UM12041

FRDM-MCXN236 Board User Manual

Rev. 2 — 15 May 2024 User manual

Document information

Information	Content
Keywords	FRDM-MCXN236, UM12041, MCXN236
Abstract	The NXP FRDM-MCXN236 board is a low-cost design and evaluation board based on MCXN236 device. This document describes the hardware for the FRDM-MCXN236 board.

1 FRDM-MCXN236 overview

The NXP FRDM-MCXN236 board is a low-cost design and evaluation board based on the MCXN236 device.

The MCXN236 device integrates a 32-bit Arm Cortex-M33 microcontroller for Industrial and Consumer IoT Applications. NXP supports the MCXN236 device with tools and software that include hardware evaluation boards, software development IDE, example applications, and drivers.

The FRDM-MCXN236 board consists of one MCXN236 device with a 64 Mbit external serial flash provided by Winbond. The board also features a high-speed USB circuit, one FXLS8974CFR3 accelerometer sensor, TJA1057GTK/3Z CAN PHY, DA7212 Audio codec (DNP), DMIC sensor SPK0641HT4H-1, RGB LED, push buttons, and MCU-Link debug probe circuit. The board is compatible with the Arduino shield modules and mikroBUS headers. The board also supports headers compatible with a camera module and NXP low-cost LCD module (PAR-LCD-S035).

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

The board is lead free and RoHS-compliant.

1.1 Block diagram

Figure 1 shows the FRDM-MCXN236 board block diagram.

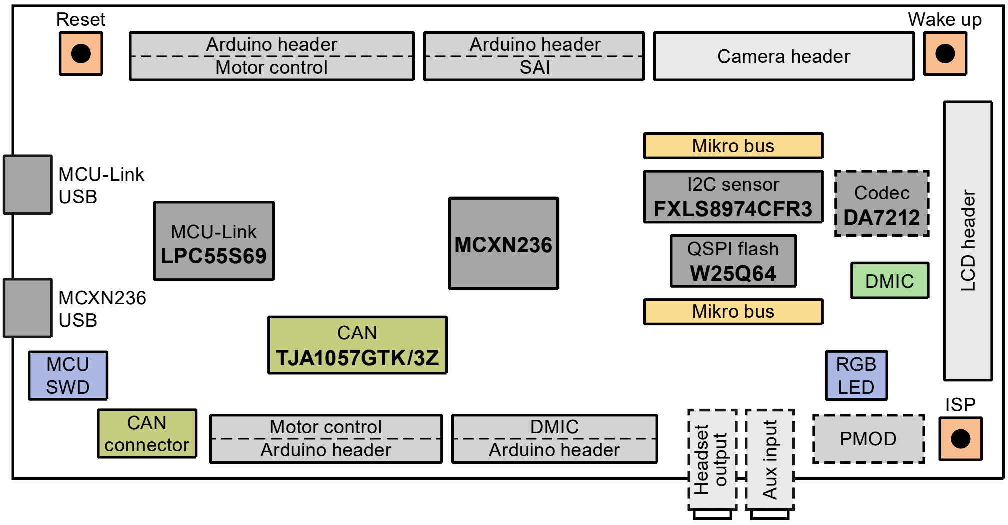


Figure 1. FRDM-MCXN236 block diagram

1.2 Board features

Table 1 describes the features of the FRDM-MCXN236 board.

Table 1. FRDM-MCXN236 features

Board feature	Target MCU features used	Description
MCXN236 MCU (target MCU)		The MCXN236 MCU is based on a 32-bit Arm Cortex-M33 TrustZone microcontroller. For details, see the *MCX N23x Reference Manual*.

Table 1. FRDM-MCXN236 features*...continued*

Board feature	Target MCU features used	Description
Power supply		• P5V0 input power supply selected from one of the following power sources: – High-speed USB2.0 Type-C connector J11 – 5 V regulator populated at 3-pin header J19 – Arduino Shield compatible header J3 – MCU-Link USB2.0 Type-C connector J10 • One LDO for LDO_3V3 power supply • One LDO for VDD_1V8 power supply • Jumpers and resistors configuration for different secondary power supplies
Clock		Crystal oscillators for: • 16 MHz clock input for MCU-Link • 24 MHz system reference clock • 32.768 kHz real-time clock (RTC)
USB	High-speed (HS) USB module	USB2.0 High-speed Device/Host on board with type-C USB connector
Flash memory	Flexcomm (FC6) SPI	Supports one Winbond W25Q64JVSSIQ - 64 Mbit QSPI flash memory (populated)
Sensor	Flexcomm (FC2) I2C	Supports NXP FXLS8974CFR3 accelerometer sensor
	ADC	Supports an ambient light sensor
I/O headers		Headers compatible with: • Arduino shields (outer rows) and FRDM header (inner rows) • Mikroe click boards • Camera module • LCD module • Peripheral module (Pmod)
CAN	FlexCAN	One TJA1057GTK/3Z CAN PHY with FD support and one 4-pin CAN header to connect to other boards
Audio	SAI1	• One Audio codec DA7212 (DNP) • One digital microphone (SPK0641HT4H-1) • One Auxiliary input port and one headset output port
LED		• One RGB LED D13 for user interface • One Green LED D3 for the Target MCU power status • One Red LED D12 for Reset status
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		118 mm x 55 mm
Orderable part number		FRDM-MCXN236

1.3 Board kit contents

The FRDM-MCXN236 board kit contains the following items:

- FRDM-MCXN236 board hardware assembly
- One 3 ft micro USB Type A to USB Type C cable

1.4 Board pictures

Figure 2 shows the top view of FRDM-MCXN236.

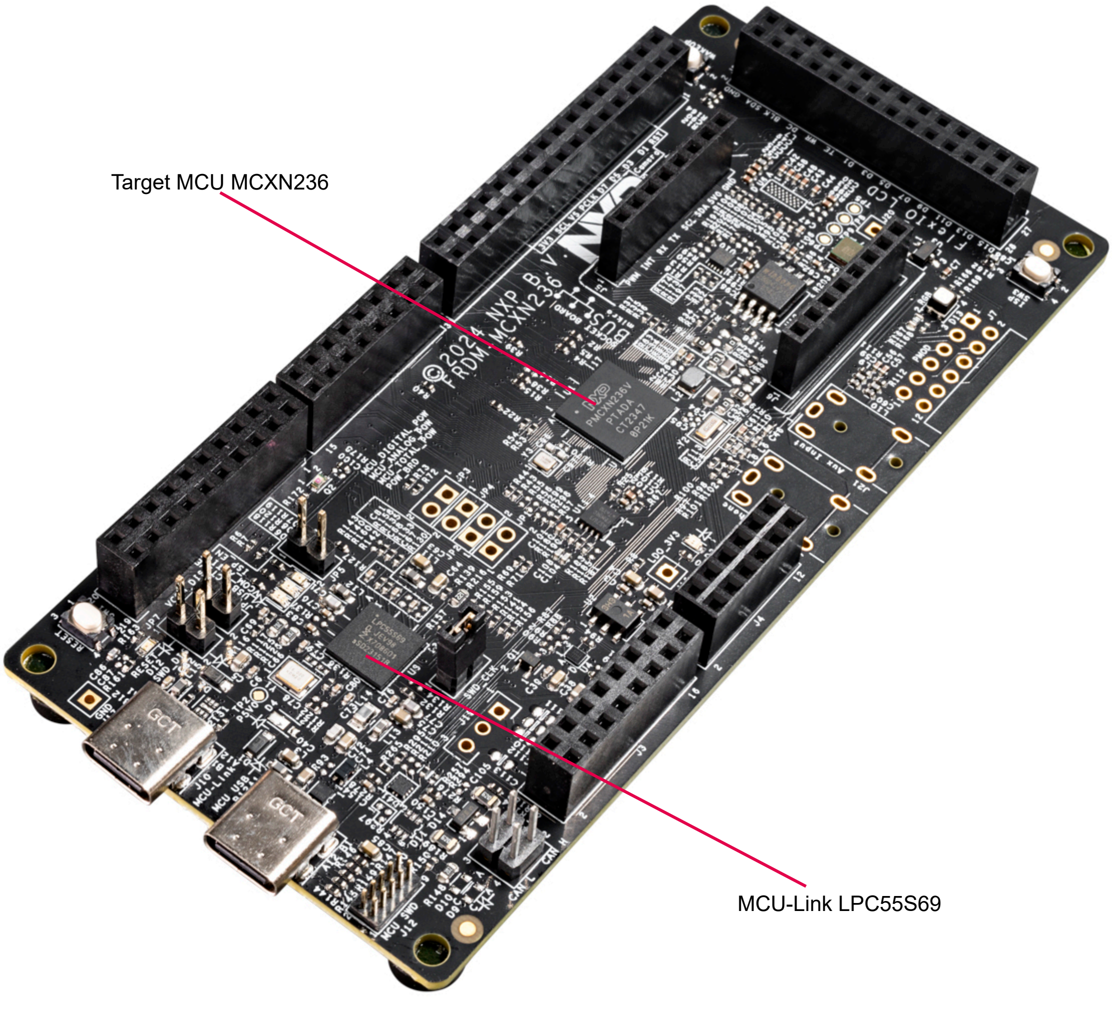


Figure 2. FRDM-MCXN236 top view

Figure 3 shows the top-side view of the FRDM-MCXN236 board, with connectors and push buttons highlighted.

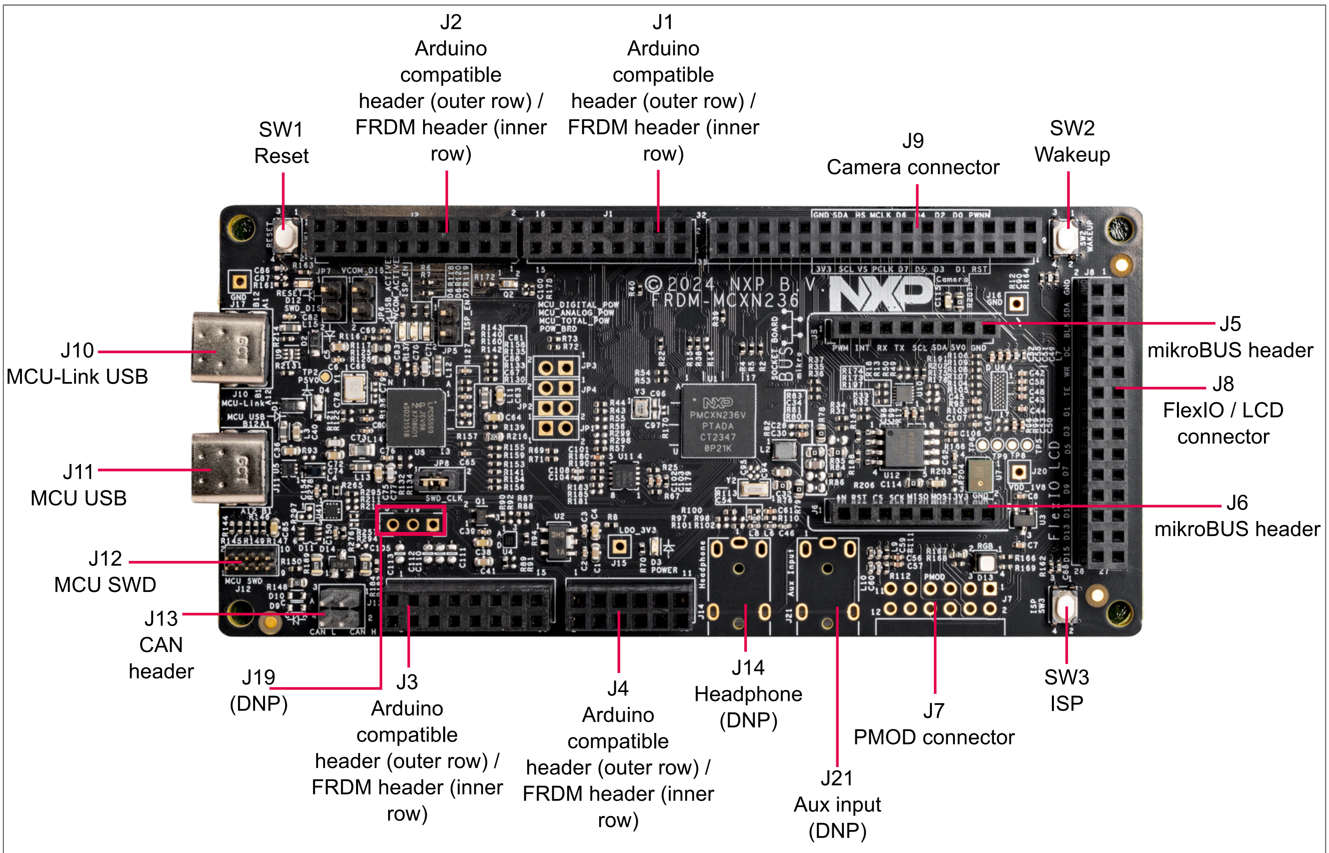

J2
Arduino compatible header (outer row) / FRDM header (inner row)
J1
Arduino compatible header (outer row) / FRDM header (inner row)
SW1
Reset
J9
Camera connector
SW2
Wakeup
J10
MCU-Link USB
J5
mikroBUS header
J8
FlexIO / LCD connector
J11
MCU USB
J6
mikroBUS header
J12
MCU SWD
J13
CAN header
J19
(DNP)
J3
Arduino compatible header (outer row) / FRDM header (inner row)
J4
Arduino compatible header (outer row) / FRDM header (inner row)
J14
Headphone
(DNP)
J7
PMOD connector
SW3
ISP
J21
Aux input
(DNP)

Figure 3. FRDM-MCXN236 connectors and push buttons (top-side view)

Figure 4 shows the jumpers and LEDs on the FRDM-MCXN236 board.

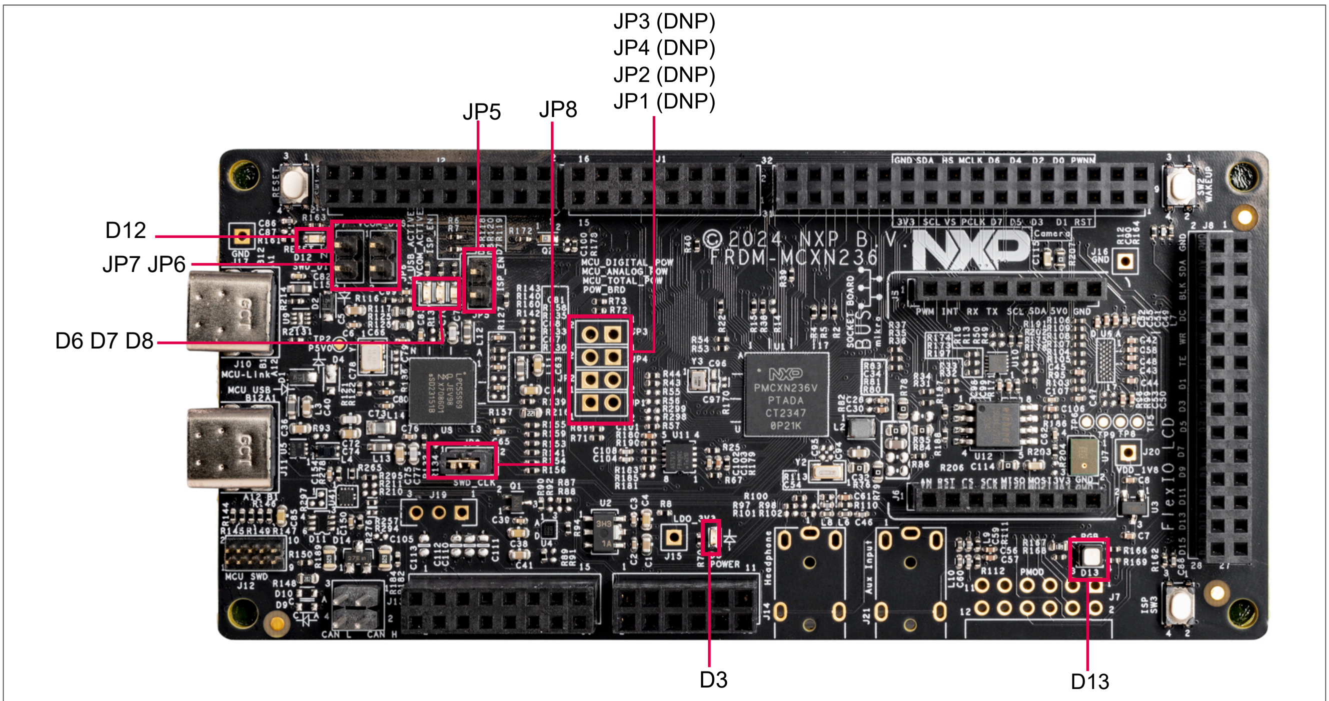


Figure 4. FRDM-MCXN236 jumpers and LEDs

Figure 5 shows the bottom view of the FRDM-MCXN236 board.

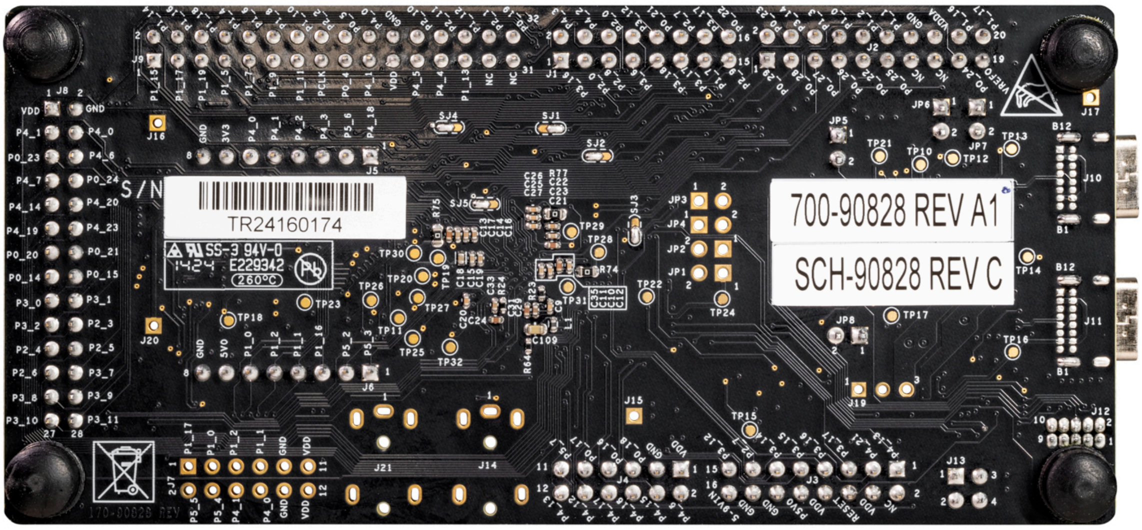


Figure 5. FRDM-MCXN236 bottom view

1.5 Connectors

Table 2 describes the FRDM-MCXN236 connectors. The connectors are shown in Figure 3.

Table 2. FRDM-MCXN236 connectors

Part identifier	Connector type	Description	Reference section
J1	2 x 8 pin header	Arduino compatible I/O header (outer rows) and FRDM header (inner rows)	Section 2.9
J2	2 x 10 pin header		
J3	2 x 8 pin header		
J4	2 x 6 pin header		
J5	1 x 8 position receptacles	mikroBUS socket connector	Section 2.11
J6	1 x 8 position receptacles	mikroBUS socket connector	Section 2.11
J7	2 x 6 header (DNP)	Pmod connector	Pmod header
J8	2 x 14 pin header	FlexIO header for LCD screen connection	Section 2.10
J9	2 x 16 pin header	Used for connecting a camera module	Section 2.12
J10	USB Type-C connector	MCU-Link USB connector	Section 3.7
J11	USB Type-C connector	MCU USB high-speed connector	Section 2.3
J12	2 x 5 pin header	Debug (JTAG / SWD) connector to connect an external debug probe or external debug target	Section 3.8
J13	2 x 2 pin header	Connects to the CAN bus and allows external connection with the bus	Section 2.4
J14 (DNP)	54-00174 jack	3.5 mm headphone jack	Section 2.8
J19 (DNP)	3-pin header	For P5V_HDR_IN output supply	Section 2.1
J21 (DNP)	54-00174 jack	3.5 mm auxiliary input jack	Section 2.8

1.6 Jumpers

Table 3 describes the FRDM-MCXN236 jumpers. The jumpers are shown in Figure 4.

Table 3. FRDM-MCXN236 jumpers

Part identifier	Jumper type	Description	Reference section
JP1 (DNP)	1x2 pin header	Allows for inserting an ammeter for VDD_BOARD current measurement	For more information on this jumper, see the FRDM-MCXN236 board schematic
JP2 (DNP)	1x2 pin header	Allows for inserting an ammeter for P3V3_MCU current measurement	For more information on this jumper, see the FRDM-MCXN236 board schematic
JP3 (DNP)	1x2 pin header	Allows for inserting an ammeter for VDD_MCU current measurement	For more information on this jumper, see the FRDM-MCXN236 board schematic
JP4 (DNP)	1x2 pin header	Allows for inserting an ammeter for VDDA_MCU current measurement	For more information on this jumper, see the FRDM-MCXN236 board schematic
JP6	1x2 pin header	• Open (**default setting**): MCU-Link VCOM port is enabled. • Shorted: Sends a low signal on LPC_HW_VER_6 to disable MCU-Link VCOM port	Section 3.8
JP7	1x2 pin header	• Open (**default setting**): Enables the MCU-Link SWD feature • Shorted: Sends a low signal on LPC_HW_VER_7 to disable the onboard MCU-Link SWD feature ***Note:*** *This configuration is required to enable target MCU debug through an external debug probe.*	Section 3.3
JP5	1x2 pin jumper	MCU-Link (LPC55S69) ISP mode enable jumper: • Open (**default setting**): MCU-Link follows the normal boot sequence (MCU-Link boots from internal flash if a boot image is found). With the internal flash erased, the MCU-Link normal boot sequence falls through to ISP boot mode. • Shorted: MCU-Link is forced to ISP mode (USB). Use this setting to reprogram the MCU-Link internal flash with a new image or use the MCUXpresso IDE with the CMSIS-DAP protocol. ***Note:*** *By default, MCU-Link flash is preprogrammed with a version of CMSIS-DAP firmware.*	Section 3.5
JP8	1x2 pin jumper	MCU-Link SWD clock enable jumper: • Open: MCU-Link SWD clock is disabled. • Shorted **(default setting)**: MCU-Link SWD clock is enabled.	For more information on this jumper, see the FRDM-MCXN236 board schematic

1.7 Push buttons

Push buttons are populated on the FRDM-MCXN236 board for human machine interaction (HMI).

Table 4 describes the FRDM-MCXN236 push buttons. The push buttons are shown in Figure 3.

All information provided in this document is subject to legal disclaimers.

Table 4. FRDM-MCXN236 push buttons

Part identifier	Switch name	Description
SW1	Reset button (MCXN236 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 D12 turns ON.
SW2	Wakeup button	SW2 connects to the P0_20 pin of the target MCU. Pressing SW2 gives a low level on P0_20/WUU0_IN4-SW, otherwise, it is a high level on P0_20/WUU0_IN4-SW.
SW3	In-system programming (ISP) mode switch	SW3 is an ISP mode switch and can also act as a general-purpose input. Pressing SW3 gives a low level on P0_6/ISPMODE_N-SW, otherwise, it is a high level on P0_6/ISPMODE_N-SW.

Figure 6 shows the circuit diagrams of the FRDM-MCXN236 push buttons.

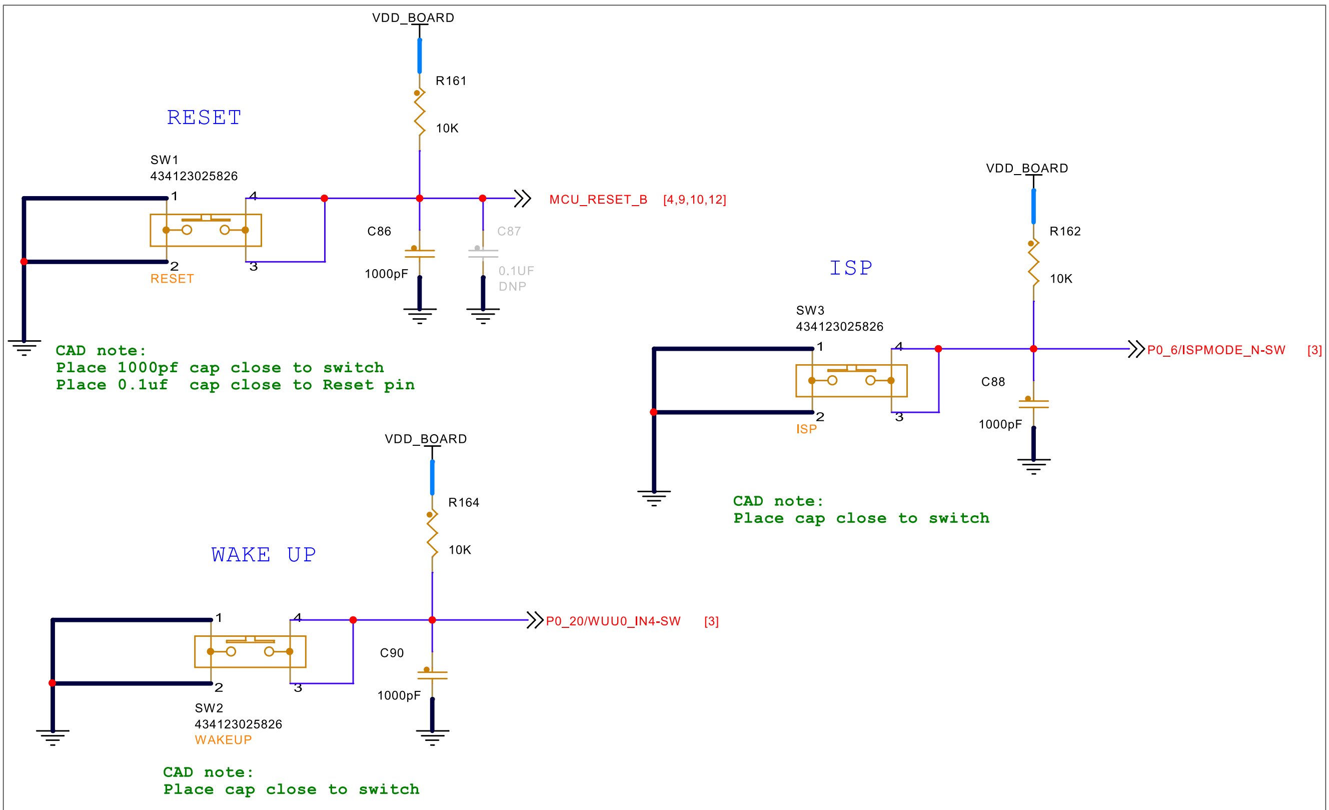


Figure 6. Push button circuit diagram

1.8 LEDs

Table 5 describes the FRDM-MCXN236 light-emitting diodes (LEDs) that correspond to the target MCU. The board also has some MCU-Link-specific LEDs, which are described in Section 3.10. The LEDs are shown in Figure 3.

Note: *The FRDM-MCXN236 board also has three status indicator LEDs for MCU-Link. For details, see Section 3.10.*

Table 5. FRDM-MCXN236 LEDs

Part identifier	LED color	LED name / function	Description
D12	Red	Reset LED	Indicates system reset activity. When board reset is initiated, for example, by pressing the SW1 reset button, the D12 LED turns ON.
D13	Red/green/blue	RGB LED	User application LEDs. Each of these LEDs can be controlled through a user application. • Red LED connects to target MCU pin P4_18 • Green LED connects to target MCU pin P4_19 • Blue LED connects to target MCU pin P4_17
D3	Green	P3V3 PWR ON	Indicates LDO_3V3 power status. When LDO_3V3 is available on board, D3 turns ON.

Figure 7 shows the circuit diagram of the RGB LEDs described in Table 5.

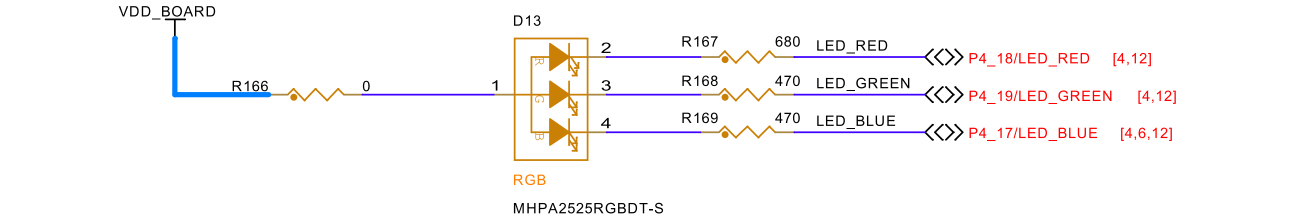


Figure 7. RGB LED circuit diagram

2 FRDM-MCXN236 functional description

This section describes the features and functions of the FRDM-MCXN236 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 MCXN236 MCU, see MCX N23x Product Family Data Sheet and MCX N23x Reference Manual.*

2.1 Power supplies

The FRDM-MCXN236 board is powered with a P5V0 (5 V) power supply. The power source of P5V0 is SYS_5V0, which is powered using one of the following source options:

- P5V_USB_HS supply from high-speed (HS) USB2.0 Type-C connector (J11)
- P5V_HDR_IN supply from 5 V regulator populated at 3-pin header (J19) (Not populated by default)
- P5V_MCU_LINK_USB supply from MCU-Link USB2.0 Type-C connector (J10)

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

Other power supplies in the FRDM-MCXN236 board are through voltage regulators. Some of the power supplies can be enabled or disabled through jumper connections on the board.

Section 2.1 shows the system power circuit on the FRDM-MCXN236 board.

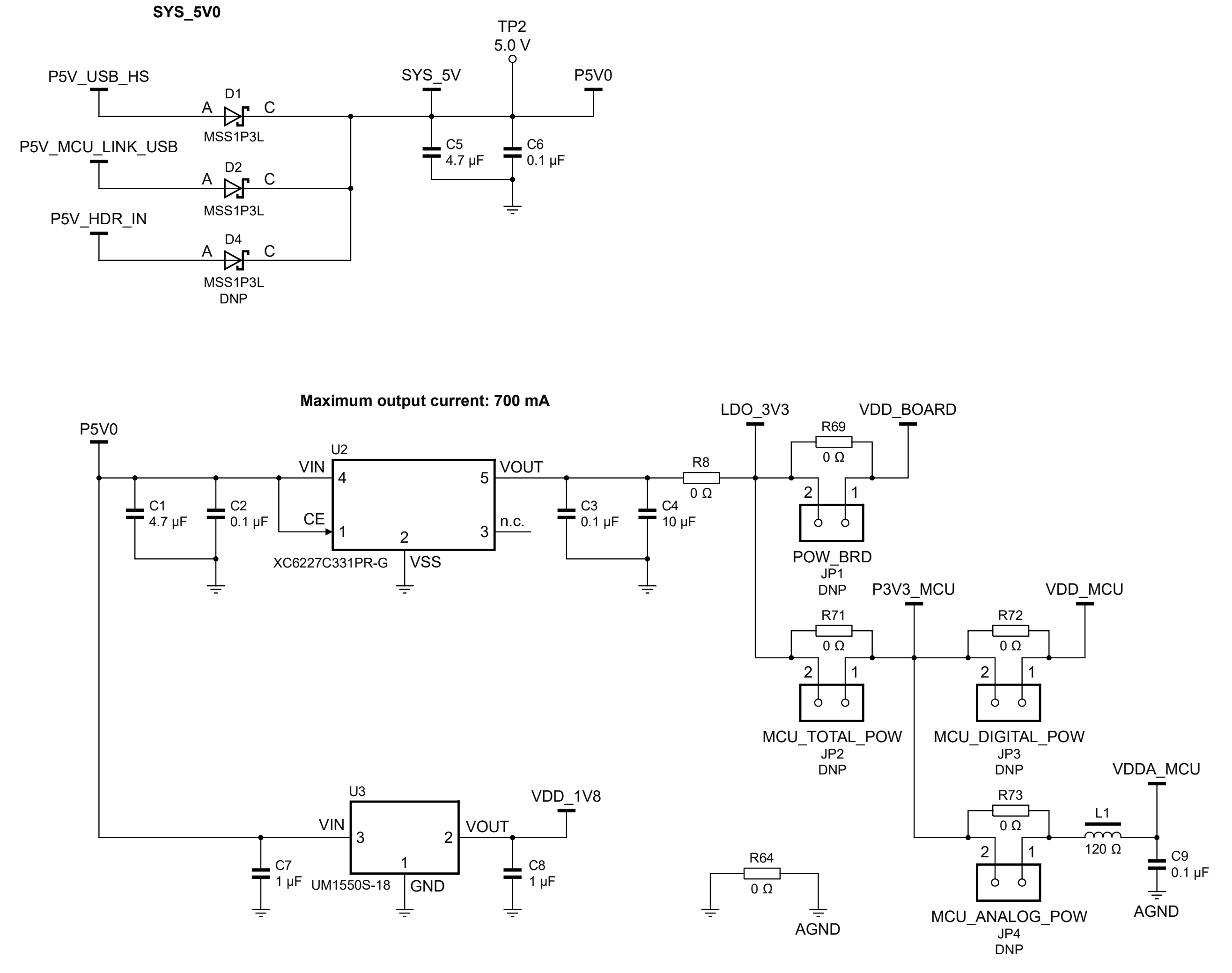


Figure 8. FRDM-MCXN236 system power circuit diagram

5 V power sources and selection

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

Table 6. 5 V power sources

Part identifier	Device / power source	Output power supply	Description
J10	MCU-Link USB2.0 Type-C connector	P5V_MCU_LINK_USB	• One of the sources of SYS_5V0 supply **(default option)** • USB regulator input power supply for MCU-Link microcontroller LPC55S69
J11	HS USB2.0 Type-C connector	P5V_USB_HS	One of the sources for SYS_5V0 supply
J19	5 V power regulator populated at J19	P5V_HDR_IN	One of the sources for SYS_5V0 supply
-	P5V_MCU_LINK_USB / P5V_USB_HS / P5V_HDR_IN / J3 (pin 10)[1]	P5V0	Input power supply for: • XC6227C331PR-G LDO voltage regulator (U2) • UM1550S-18 Linear regulator (U3)

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

Part identifier	Device / power source	Output power supply	Description
			• TJA1057 CAN PHY (U11) and CAN 2x2-pin header (J13) • mikroBUS connector (J5) • HS USB connector power switch NX5P3090 UK (U4)

[1] The J3 pin 10 is the onboard 5 V output by default, and can also be used as an external 5 V input option.

3.3 V power sources and selection

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

Table 7. P3V3 power sources

Part identifier	Device / power source	Output power supply	Description
U2	XC6227C331PR-G (TOREX)	LDO_3V3	• One of the power sources for the VDD_BOARD supply either through the zero-ohm resistor R69 (default selection) or through the 2-pin jumper JP1 (DNP). For details, see Section 1.6. • Power source for the P3V3_MCU supply either through the zero-ohm resistor R71 (default selection) or through the 2-pin jumper JP2 (DNP). For details, see Section 1.6.
-	LDO_3V3	VDD_BOARD	• Power supply for: – PTN5150A USB Type-C CC logic – QSPI Flash (U12) – CAN transceiver TJA1057 (U11) – mikroBUS connector (J6) – Pmod connector J7 (DNP) – Parallel camera header (J9) – FLEXIO / LCD connector (J8) – MCU-Link LEDs (D6, D7, and D8) – RGB LED (D13) • Power source for: – VDD_AUDIO supply of the audio codec – MCU_LINK_3V3 and MCULink_VDDA supplies of the MCU-Link LPC55S69 (U8) – VDD_ACCL supply of I2C sensor (FXLS8974CFR3)
-	LDO_3V3	P3V3_MCU	Power source for: • VDD_MCU supply either through the zero-ohm resistor R72 (default selection) or through the 2-pin jumper JP3 (DNP). For details, see Section 1.6. • VDDA_MCU supply either through the zero-ohm resistor R73 (default selection) or through the 2-pin jumper JP4 (DNP). For details, see Section 1.6.

1.8 V power supply

Table 8 describes the device providing 1.8 V supply.

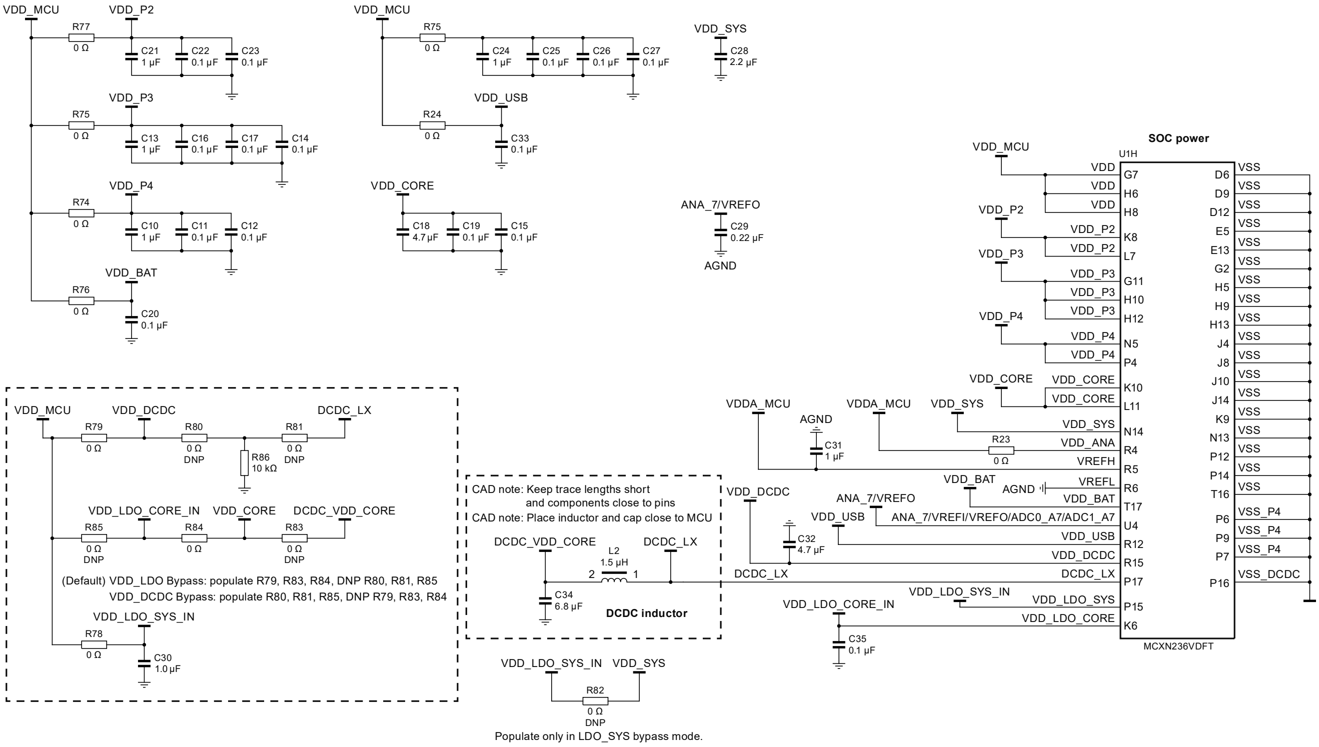
Table 8. 1V8 power supply

Part identifier	Device / power source	Output power supply	Description
U3	UM1550S-18 (Union Semiconductors)	VDD_1V8	Power supply for the Audio codec (U6)

2.1.1 Power supply configuration

Once the main power configurations are set, the target MCU power configurations must be made. The MCU power is configured by a network of jumpers or by a combination of resistors, capacitors, and diodes as shown in Figure 9.

These jumpers provide access to insert ammeters in all the supplies connecting to the MCXN236 device. They also provide a means of connecting external supplies to any of the MCU power pins.



(Default) VDD_LDO Bypass: populate R79, R83, R84, DNP R80, R81, R85
VDD_DCDC Bypass: populate R80, R81, R85, DNP R79, R83, R84

CAD note: Keep trace lengths short and components close to pins
CAD note: Place inductor and cap close to MCU

Populate only in LDO_SYS bypass mode.

Figure 9. Power configuration

The target MCU has many voltage inputs. Table 9 describes the supply options that the board provides for powering each of the MCU voltage inputs. Each supply rail is implemented as an isolated domain to allow for measuring the power consumption of the MCU in all power modes and measuring the individual voltage levels.

Table 9. MCU power supplies

Power source	Zero-ohm resistor / Jumper used	Power supply rail	Description
P3V3_MCU	• R72 resistor (installed) • JP3 jumper (DNP)	VDD_MCU	MCU digital power
	• R73 resistor (installed) • JP4 jumper (DNP)	VDDA_MCU	MCU analog power
VDD_MCU	R85 resistor (DNP)	VDD_LDO_CORE_IN	Input supply for CORE LDO

Table 9. MCU power supplies...*continued*

Power source	Zero-ohm resistor / Jumper used	Power supply rail	Description
VDD_MCU	R77 resistor (installed)	VDD_P2	Power supply for the target MCU port P2 I/O
	R75 resistor (installed)	VDD_P3	Power supply for the target MCU port P3 I/O
	R74 resistor (installed)	VDD_P4	Power supply for the target MCU port P4 I/O
	R76 resistor (installed)	VDD_VBAT	Power supply for RTC, Tamper, and Port 5
	R24 resistor (installed)	VDD_USB	Power supply for USB PHY
	R79 resistor (installed)	VDD_DCDC	Power supply for DCDC
	R78 resistor (installed)	VDD_LDO_SYS_IN	Input supply for SYS_LDO ***Note:*** *By default, VDD_LDO_SYS_IN is sourced from VDD_MCU. However, in LDO_SYS bypass mode, it is sourced from VDD_SYS by populating the R82 resistor.*

Table 10 describes the typical power supply configurations for DCDC_CORE and LDO_CORE operations.

Table 10. Power supply configurations for DCDC_CORE and LDO_CORE operation

Mode	Solder options for resistors					
	R79	R83	R84	R80	R81	R85
• DCDC_CORE enable • LDO_CORE disable **(Default setting)**	Populate	Populate	Populate	DNP	DNP	DNP
• DCDC_CORE disable • LDO_CORE enable	DNP	DNP	DNP	Populate	Populate	Populate

2.1.2 DC-DC inductor

The FRDM-MCXN236 board uses a 1.5 µH DC-DC inductor L2 (SHENZHEN SUNLORD SPH252012H1R5MT). The inductor is enabled when the board is configured in DC-DC Buck mode. Figure 9 shows the DC-DC inductor circuit diagram of the FRDM-MCXN236 board.

Choosing the right DC-DC inductor for your target board is important. When selecting a DC-DC inductor, refer to the specifications mentioned in the *MCX N23x Product Family Data Sheet*.

2.2 Clocks

The FRDM-MCXN236 board provides crystal oscillators to provide accurate time bases for the device and different components on the board.

Table 11 describes the clock sources available on the FRDM-MCXN236 board.

Table 11. FRDM-MCXN236 clocks

Clock generator	Clock frequency	Destination	Description
Crystal oscillator, Y1 (830064296, Wurth Electronics)	16 MHz	XTAL32M_N/P pins of LPC55S69 MCU-Link	Option for external clock input
Crystal oscillator, Y2 (830009678, Wurth Electronics)	32.768 kHz	• Port 5 pin 1 (XTAL32K) of target MCU MCXN236 • Port 5 pin 0 (EXTAL32K) of target MCU MCXN236	For accurate low-power timebase • Internal load capacitors provide the entire crystal load capacitance • To measure the 32.768 kHz oscillator frequency, enable the RTC_CLKOUT signal to be available on the P5_3 pin. It can be observed at pin1 of connector J6
Crystal oscillator, Y3 (830108212309, Wurth Electronics)	24 MHz	• Port 1 pin 30 (XTAL48M) of target MCU MCXN236 • Port 1 pin 31 (EXTAL48M) of target MCU MCXN236	For high-frequency accurate timebase • Required external load capacitors are provided • Small package size (2.0 mm x 1.6 mm) • Low-ESR (100 Ω max) crystal

2.3 USB interface

The target MCU (MCXN236) features one high-speed USB module with device and host capabilities and a built-in transceiver.

On the FRDM-MCXN236 board, the HS USB controller and PHY interface connected to the USB Type-C connector (J11).

Figure 10 shows the high-speed USB circuit diagram.

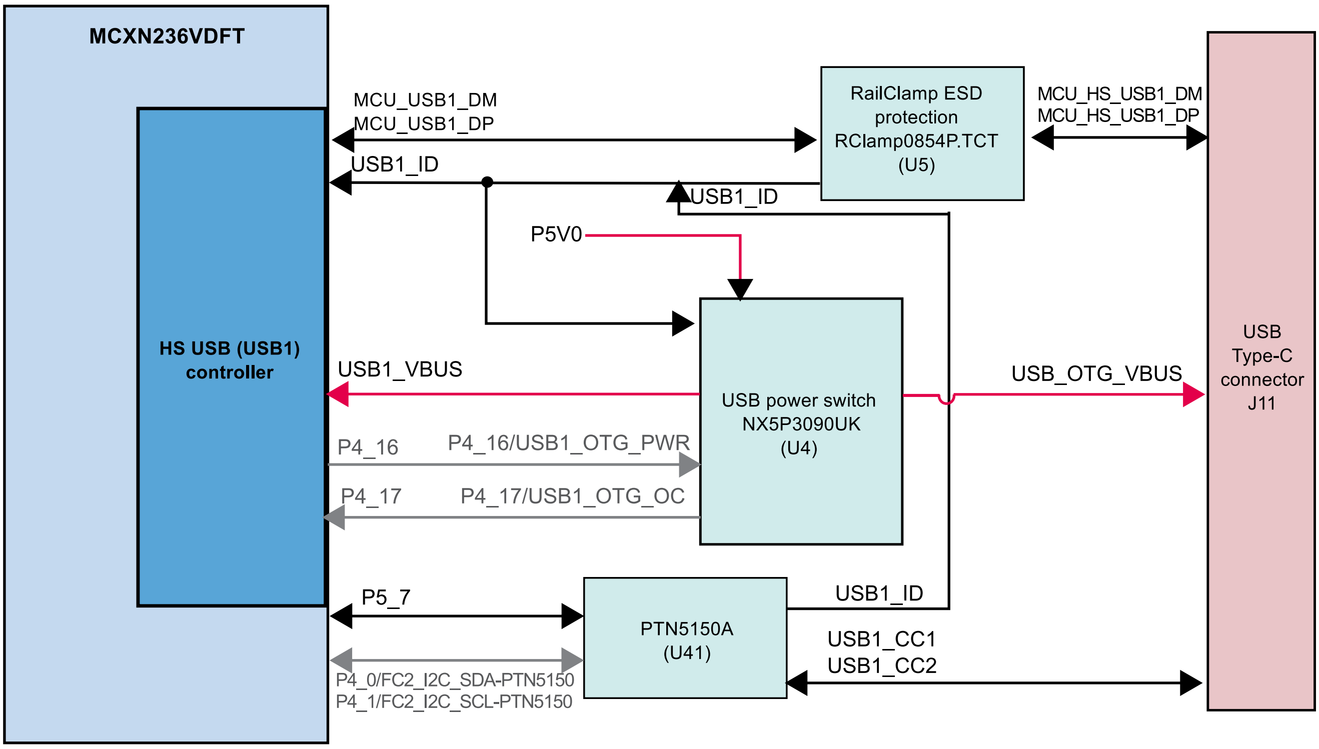


Figure 10. High-speed USB circuit diagram

Table 12 describes the devices used for connection between high-speed USB controller and USB Type-C connector.

Table 12. USB ports

Part identifier	Connector type	Description
J11	USB2.0 Type-C connector	Port can connect in both Host and Device mode. In Device mode, this port provides the 5 V power supply (P5V_USB_HS) to the board.
U4	NX5P3090UK	USB Power Delivery (PD) and type C current-limited power switch
U41	PTN5150A	CC Logic chip supporting the USB Type-C connector application with Configuration Channel (CC) control logic detection and indication functions • ADR/CON_DET pin configuration: – CON_DET=1: Connected detected, I2C address: 0x7A – CON_DET=0: No connection, I2C address: 0x3A – When floating: No I2C function • PORT pin configuration: – When pull up to VDD_BOARD with 10 kΩ resistor (R297 Pin 1-2 selection), PORT=1: DFP mode – When pull down to GND with 10 kΩ resistor (R297 Pin 2-3 selection), PORT=0: UFP mode – When floating **(default setting)**: DRP mode

On the FRDM-MCXN236 board, the MCU_USB1_DP and MCU_USB1_DM signals from the target MCU connect to the onboard USB connector (J11) directly through a common mode choke. The common mode choke is included for noise suppression on the DM / DP signals.

2.4 FlexCAN interface

The flexible 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 (MCXN236) supports two CAN (w/ wo FD) controllers (CAN0 and CAN1).

On FRDM-MCXN236, only the CAN1 controller is used. The CAN1 controller connects to a 4-pin CAN header through a CAN transceiver (TJA1057GTK/3Z). The CAN1_TXD and CAN1_RXD signals are through ports P4_16 and P4_15, respectively.

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

Table 13. High-speed CAN transceiver and header

Part identifier	Manufacturing part number	Description
U11	TJA1057GTK/3Z	High-speed CAN transceiver. It provides an interface between the CAN1 controller and the physical two-wire CAN1 bus.
J13	-	4-pin CAN header. It is connected to the CAN1 bus and allows external connection with the bus.

Figure 11 shows the CAN interface circuit diagram.

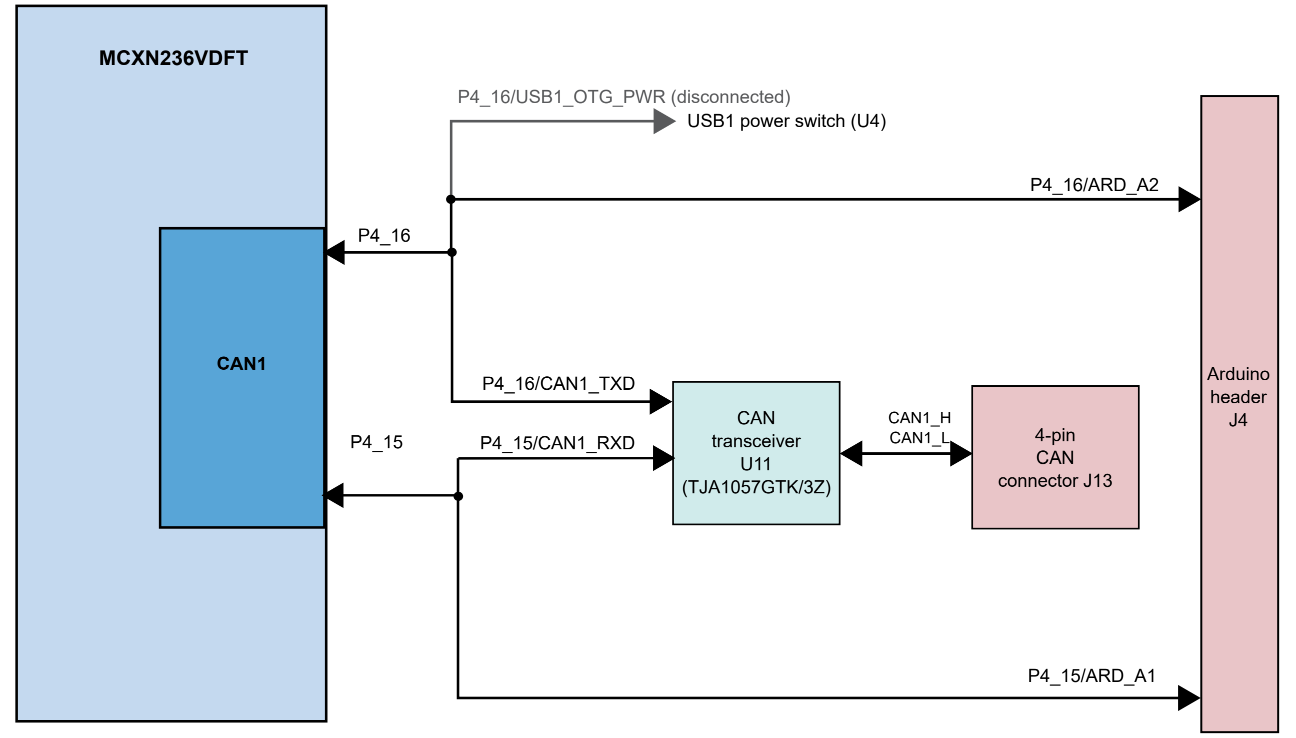


Figure 11. CAN interface circuit diagram

The CAN interface lines are connected via zero ohm resistors to allow the lines to be disconnected from the CAN PHY. These lines are also sharing the connections with other board components (Arduino and USB power switch U4). Series zero ohm resistors (R25 and R67) are provided to isolate the CAN PHY from the MCXN236 device.

Table 14 describes the 4-pin J13 CAN header pinout.

Table 14. CAN header - pinout

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

2.5 Accelerometer sensor interface

On the FRDM-MCXN236 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 FXLS8974CFR3 (U10) is used.
- The sensor device is powered by the VDD_ACCL supply, which is tied to the VDD_BOARD supply through a zero ohm resistor (R171).
- Discrete pull-up resistors for the I2C bus lines are provided.
- The default 8-bit I2C address for the device is configured as 0x30. Address can be changed by pull-up / pull-down resistors on the SA0 line.
  - SA0:0 → 8-bit I2C read address: 0x31, 8-bit I2C write address: 0x30
  - SA0:1 → 8-bit I2C read address: 0x33, 8-bit I2C write address: 0x32
- The I2C uses shared lines for the I2C interface.
- Series zero ohm resistors (R49 and R50) are provided to isolate the sensor from the MCXN236 device.

Figure 12 shows the FXLS8974CFR3 sensor circuit diagram.

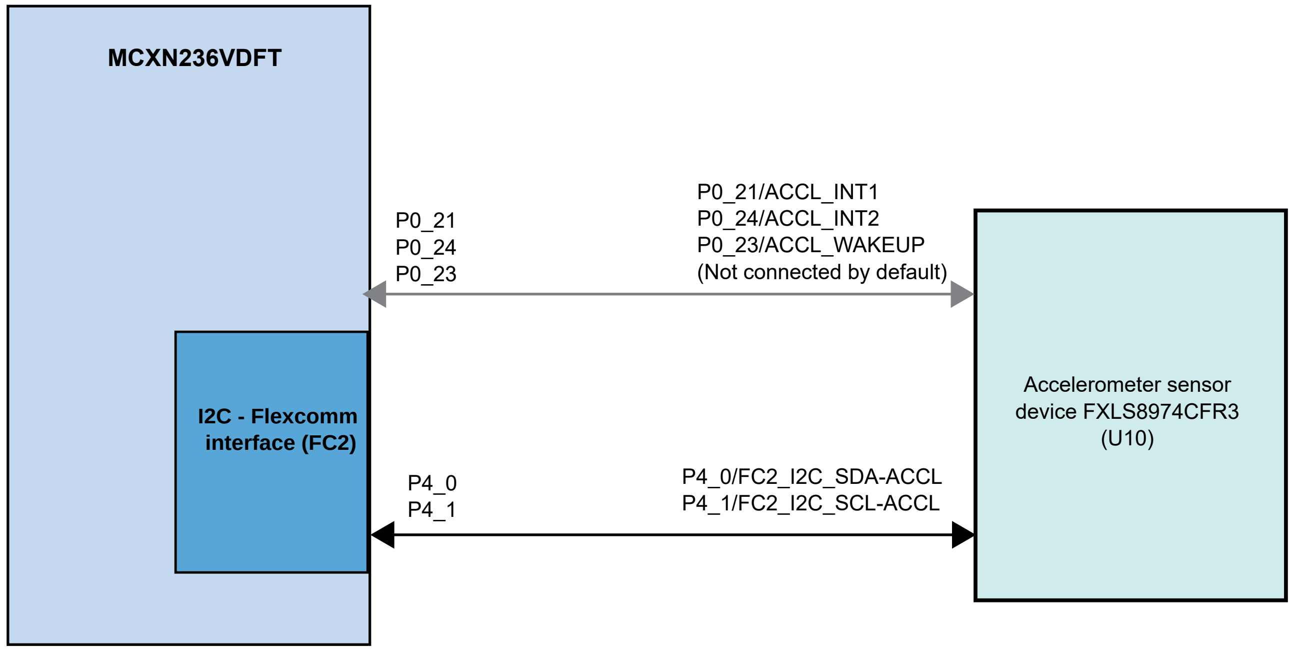


Figure 12. FXLS8974CFR3 sensor circuit

The sensor device connect to the I2C controller of the device through P4_[0:1] port. The wakeup pin is connected to the P0_23 port.

2.6 Flash memory interface

On the FRDM-MCXN236 board, one serial flash memory is provided. The flash memory VCC pin is supplied by the VDD_BOARD rail. The QSPI data and clock signals for the flash memory interface are available on Port P3_[12:13] and P3_[20:23] pins.

Table 1 provides the detail of the QSPI flash memory on the board.

Table 15. QSPI flash memory

Part identifier	Manufacturer and part name	Description
U12	Winbond W25Q64JVSSIQ	It is a 3 V 64-Mbit (8 MB) serial flash memory with dual and quad SPI, which is intended for demonstrating SPI boot applications, and general SPI operation.

Figure 13 shows the flash memory circuit diagram.

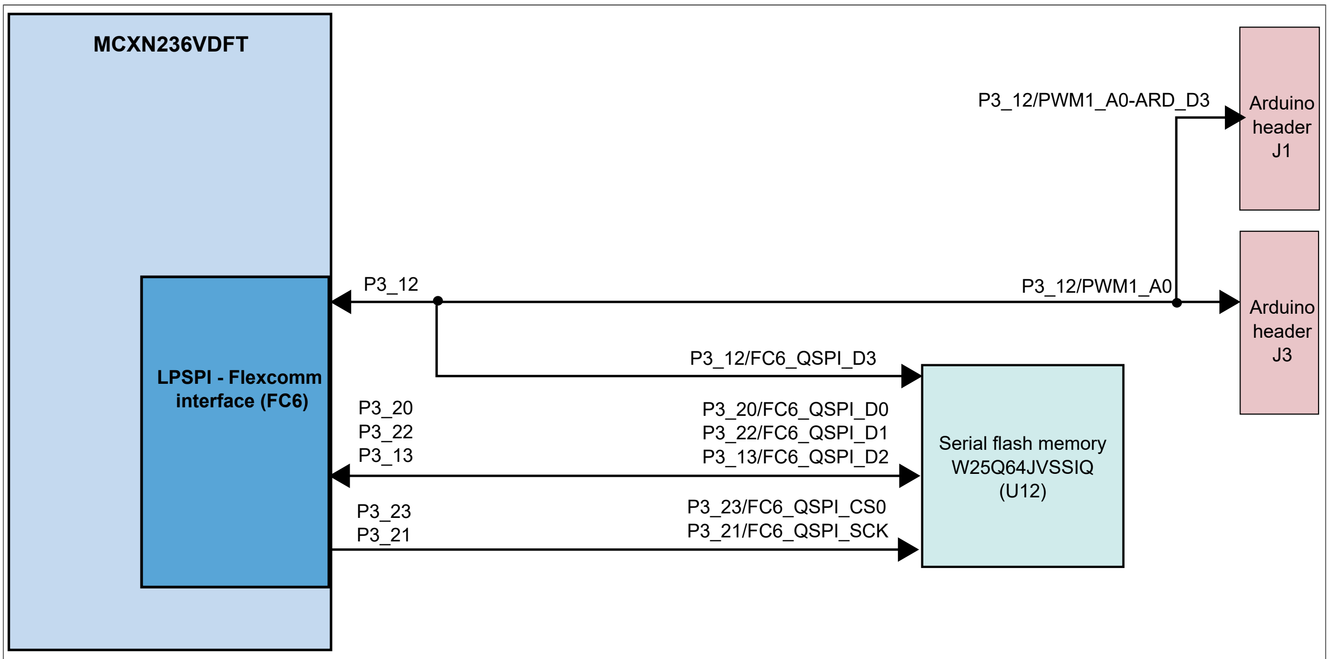


Figure 13. Flash memory circuit diagram

The LPSPI interface line P3_12 is shared with signals on the Arduino compatible headers J1 and J3. A zero-ohm resistor allows for selection of the signal between the flash memory U12 or Arduino headers.

Figure 14 shows isolation resistors that are provided to disconnect the flash memory circuit.

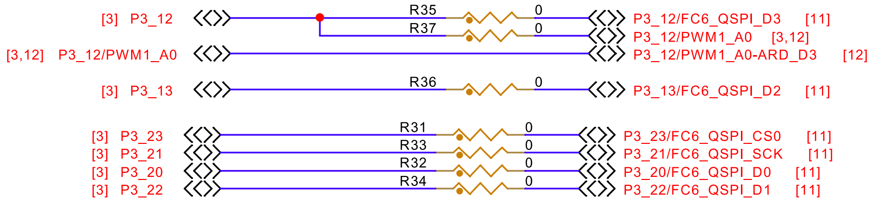


Figure 14. Isolation resistors for flash memory

2.7 Visible light sensor interface

On the FRDM-MCX236 board, one light sensor is provided, which connects to the P0_25 port of the target device (MCXN236) for evaluating the ADC module.

Table 1 provides the detail of the light sensor device on the board.

Table 16. Light sensor device

Part identifier	Manufacturer and part name	Description
Q2	Everlight ALS-PT19-315C/L177/TR8	It is a low-cost ambient light sensor, consisting of phototransistor in miniature SMD.

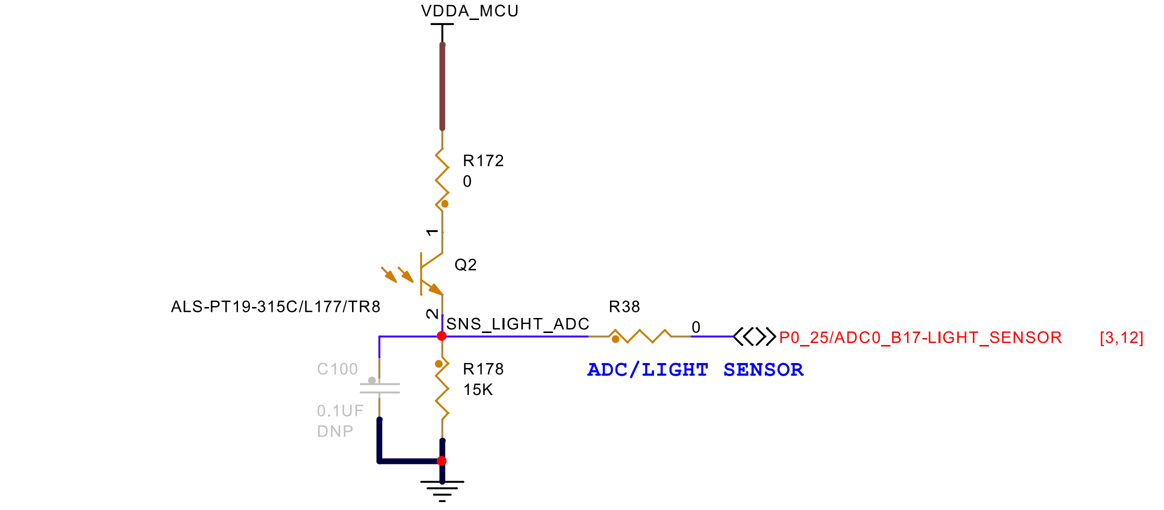


Figure 15. Light sensor circuit diagram

The zero-ohm resistor R38 is provided to isolate the light sensor from the MCXN236 device.

The input voltage to the light sensor is VDDA_MCU through the zero-ohm resistor (R172) to allow for removing the light sensor from any current measurements being made. If VDDA_MCU 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_MCU.

2.8 Audio interface

The MCXN236 device features two instances of the SAI module, SAI0 and SAI1.

The FRDM-MCXN236 board features an audio codec DA7212 (DNP), which connects to the SAI1 module of the target MCU. By default, the SAI1 receive and transmit signals from the MCXN236 MCU connect to the onboard audio codec through Port P3 and P2 pins. These signals also connect optionally to the odd-numbered pins of Arduino compatible header J1.

The 7-bit I2C slave address for the audio codec is 0x1A, therefore, the 8-bit address for writing is 0x34 and for reading is 0x35.

The VDD_1V8 supply is provided to the analog circuits of the audio codec. The VDD_AUDIO supply is provided to the digital interface and microphone bias circuits of the audio codec. The source for the VDD_AUDIO supply is VDD_BOARD through the R95 resistor.

One digital microphone (SPK0641HT4H-1 DMIC) is provided to supplement the audio circuit.

Figure 16 shows the audio codec and digital microphone circuit.

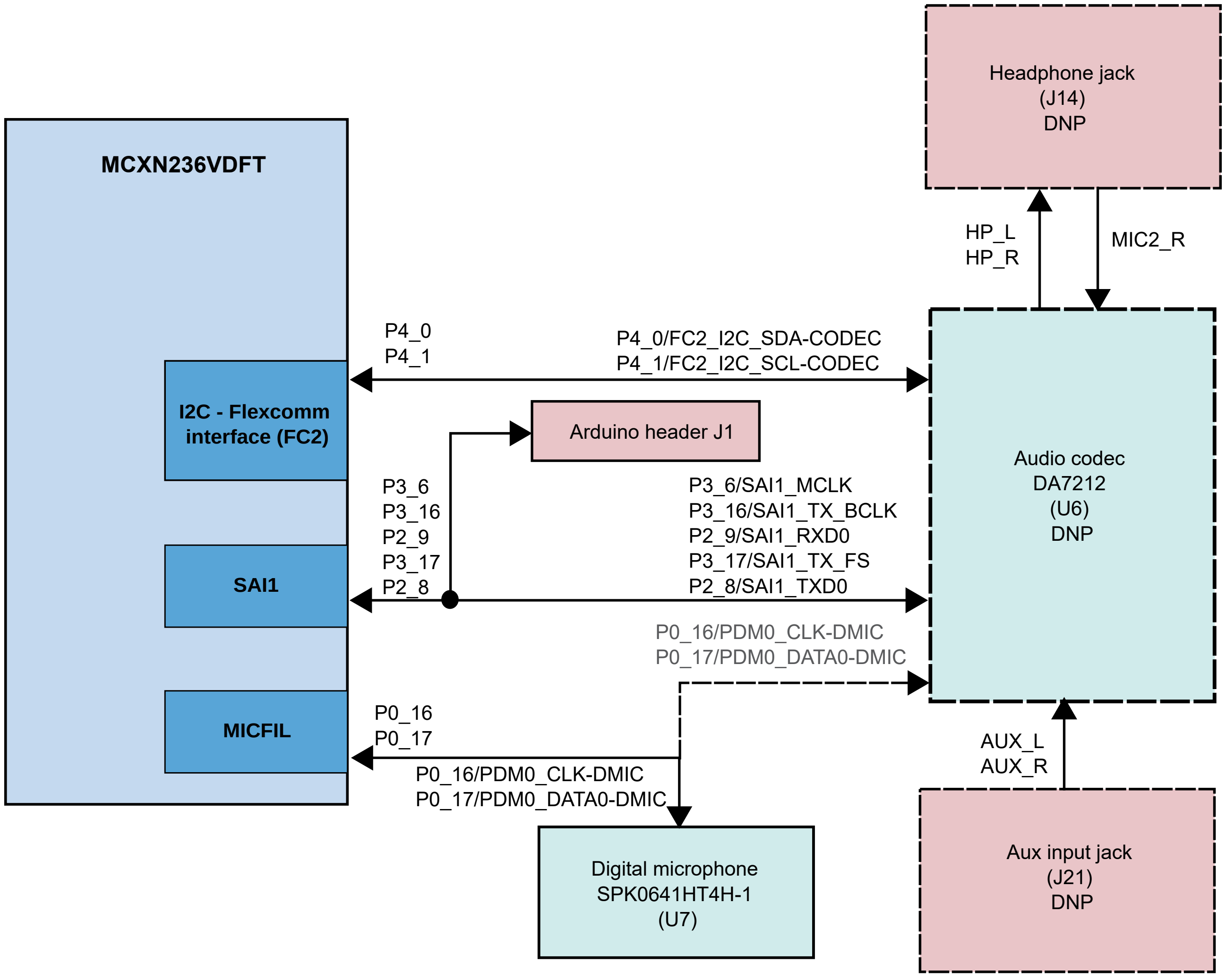


Figure 16. Audio circuit

Table 1 describes the details of the onboard audio codec and other audio device and connectors.

Table 17. Audio devices

Part identifier	Part name and manufacturer name	Description
U6 (DNP)	Renesas DA7212	Ultra low-power audio codec that allows for FM stereo line in inputs, analog / digital microphone inputs, and a true-ground class G headphone output
U7	Knowles SPK0641HT4H-1	High-performance, low-power digital microphone (DMIC) with a single-bit PDM output
J21 (DNP)	54-00174	Auxiliary input jack
J14 (DNP)	54-00174	Headphone jack

2.9 Arduino compatible I/O headers

The FRDM-MCXN236 board provides Arduino Uno compatible headers to support the Arduino and FRDM ecosystem shield modules. These headers are dual-row headers with the outer rows supporting the Arduino compatible shields and the inner rows supporting the various FRDM shields. These headers are designed to support the following shields:

- Sensor: FRDM-STBC-AGM01, FRDM-STBC-AGM04, FRDM-FXS-MULT2-B
- NFC: OM5577, OM5578
- USB Type C: OM13790 (Host)
- Motor control: FRDM-MC-LVBLDC, FRDM-MC-LVPMSM
- Audio: ARD-AUDIO-DA7212

Table 18 describes the connectors of the Arduino socket.

Table 18. Arduino socket connectors

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

Figure 17 shows the pinout of the Arduino socket connectors.

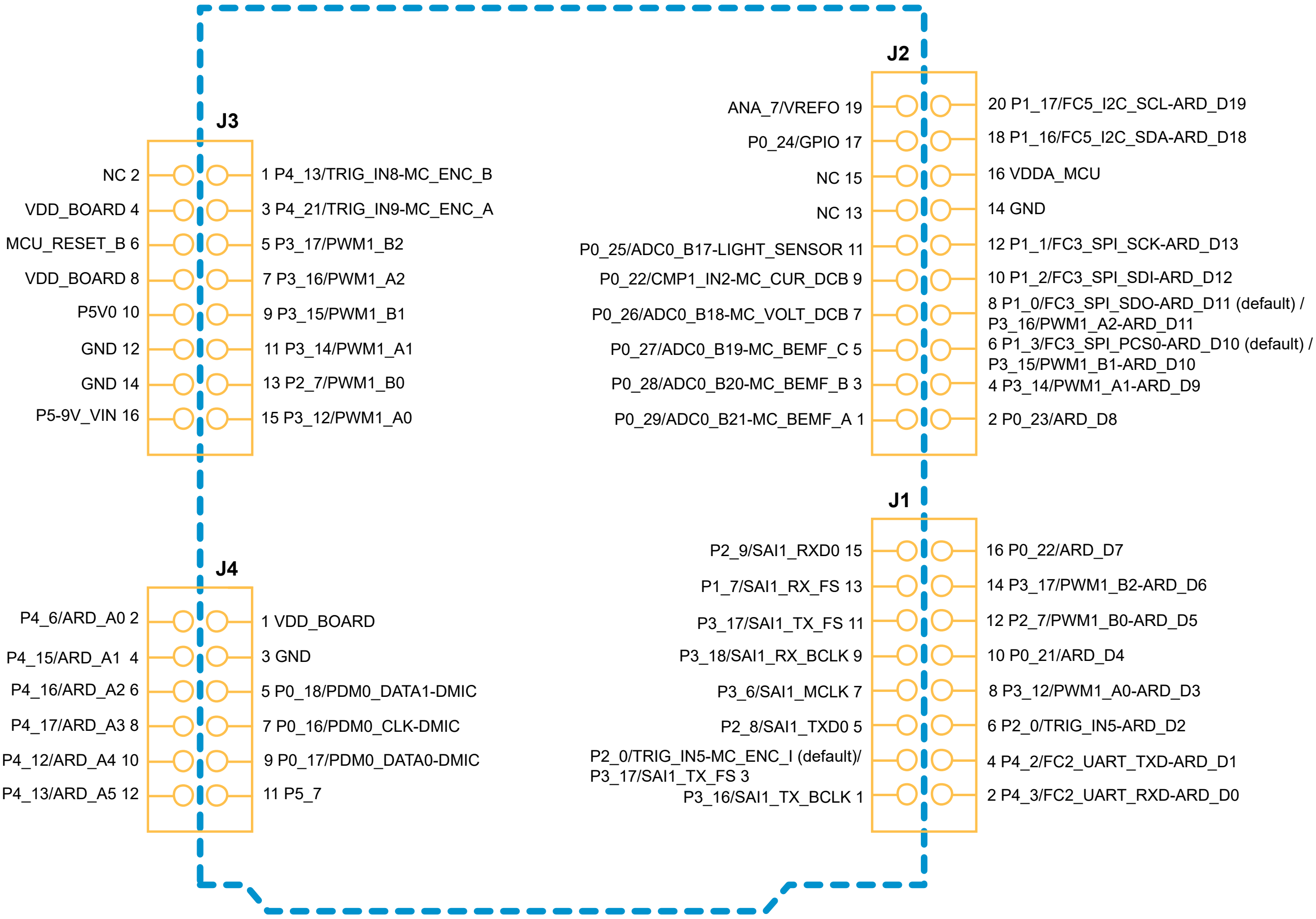


Figure 17. Arduino socket connector pinout

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 18 describes such signals.

Table 19. Arduino compatible header J1 pinout

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
1	P3_16	SAI1_TX_BCLK	-	• Arduino connector J2 pin 10 (P3_16/PWM1_A2-ARD_D11 through SJ2 resistor pin 2-3 selection (DNP by default)) • Arduino connector J3 pin 7 (P3_16/PWM1_A2)
2	P4_3	FC2_UART_RXD-ARD_D0	-	• mikroBUS header J5 pin 3 (P4_3/FC2_UART_RXD-MIKROE) • MCU-Link VCOM (P4_3/FC2_UART_RXD-MCULINK_VCOM_RX)
3	P3_17	SAI1_TX_FS	SJ3 Pin 2-3 selection	• Arduino connector J1 pin 14 (P3_17/PWM1_B2-ARD_D6) • Arduino connector J3 pin 5 (P3_17/PWM1_B2)
	P2_0	TRIG_IN5-MC_ENC_I	SJ3 Pin 1-2 selection (Default setting)	• Arduino connector J1 pin 6 (P2_0/TRIG_IN5-ARD_D2)
4	P4_2	FC2_UART_TXD-ARD_D1	-	• mikroBUS header J5 pin 4 (P4_2/FC2_UART_TXD-MIKROE) • MCU-Link VCOM (P4_2/FC2_UART_TXD-MCULINK_VCOM_TX)
5	P2_8	SAI1_TXD0	-	-
6	P2_0	TRIG_IN5-ARD_D2	-	• Arduino connector J1 pin 3 (P2_0/TRIG_IN5-MC_ENC_I)
7	P3_6	SAI1_MCLK	-	-
8	P3_12	PWM1_A0-ARD_D3		• QSPI Flash memory U12 (P3_12/FC6_QSPI_D3) • Arduino connector J3 pin 15 (P3_12/PWM1_A0)
9	P3_18	SAI1_RX_BCLK	-	-
10	P0_21	ARD_D4	-	• FlexIO header J8 pin 14 (P0_21/FXIO D5) • I2C sensor (U10) pin 8 (P0_21/ACCL_INT1)
11	P3_17	SAI1_TX_FS	-	• Arduino connector J1 pin 14 (P3_17/PWM1_B2-ARD_D6) • Audio codec (P3_17/SAI1_TX_FS)
12	P2_7	PWM1_B0-ARD_D5	-	• Arduino connector J3 pin 13 (P2_7/PWM1_B0)
13	P1_7	SAI1_RX_FS	-	• Camera header J9 pin 9 (P1_7/SmartDMA_PIO3-CAMERA_D3)
14	P3_17	PWM1_B2-ARD_D6	-	• Arduino connector J1 pin 11 (P3_17/SAI1_TX_FS)

Table 19. Arduino compatible header J1 pinout*...continued*

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
15	P2_9	SAI1_RXD0	-	-
16	P0_22	ARD_D7	-	• Arduino connector J2 pin 9 (P0_22/CMP1_IN2-MC_CUR_DCB)

Table 20. Arduino compatible header J2 pinout

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
1	P0_29	ADC0_B21-MC_BEMF_A	-	-
2	P0_23	ARD_D8	-	• FlexIO header J8 pin 5 (P0_23/FXIO_LCD_INT) • I2C sensor (U10) pin 7 (P0_23/ACCL_WAKEUP)
3	P0_28	ADC0_B20-MC_BEMF_B	-	-
4	P3_14	PWM1_A1-ARD_D9	-	• Arduino header J3 pin 11 (P3_14/PWM1_A1)
5	P0_27	ADC0_B19-MC_BEMF_C	-	-
6	P1_3	FC3_SPI_PCS0-ARD_D10	SJ1 Pin 1-2 selection (default setting)	• MCU-Link USB-SPI bridge (P1_3/FC3_SPI_PCS0-MCULINK_VIO)
	P3_15	PWM1_B1-ARD_D10	SJ1 Pin 2-3 selection	• Arduino header J3 pin 9 (P3_15/PWM1_B1)
7	P0_26	ADC0_B18-MC_VOLT_DCB	-	-
8	P1_0	FC3_SPI_SDO-ARD_D11	SJ2 Pin 1-2 selection (default setting)	• MCU-Link USB-SPI bridge (P1_0/FC3_SPI_SDO-MCULINK_VIO) • mikroBUS header J6 pin 6 (P1_0/FC3_SPI_SDO-MIKROE)
	P3_16	PWM1_A2-ARD_D11	SJ2 Pin 2-3 selection	• Arduino header J1 pin 1 (P3_16/SAI1_TX_BCLK) • Audio codec U6 (P3_16/SAI1_TX_BCLK) • Arduino header J3 pin 7 (P3_16/PWM1_A2)
9	P0_22	CMP1_IN2-MC_CUR_DCB	-	• Arduino header J1 pin 16 (P0_22/ARD_D7)
10	P1_2	FC3_SPI_SDI-ARD_D12	-	• MCU-Link USB bridge (P1_2/FC3_SPI_SDI-MCULINK_VIO) • mikroBUS header J6 pin 5 (P1_2/FC3_SPI_SDI-MIKROE)
11	P0_25	ADC0_B17-LIGHT_SENSOR	-	-
12	P1_1	FC3_SPI_SCK-ARD_D13	-	• mikroBUS header J6 pin 4 (P1_1/FC3_SPI_SCK-MIKROE)
13	-	-	-	-
14	-	GND	-	-

Table 20. Arduino compatible header J2 pinout*...continued*

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
15	-	-	-	-
16	-	VDDA_MCU	-	-
17	P0_24	GPIO	-	• I2C sensor (U10) pin 6 (P0_24/ ACCL_INT2) • FlexIO header J8 pin 8 (P0_24/ FXIO_LCD_DC)
18	P1_16	FC5_I2C_SDA-ARD_D18	-	• Camera header (U9) pin 4 (P1_ 16/SmartDMA_PIO12) • mikroBUS header J6 pin 3 (P1_ 16/FC3_SPI_PCS3-MIKROE)
19	-	ANA_7/VREFO	-	-
20	P1_17	FC5_I2C_SCL-ARD_D19	-	• Camera header (U9) pin 3 (P1_ 17/SmartDMA_PIO13)

Table 21. Arduino compatible header J3 pinout

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
1	P4_13	TRIG_IN8-MC_ENC_B	-	• Arduino header J4 pin 12 (P4_ 13/ARD_A5)
2	-	-	-	-
3	P4_21	TRIG_IN9-MC_ENC_A	-	-
4	-	VDD_BOARD	-	-
5	P3_17	PWM1_B2	-	• Arduino connector J1 pin 11 (P3_17/SAI1_TX_FS) • Arduino connector J1 pin 14 (P3_17/PWM1_B2-ARD_D6) • Audio codec (P3_17/SAI1_TX_ FS)
6	RESET_B	MCU_RESET_B	-	-
7	P3_16	PWM1_A2	-	• Arduino header J1 pin 1 (P3_ 16/SAI1_TX_BCLK) • Arduino connector J2 pin 10 (P3_16/PWM1_A2-ARD_D11 through SJ2 resistor pin 2-3 selection (DNP by default)) • Audio codec U6 (P3_16/SAI1_ TX_BCLK)
8	-	VDD_BOARD	-	-
9	P3_15	PWM1_B1	-	• Arduino connector J2 pin 6 (P3_15/PWM1_B1-ARD_D10 through SJ1 resistor pin 2-3 selection (DNP by default))
10	P5V0	-	-	-
11	P3_14	PWM1_A1	-	• Arduino connector J2 pin 4 (P3_14/PWM1_A1-ARD_D9)

Table 21. Arduino compatible header J3 pinout*...continued*

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
12	GND	-	-	-
13	P2_7	PWM1_B0	-	• Arduino connector J1 pin 12 (P2_7/PWM1_B0-ARD_D5)
14	GND	-	-	-
15	P3_12	PWM1_A0	-	• QSPI Flash memory U12 (P3_12/FC6_QSPI_D3) • Arduino header J1 pin 8 (P3_12/PWM1_A0-ARD_D3)
16	-	P5-9V_VIN	-	-

Table 22. Arduino compatible header J4 pinout

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
1	-	VDD_BOARD	-	-
2	P4_6	ARD_A0	-	• FlexIO header J8 pin 6 (P4_6/FXIO_LCD_GPIO)
3	-	GND	-	-
4	P4_15	ARD_A1	-	• CAN PHY U11 (P4_15/CAN1_RXD)
5	P0_18	PDM0_DATA1-DMIC	-	• Camera header (U9) pin 18 (P0_18/FXIO_D2-CAMERA_HSYNC) through the resistor SJ5 Pin 2-3 selection (DNP by default)
6	P4_16	ARD_A2	-	• CAN PHY U11 (P4_16/CAN1_TXD) • USB Type-C VBUS control (P4_16/USB1_OTG_PWR) through the zero-ohm resistor R92 (DNP)
7	P0_16	PDM0_CLK-DMIC	-	• MCU-Link USB-I2C bridge (P0_16/FC0_I2C_SDA-MCULINK_VIO)
8	P4_17	ARD_A3	-	• USB Type-C VBUS control (P4_17/USB1_OTG_OC) through the zero-ohm resistor R88 (DNP)
9	P0_17	PDM0_DATA0-DMIC	-	• MCU-Link USB-I2C bridge (P0_17/FC0_I2C_SCL-MCULINK_VIO)
10	P4_12	ARD_A4	-	-
11	P5_7	-	-	-

Table 22. Arduino compatible header J4 pinout*...continued*

Pin number	Device pin / GPIO	Function / Signal name	Resistor setting	Potential conflict
12	P4_13	ARD_A5	-	• Arduino header J1 pin 1 (P4_13/TRIG_IN8-MC_ENC_B)

2.10 FlexIO header

On the FRDM-MCXN236 board, one 28-pin FlexIO header (J8) is provided to support the LCD display and camera applications. Table 23 describes the pinout of the FlexIO header.

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

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

Table 23. FlexIO header J8 pinout

Pin number	GPIO	Function / Signal name	Description	Potential conflict
1	-	VDD_BOARD	-	-
2	-	GND	-	-
3	P4_1	FC2_I2C_SCL-FXIO_HDR	I2C_SCL	-
4	P4_0	FC2_I2C_SDA-FXIO_HDR	I2C_SDA	-
5	P0_23	FXIO_LCD_INT	GPIO	• I2C sensor (U10) pin 7 (P0_23/ACCL_WAKEUP) • Arduino connector J2 pin 2 (P0_23/ARD_D8)
6	P4_6	FXIO_LCD_GPIO	GPIO	• Arduino connector J4 pin 2 (P4_6/ARD_A0)
7	P4_7	FXIO_LCD_RST	GPIO	-
8	P0_24	FXIO_LCD_DC	GPIO	• I2C sensor (U10) pin 6 (P0_24/ACCL_INT2) • Arduino connector J2 pin 17 (P0_24/GPIO)
9	P4_14	FXIO_LCD_CS	GPIO	-
10	P4_20	FXIO_LCD_WR	FXIO D28	-
11	P4_19	FXIO_LCD_RD	FXIO D27	• RGB LED (P4_19/LED_GREEN)
12	P4_23	FXIO_LCD_TE	GPIO	-
13	P0_20	FXIO_D4	FXIO D4	• Wakeup switch (SW2) through zero-ohm resistor (R12)
14	P0_21	FXIO D5	FXIO D5	• Arduino connector J1 pin 10 (P0_21/ARD_D4) • I2C sensor (U10) pin 8 (P0_21/ACCL_INT1)

Table 23. FlexIO header J8 pinout*...continued*

Pin number	GPIO	Function / Signal name	Description	Potential conflict
15	P0_14	FXIO_D6	FXIO D6	-
16	P0_15	FXIO_D7	FXIO D7	-
17	P3_0	FXIO_D8	FXIO_D8	-
18	P3_1	FXIO_D9	FXIO_D9	-
19	P3_2	FXIO_D10	FXIO_D10	-
20	P2_3	FXIO_D11	FXIO_D11	-
21	P2_4	FXIO_D12	FXIO_D12	-
22	P2_5	FXIO_D13	FXIO_D13	-
23	P2_6	FXIO_D14	FXIO_D14	-
24	P3_7	FXIO_D15	FXIO_D15	-
25	P3_8	FXIO_D16	FXIO_D16	-
26	P3_9	FXIO_D17	FXIO_D17	-
27	P3_10	FXIO_D18	FXIO_D18	-
28	P3_11	FXIO_D19	FXIO_D19	-

2.11 mikroBUS headers

Figure 18 shows the mikroBUS schematic diagram.

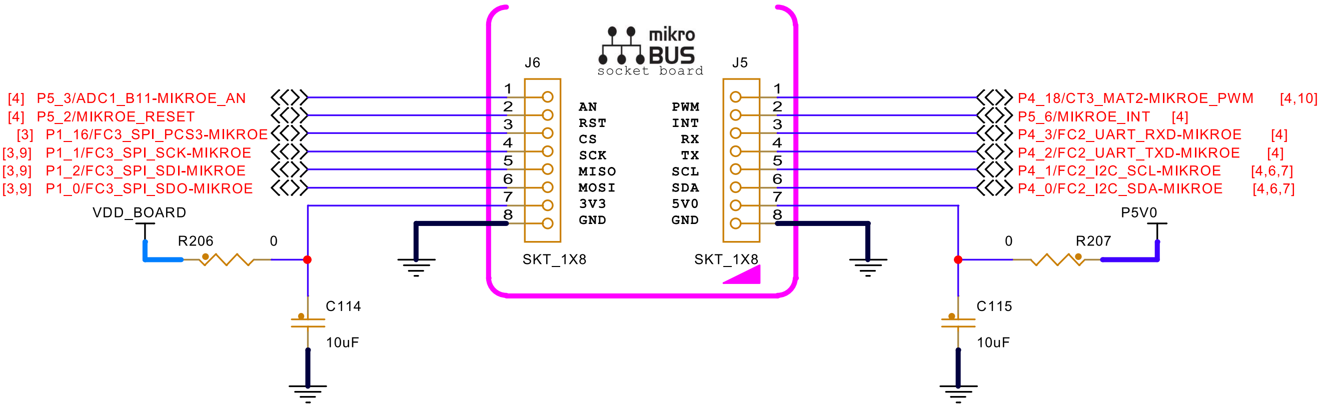

[4] P5_3/ADC1_B11-MIKROE_AN
[4] P5_2/MIKROE_RESET
[3] P1_16/FC3_SPI_PCS3-MIKROE
[3,9] P1_1/FC3_SPI_SCK-MIKROE
[3,9] P1_2/FC3_SPI_SDI-MIKROE
[3,9] P1_0/FC3_SPI_SDO-MIKROE
VDD_BOARD
R206
0
C114
10uF
J6
1
2
3
4
5
6
7
8
SKT_1X8
AN
RST
CS
SCK
MISO
MOSI
3V3
GND
mikro BUS socket board
PWM
INT
RX
TX
SCL
SDA
5V0
GND
J5
SKT_1X8
P4_18/CT3_MAT2-MIKROE_PWM [4,10]
P5_6/MIKROE_INT [4]
P4_3/FC2_UART_RXD-MIKROE [4]
P4_2/FC2_UART_TXD-MIKROE [4]
P4_1/FC2_I2C_SCL-MIKROE [4,6,7]
P4_0/FC2_I2C_SDA-MIKROE [4,6,7]
P5V0
0
R207
C115
10uF

Figure 18. mikroBUS header pinout

Table 25 and Table 24 describe the pinout of the mikroBUS headers (J6 and J5).

Table 24. J6 header pinout

Pin number	GPIO	Function / Signal name	Potential conflict
1	P5_3	ADC1_B11-MIKROE_AN	-
2	P5_2	MIKROE_RESET	
3	P1_16	FC3_SPI_PCS3-MIKROE	• Camera header (J9) pin 4 (P1_16/Smart DMA_PIO12)

Table 24. J6 header pinout*...continued*

Pin number	GPIO	Function / Signal name	Potential conflict
			• Arduino shield compatible header J2 pin 18 (P1_16/FC5_I2C_SDA-ARD_D18)
4	P1_1	ME_FC6_SPI_CLK	• MCU-LINK USB bridge (P1_1/FC3_SPI_SCK-MCULINK_VIO) • Arduino shield compatible header J2 pin 12 (P1_1/FC3_SPI_SCK-ARD_D13)
5	P1_2	FC3_SPI_SCK-MIKROE	• MCU-LINK USB bridge (P1_2/FC3_SPI_SDI-MCULINK_VIO) • Arduino shield compatible header J2 pin 10 (P1_2/FC3_SPI_SDI-ARD_D12)
6	P1_0	FC3_SPI_SDO-MIKROE	• MCU-LINK USB bridge (P1_0/FC3_SPI_SDI-MCULINK_VIO) • Arduino connector (J2) pin 8 (P1_0/FC3_SPI_SDO-ARD_D11)
7	-	VDD_BOARD	VDD_BOARD
8	-	GND	Ground

Table 25. J5 header pinout

Pin number	GPIO	Function / Signal name	Potential conflict
1	P4_18	CT3_MAT2-MIKROE_PWM	-
2	P5_6	MIKROE_INT	-
3	P4_3	FC2_UART_RXD-MIKROE	• Arduino connector J1 pin 2 (P4_3/FC2_UART_RXD-ARD_D) • MCU-Link VCOM (P4_3/FC2_UART_RXD-MCULINK_VCOM_RX through R56 resistor (DNP by default))
4	P4_2	FC2_UART_TXD-MIKROE	• Arduino connector J1 pin 2 (P4_3/FC2_UART_RXD-ARD_D) • MCU-Link VCOM (P4_2/FC2_UART_TXD-MCULINK_VCOM_TX through R55 resistor (DNP by default))
5	P4_1	FC2_I2C_SCL-MIKROE	-
6	P4_0	FC2_I2C_SDA-MIKROE	-
7	-	P5V0	5 V power line
8	-	GND	Ground

2.12 Camera header

The FRDM-MCXN236 provides a header for the camera connection. This is to demonstrate the camera interface features of the MCXN236 device.

Note: *The FRDM-MCXN236 board is tested with the OV7670 camera. Only pin 7- pin 22 are necessary for the OV7670 camera.*

Table 26 describes the camera header (J9) pinout.

Table 26. Camera header connections

Pin number	GPIO	Function	Resistor setting	Potential conflict
1	P1_15	SmartDMA_PIO11	-	-
2	P1_14	SmartDMA_PIO10	-	-
3	P1_17	SmartDMA_PIO13	-	• Arduino header J2 pin 20 (P1_17/FC5_I2C_SCL-ARD_D19)
4	P1_16	SmartDMA_PIO12	-	• mikroBUS header J6 pin 3 (P1_16/FC3_SPI_PCS3-MIKROE) • Arduino header J2 pin 18 (P1_16/FC5_I2C_SDA-ARD_D18)
5	P1_19	SmartDMA_PIO15-CAMERA_RST	-	-
6	P1_18	SmartDMA_PIO14-CAMERA_PDOWN	-	-
7	P1_5	SmartDMA_PIO1-CAMERA_D1	-	-
8	P1_4	SmartDMA_PIO0-CAMERA_D0	-	-
9	P1_7	SmartDMA_PIO3-CAMERA_D3	-	• Arduino header J2 pin 13 (P1_7/SAI1_RX_FS)
10	P1_6	SmartDMA_PIO2-CAMERA_D2	-	-
11	P1_9	SmartDMA_PIO5-CAMERA_D5	-	• MCU-Link VCOM (P1_9/FC4_TXD-MCULINK_VCOM_TX)
12	P1_8	SmartDMA_PIO4-CAMERA_D4	-	• MCU-Link VCOM (P1_8/FC4_RXD-MCULINK_VCOM_RX)
13	P1_11	SmartDMA_PIO7-CAMERA_D7	-	-
14	P1_10	SmartDMA_PIO6-CAMERA_D6	-	-
15	P0_7	SmartDMA_CAMERA_PCLK	SJ4 pin 1-2 selection (default setting)	-
	P4_22	FXIO_D30-CAMERA_PCLK	SJ4 pin 2-3 selection	-
16	P2_2	CLKOUT-SmartDMA_CAMERA_CLKIN	-	-
17	P0_4	SmartDMA_CAMERA_VSYNC	-	-
18	P0_5	SmartDMA_CAMERA_HSYNC	SJ5 pin 1-2 selection (default setting)	-
	P0_18	FXIO_D2-CAMERA_HSYNC	SJ5 pin 2-3 selection	• Arduino header J4 pin 5 (P0_18/PDM0_DATA1-DMIC)
19	P4_1	FC2_I2C_SCL-CAMERA	-	-
20	P4_0	FC2_I2C_SDA-CAMERA	-	-
21	-	VDD_BOARD	-	-
22	-	GND	-	-
23	P4_5	SmartDMA_PIO29-LCD_RD	-	-
24	P2_10	SmartDMA_PIO31-LCD_DC	-	-
25	P4_4	SmartDMA_PIO28-LCD_WR	-	-

Table 26. Camera header connections...*continued*

Pin number	GPIO	Function	Resistor setting	Potential conflict
26	P2_11	SmartDMA_PIO30-LCD_CS	-	-
27	P1_13	SmartDMA_PIO9	-	-
28	P1_12	SmartDMA_PIO8	-	-
29	-	-	-	-
30	P2_1	GPIO	-	-
31	-	-	-	-
32	P0_19	GPIO	-	-

2.13 Board operating conditions

The operating temperature range for the FRDM-MCXN236 board is -40 ℃ to +105 ℃. The MCXN23x device supports up to 125 ℃ junction temperature. See *MCX N23x 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. The architecture is used both in standalone debug probes (such as MCU-Link Pro) and onboard debug probes in evaluation boards, such as FRDM-MCXN236. The onboard implementations of MCU-Link are referred to as MCU-Link OB.

The FRDM-MCXN236 board implements a subset of the MCU-Link architecture features, as described in Section 3.2. For more information on MCU-Link, visit MCU-Link Debug Probe Architecture.

The MCU-Link OB on the FRDM-MCXN236 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 27 summarizes the MCU-Link features supported on the FRDM-MCXN236 board.

Table 27. 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 27. 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 (MCXN236) using an external debug probe, instead of MCU-Link. Support for an external debug probe is enabled by disabling the SWD feature. While using an external debug probe, the VCOM and USBSIO features can be used.
External target support[1]	Allows debugging an external target MCU using MCU-Link

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

3.3 Supported debug scenarios

In the FRDM-MCXN236 board, the MCU-Link debug probe target can be either the MCXN236 MCU or an external target compliant with MCU-Link. The board also allows to use an external debugger for debugging the MCXN236 MCU, in place of the MCU-Link debug probe.

Table 28 describes the debug scenarios supported on the FRDM-MCXN236 board.

Table 28. Supported debug scenarios

Debug scenario	Feature support	Resistor / Jumper / connector settings
Use MCU-Link as a debugger for the target MCU (MCXN236)	SWD is enabled	JP7 must be open
	VCOM is enabled	JP6 must be open
	USBSIO is enabled	R134 (1 kΩ) DNP
Use an external debugger to debug the target MCU (MCXN236)	SWD is disabled	• JP7 must be shorted • Connect an external debugger to the target MCU SWD connector J12
	VCOM is enabled	JP6 must be open
	USBSIO is enabled	R134 (1 kΩ) DNP
Use MCU-Link as a debugger for an external target MCU	SWD is enabled	• JP7 must be open • Connect an external target MCU to the target MCU SWD connector J12
	VCOM is not supported	JP6 must be shorted
	USBSIO is not supported	Populate R134 (1 kΩ)

3.4 MCU-Link host driver and utility installation

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

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

Note: *If the firmware version of the onboard MCU-Link probe is 3.122 or later, LinkServer version 1.4.85 or later provides the support of automatic firmware update. For further details on automatic firmware update, refer to the readme markdown file in the LinkServer package. However, If the current firmware version is earlier than 3.122, you can update the firmware (see Updating MCU-Link firmware using the firmware utility) for the MCU-Link probe using the MCU-Link firmware update utility, which is included in the LinkServer installation package.*

Note: *In case you are using MCUXpresso IDE version 11.6.1 or earlier, you must install the firmware update utility version 2.263, which is not included in the LinkServer installation.*

You are recommended to update the MCU-Link firmware on the board to the latest firmware version to get the latest functionality. However, before updating the firmware, check compatibility with the MCUXpresso IDE and LIBUSBIO versions in Table 29 (if you are using these tools). *If you are using the MCUXpresso for Visual Studio Code extension or third-party IDEs from IAR or Keil, the latest firmware version is recommended..*

Table 29. 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 the firmware utility

To update the firmware using the firmware update utility, which is 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 JP5, and reconnect the board. The D7 (Red) MCU-Link status LED lights up and stays on. For the D7 LED details, see Section 3.10.
2. Download the LinkServer utility from https://nxp.com/linkserver and install the LinkServer installer. 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, MCU-LINK_installer_3.119.
4. Follow the instructions in the `readme.txt` to find and run the firmware update utilities for CMSIS-DAP or J-Link versions.
5. Disconnect the board from the host computer, open jumper JP5, 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 29 shows the compatibility between the MCU-Link firmware and the MCUXpresso IDE.

3.6.2 Using MCU-Link with MCUXpresso for Visual Studio Code

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

3.6.3 Using MCU-Link with third-party IDEs

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

3.7 MCU-Link USB connector

The FRDM-MCXN236 board has a USB 2.0 Type-C connector J10. 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 J10 is plugged into a USB host.

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

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

On the FRDM-MCXN236 board, MCU-Link LPC55S69 is connected to the P1_8 and P1_9 pins of the target MCU through the R54 and R53 resistors, respectively.

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

To use MCU-Link as a USB-to-UART bridge, ensure that the JP6 jumper is open and connect the J10 connector on the board to the USB port of the host computer.

When you boot the FRDM-MCXN236 board, a VCOM port with the name MCU-Link Vcom Port (COMxx) is enumerated on the host computer, where “xx” may vary from one computer to another. Each MCU-Link based board has a unique VCOM number associated with it.

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

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

In the FRDM-MCXN236 board, MCU-Link connects to the P1_[3:0] pins of the target MCU using the FC3 SPI interface connection through zero-ohm resistors. By default, these resistors are populated and enable the communication between MCU-Link and the target MCU through the USB-to-SPI bridge.

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

A USB-to-SPI bridge can be used to emulate the host system. To use MCU-Link as a USB-to-SPI bridge, the board must be connected to the host computer through a USB cable from its J10 connector. Also, ensure the following resistor configuration on the board to enable the USBSIO bridge feature for SPI:

- Resistors R153, R154, R155, and R156 are populated (default setting)
- Resistor R134 is DNP (default setting)

On the FRDM-MCXN236 board, MCU-Link is also connected to the P0_[17:16] pins of the target MCU using the FC2 I2C interface connection through zero-ohm resistors. By default, these resistors are populated and enable the communication between MCU-Link and the target MCU through the USB-to-I2C bridge.

A USB-to-I2C bridge can be used to emulate the host system / board peripherals. To use MCU-Link as a USB-to-I2C bridge, the board must be connected to the host computer through a USB cable from its J10 connector. Also, ensure the following resistor configuration on the board to enable the USBSIO bridge feature for I2C:

- Zero-ohm resistors R159 and R160 are populated
- Resistor R134 is DNP (default setting)
- 2.2 kΩ resistors R157 and R158 should be populated

3.10 MCU-Link status LEDs

The FRDM-MCXN236 board has three status indicator LEDs for MCU-Link. Table 30 lists these LEDs and describes how each LED behaves in different MCU-Link modes. These LEDs are shown in Figure 3.

Table 30. 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
D6	USB_ACTIVE / Green	Lights up after successful USB enumeration at startup. Afterward, the LED stays ON.	Remains OFF	Remains OFF
D7	ISP_EN / 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
D8	VCOM_ ACTIVE / green	Indicates if VCOM port is transmitting/receiving data	Lights up when MCU-Link boots, and blinks when debug activity happens	Remains OFF

4 Board errata

Not applicable for the current board revision.

5 Related documentation

Table 31 lists and explains the additional documents and resources that you can refer to for more information on the FRDM-MCXN236 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 31. Related documentation

Document	Description	Link / how to access
MCX N23x Product Family Data Sheet	It provides information about electrical characteristics, hardware design considerations, and ordering information	MCXN23x
MCX N23x Reference Manual	It is intended for the board-level product designers and product software developers who want to develop products with MCX Nx3x MCU	MCXN23XRM
MCX N23x Chip Errata (MCXN23x_0P21K)	Lists the details of all known silicon errata for the MCX N23x device.	Contact NXP FAE or sales representative
FRDM-MCXN236 design file	A zip file including *.DSN, ASY, Layout, schematic files, and so on	Contact NXP FAE or sales representative
LPC55S6x/LPC55S2x/LPC552x User manual (UM11126)	Intended for system software and hardware developers and application programmers who want to develop products with LPC55S6x/ LPC55S2x/LPC552x MCU	UM11126.pdf

6 Acronyms

Table 32 lists and defines the acronyms used in this document.

Table 32. Acronyms

Term	Description
ADC	Analog-to-digital converter
CAN	Controller area network
DNP	Do not populate
ESR	Equivalent series resistor
GPIO	General-purpose input/output
I2C	Inter-integrated circuit
ISP	In-system programming
LPI2C	Low-power inter-integrated circuit
PCB	Printed-circuit board
PHY	Physical interface of the OSI model
POR	Power-on reset

Table 32. Acronyms*...continued*

Term	Description
PWM	Pulse width modulation
QSPI	Quadruple serial peripheral interface
RTC	Real-time clock
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

7 Revision history

Table 33 summarizes the revisions to this document.

Table 33. Revision history

Document ID	Release date	Description
UM12041 v.2	15 May 2024	Initial public release

Legal information

Definitions

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

Disclaimers

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Trademarks

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

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

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'.

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

Date of release: 15 May 2024
Document identifier: UM12041