

Note: *If the MCU-Link firmware version is 3.122 or later, an automatic firmware update can be done using LinkServer installer version 1.4.85 or later. For more details on automatic firmware update, refer to the Readme mark-down file in the LinkServer installation package. However, if the current firmware version is earlier than 3.122, you have to run manually the MCU-Link firmware update utility, which is included in the LinkServer installation package. To update the MCU-Link firmware using the firmware update utility, see Section 3.5.*

To work with MCU-Link, NXP recommends using the latest MCU-Link firmware. The steps to update the MCU-Link firmware manually are provided in Section 3.5. Before updating the MCU-Link firmware, check the versions of the MCUXpresso IDE and LIBUSBIO (if you are using these tools) installed on your host computer. Then, check the compatibility of these tools with the MCU-Link firmware using Table 33. If you are using the MCUXpresso for Visual Studio Code extension or a third-party IDE from IAR or Keil, NXP recommends using the latest MCU-Link firmware version.

Table 33. 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
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 using firmware update utility

To update the MCU-Link firmware using the firmware update utility included in the LinkServer installation package, the 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 JP24, and reconnect the board. The red MCU-Link status D10 LED lights up and stays on. For the D10 LED details, see Section 3.11.
2. Download the LinkServer installation package from https://nxp.com/linkserver and install the LinkServer utility. For example, download and install "Linkserver 1.4.85 installer for Windows".
3. Navigate to the `MCU-LINK_installer_Vx_xxx` directory, where `Vx_xxx` indicates the version number, for example, `V3.117`.
4. Follow the instructions in the `readme.txt` to find and run the firmware update utility for the CMSIS-DAP or J-Link firmware version.
5. Disconnect the board from the host computer, open jumper JP24, 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 33 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 MCX-N9XX-EVK board has a universal serial bus (USB) 2.0 micro-B connector J5 (Hirose Electric ZX62D-B-5PA8(30)). 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 J5 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.

In the MCX-N9XX-EVK board, MCU-Link LPC55S69 is connected to the P1_8 and P1_9 pins of the target MCU through a voltage translator U21.

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

The voltage translator U21 enables communication between MCU-Link and the target MCU, by shifting voltage levels of signals between the two devices from 3V3 to VDD_BRD and vice versa.

To use MCU-Link as a USB-to-UART bridge, ensure that the JP19 jumper is open and connect the J5 connector on the board to the USB port of the host computer.

When you boot the MCX-N9XX-EVK 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.

The P1_8 and P1_9 pins are shared with the trace debug header (J11). If the trace functionality through these pins is desired, ensure the following resistor and jumper configuration:

- Resistors R19 and R20 are populated
- Resistors R17 and R18 are depopulated
- Jumper JP19 is shorted (to disable the VCOM feature)

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 MCX-N9XX-EVK board, MCU-Link is connected to the P0_[27:24] pins of the target MCU using the FC1 SPI interface connection, through a voltage translator U22. The voltage translator enables the communication between MCU-Link and the target MCU, by shifting voltage levels of signals between the two devices from 3V3 to VDD_BRD and vice versa.

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 J5 connector. Also, ensure the following resistor and jumper configuration on the board:

- Resistors R11, R12, R13, and R14 are populated
- Jumper JP5 is open (to enable the USBSIO bridge feature for SPI; by default JP5 is shorted)

In the MCX-N9XX-EVK board, MCU-Link is also connected to the P4_[1:0] pins of the target MCU using the FC2 I2C interface connection, through a voltage translator U23. The voltage translator enables communication between MCU-Link and the target MCU, by shifting voltage levels of signals between the two devices from 3V3 to VDD_P4 and vice versa.

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 J5 connector. Also, ensure the following resistor and jumper configuration on the board:

- Zero-ohm resistors R46 and R47 are populated
- Jumper JP25 is shorted (to enable USBSIO bridge feature for I2C)

The USBSIO feature can be disabled for SPI or I2C so that the target MCU SPI/I2C port can be used for other purposes. Disabling this feature instructs the firmware not to enumerate the USB endpoint for USBSIO (which is called “MCU-Link LPCSIO” for backward compatibility reasons). Disabling the USBSIO feature also frees more USB bandwidth for the SWO profiling and energy measurement features of MCU-Link.

The USBSIO feature can be disabled for I2C by opening jumper JP25.

3.10 Measuring target MCU power consumption

The MCX-N9XX-EVK board includes circuitry to measure voltage, current, and power/energy consumption for the target MCU (MCX N94x) using the onboard MCU-Link debug probe. This measurement data can be analyzed and displayed using MCUXpresso IDE version 11.5.1 or later. For more details on the MCUXpresso IDE, see https://www.nxp.com/mcuxpresso/ide.

It is recommended to use the latest MCU-Link firmware. If the latest firmware version is not installed, then the MCUXpresso IDE shows a message indicating that the firmware version is not the latest one. For instructions on updating MCU-Link firmware, see Section 3.5.

The energy monitor circuit used on the MCX-N9XX-EVK board measures the current of all supplies on the MCU_PWR net (only critical supplies are sourced by MCU_PWR). The following is the list of power supplies for which current can be measured:

- Core supplies (DCDC_CORE input via VDD_DCDC net or LDO_CORE input via VDD_CORE net)
- System supply (LDO_SYS input via VDD_LDO_SYS_IN net, provided it is not bypassed by jumper JP36)
- VDD via VDD_MCU net
- VBAT Domain via VDD_BAT net

This allows for the measurement of all necessary power rails to be measured and allows for the measurement of DCDC and LDO power modes.

Note: *The above power rails must be at the same voltage level to be included in the measurement circuit.*

To use the Energy Monitor circuit, jumpers JP33, JP32, and JP38 must be populated and JP29 must be depopulated. JP38 can be used to select either P3V3 or P1V8 nets to power MCU_PWR. This allows for 3.3 V or 1.8 V measurements to be made.

For more information on the Energy Monitor circuit and its functionality, see the MCU-Link Pro standalone debug probe user manual.

3.11 MCU-Link status LEDs

The MCX-N9XX-EVK board has seven status indicator LEDs for MCU-Link. Table 34 lists these LEDs and describes how each LED behaves in different MCU-Link modes. These LEDs are shown in Figure 3.

Table 34. 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
D8	USB COMM / green	Lights up after successful USB enumeration at startup. Afterward, the LED stays ON.	Remains OFF	Remains OFF
D9	SWO ACT / green	Indicates if serial wire debug trace output (SWO) data is being received from the target MCU	Remains OFF	Remains OFF
D10	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
D11	VCOM ACT / green	Indicates if the VCOM port is transmitting/receiving data	Lights up when MCU-Link boots, and blinks when debug activity happens	Remains OFF
D12	FUNC / green	Reserved for future use	Remains OFF	Remains OFF
D13	SIO / green	Indicates if USBSIO bridge traffic is present	Remains OFF	Remains OFF
D14	NRG / green	Indicates if energy measurement communication is happening between the target MCU and MCU-Link	Remains OFF	Remains OFF

3.12 MCU-Link GPIO header

Table 35 describes the MCU-Link GPIO header (J13) pinout.

Table 35. MCU-Link GPIO header

Pin number	MCU-Link GPIO
1	P1_1
2	P1_9
3	P1_20
4	P1_21
5	P1_31
6	P1_7

4 Related documentation

Table 36 lists and explains the additional documents and resources that you can refer to for more information on the MCX-N9XX-EVK 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 36. 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_x P02G)	Lists the details of all known silicon errata for the MCX Nx4x device.	Contact NXP FAE or sales representative
MCX-N9XX-EVK design files	A zip file including *.DSN, ASY, Layout, schematic files, and so on	MCX-N9XX-EVK-DESIGNFILES
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 37 lists and defines the acronyms used in this document.

Table 37. Acronyms

Term	Description
ADC	Analog-to-digital converter
CAN	Controller area network
DNP	Do not populate
ESR	Equivalent series resistance
GPIO	General-purpose input/output
I2C	Inter-integrated circuit
I3C	Improved inter-integrated circuit
ISP	In-system programming
LPI2C	Low-power inter-integrated circuit
NMI	Non-maskable interrupt
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

Table 37. Acronyms*...continued*

Term	Description
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 38 summarizes revisions to this document.

Table 38. Revision history

Document ID	Release date	Description
UM12036 v.2	10 July 2024	• Added a note for the bypass capacitor C150 in Table 2 • Updated description in Table 9 • Added a note for J35 description in Section 1.6 • Updated the jumpers detail in Section 3.10 to use MCU-Link as the Energy Monitor circuit • Updated Section 3.4 • Updated to add LinkServer detail in Section 3.5 • Updated Table 36
UM12036 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.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

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.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

IAR — is a trademark of IAR Systems AB.

J-Link — is a trademark of SEGGER Microcontroller GmbH.

MCX — is a trademark of NXP B.V.

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.