UM12120

FRDM-MCXC444 Board User Manual

Rev. 1 — 15 July 2024 User manual

Document information

Information	Content
Keywords	UM12120, FRDM-MCXC444, MCX C, MCXC444, Arduino, mikroBUS, MCU-Link
Abstract	The FRDM-MCXC444 board is a design and evaluation platform based on the NXP MCXC444 MCU.

1 Board overview

The FRDM-MCXC444 board is a standalone development platform that supports two microcontrollers (MCUs): the target MCU and an onboard debugger MCU. The target MCU is MCXC444VLH, which is a part of the MCX C series of Arm Cortex-M0+ MCUs product family. The onboard debugger MCU is an LPC55S6x MCU family device, LPC55S69JEV98.

The FRDM-MCXC444 board is compatible with the Arduino shield module, Mikroe click board, and Pmod board. The FRDM-MCXC444 board comes preloaded with a LED blinky demo. The demo is available at the boards \frdmmcxc444\demo_apps\led_blinky folder of MCUXpresso SDK.

The board is lead-free and RoHS-compliant. It can be used with a wide range of development tools, including NXP MCUXpresso IDE, MCUXpresso IDE for Visual Studio Code, IAR Embedded Workbench, and Arm Keil MDK.

This document provides detailed information about the FRDM-MCXC444 board including power supplies, clocks, LEDs, motion and light sensors, SLCD module, MCU-Link debug probe circuit, and other interfaces on the board.

For information on how to set up and boot the FRDM-MCXC444 board, see *FRDM-MCXC444 Quick Start Guide* provided in the FRDM-MCXC444 hardware kit.

1.1 Block diagram

Figure 1 shows the FRDM-MCXC444 board block diagram.

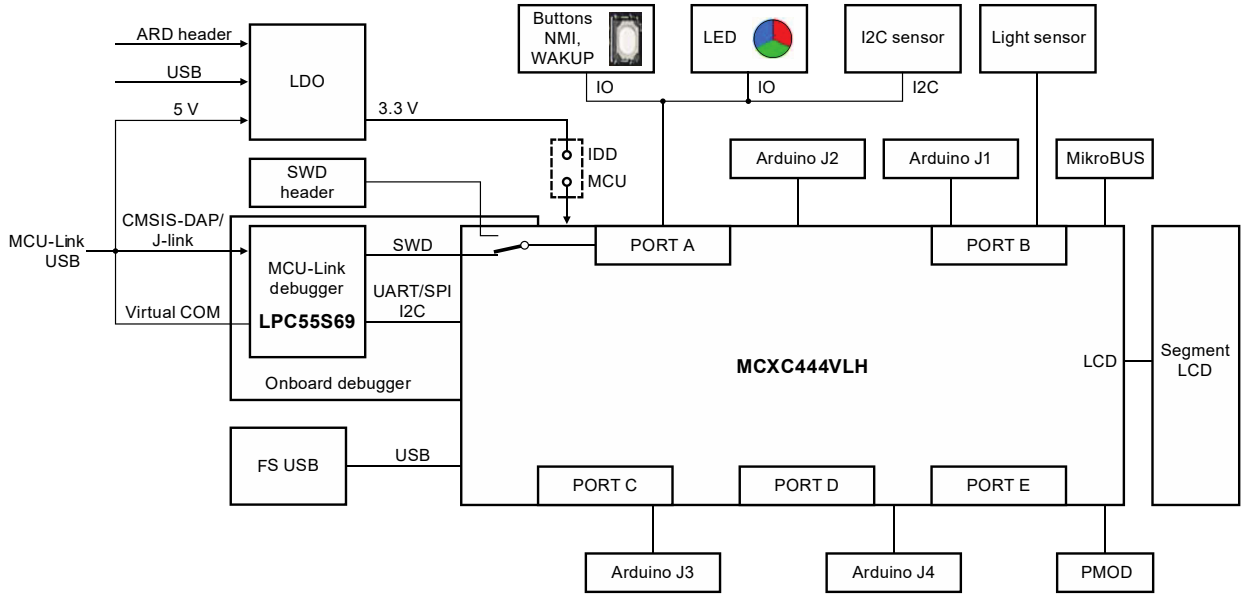

Figure 1. FRDM-MCXC444 block diagram

1.2 Board features

Table 1 lists the features of the FRDM-MCXC444 board.

Table 1. FRDM-MCXC444 features

Board feature	Target MCU features used	Description
MCU (target MCU)		NXP MCXC444VLH MCU based on an Arm Cortex-M0+ core running at speeds of up to 48 MHz ***Note:*** *For details on the MCXC444 MCU, see MCX C44X Sub-Family Reference Manual and MCX C44X Microcontroller Data Sheet.*
Connectivity	FS USB	USB2.0 Type-C connector (J12)
	SPI / LPUART / I2C	• mikroBUS socket (J5 and J6 headers) • Pmod header (J7) *Note: DNP by default.*
Clock		• 32.768 kHz crystal clock for the MCXC444VLH MCU • 16 MHz crystal clock for the LPC55S69JEV98 MCU-Link
7-segment LCD	SLCD controller	12-pin, 4 digit, 7 Segment LCD panel (LCD-S401M16KR)
Sensor	I2C0 module	Supports NXP FXLS8974CFR3 device supporting motion sensing
	ADC	Visible light sensor
Debug		• Onboard MCU-Link debug probe with CMSIS-DAP and SEGGER J-Link protocol options. It can act as a USB-to-UART, USB-to-SPI, or USB-to-I2C bridge between the target MCU and Host computer • 10-pin Arm JTAG/SWD connector for connecting an external debug probe
I/O expansion headers		Headers compatible with: • Arduino headers (outer rows) and FRDM header (inner rows) • Mikroe click boards • Peripheral module (Pmod) *Note: DNP by default.*
Buttons		• SW1 is used to reset the target MCU • SW2 is used to issue a wake-up input to the low-leakage wakeup unit (LLWU) module of the target MCU • SW3 is used to issue a non-maskable interrupt (NMI) signal to the MCXC444VLH MCU
RGB LED	Timer / PWM module (TPM) - TPM0	Supports RGB LED controlled by the embedded software application
Power supply		• P5V0 (5 V) input power supply using one of the following power sources: – Full-speed USB2.0 Type-C connector J12 – MCU-Link USB2.0 Type-C connector J13 • One LDO for 3.3 V power supply • Jumpers and resistors configuration for different power supplies
PCB		9 cm x 6 cm
Orderable part number		FRDM-MCXC444

1.3 Kit contents

Table 2 lists the items included in the FRDM-MCXC444 board hardware kit.

Table 2. Kit contents

Item	Quantity
FRDM-MCXC444 board hardware assembly	1
USB 2.0 Type-A to Type-C cable	1

1.4 Board pictures

Figure 2 shows the top-side view of the FRDM-MCXC444 board with the MCXC444VLH MCU (target MCU) and LPC55S69 MCU (MCU-Link) highlighted.

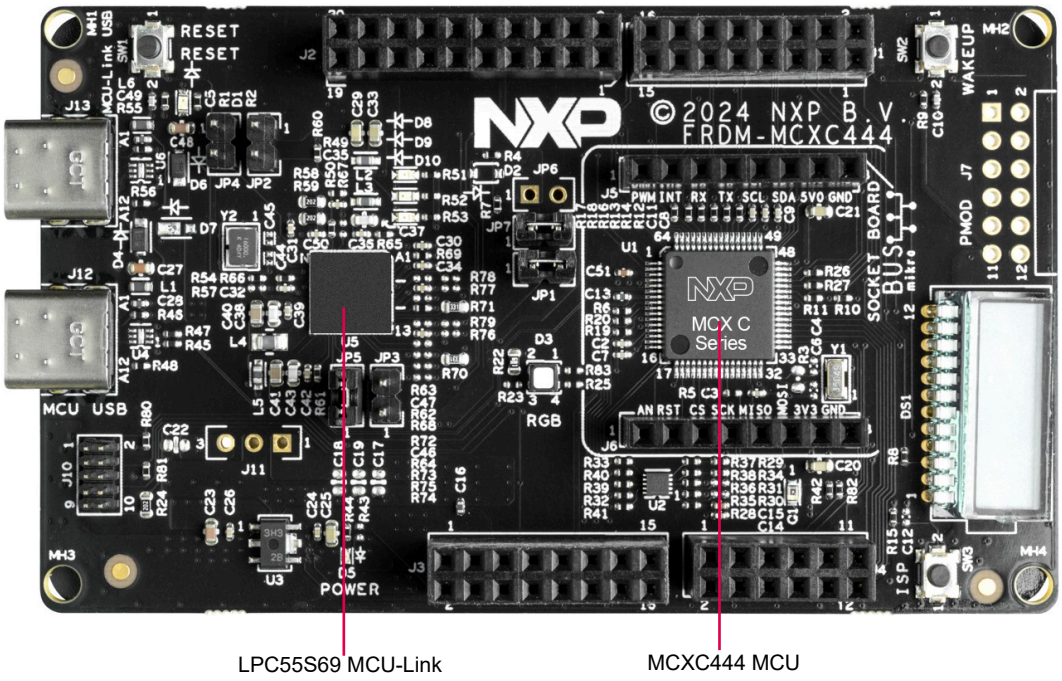

Figure 2. Board top-side view

Figure 3 shows the top-side view of the FRDM-MCXC444 board, with onboard connectors, jumpers, push buttons, and LEDs highlighted.

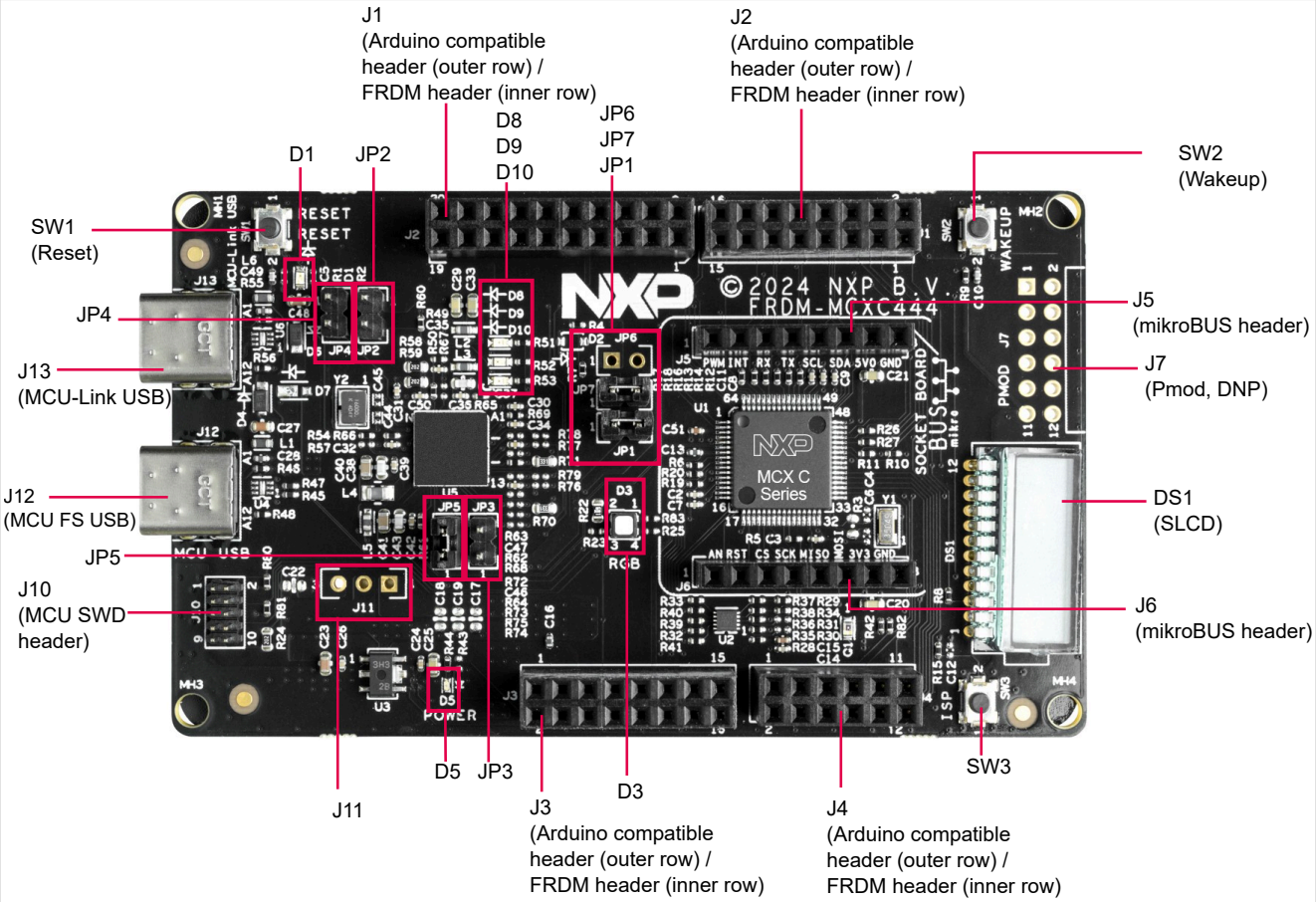

Figure 3. FRDM-MCXC444 connectors, jumpers, push buttons, and LEDs

Figure 4 shows the bottom-side view of the FRDM-MCXC444 board.

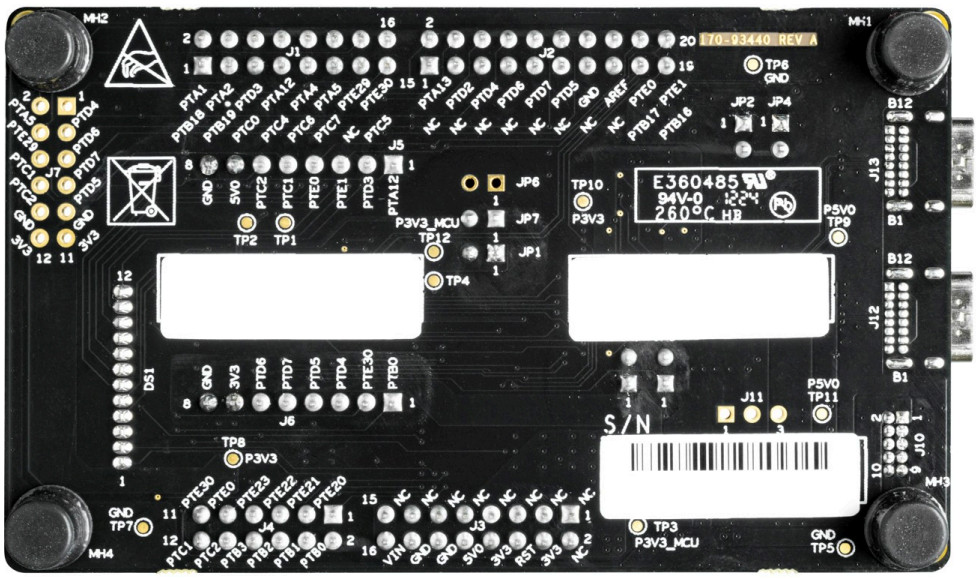

Figure 4. Board bottom-side view

1.5 Connectors

Figure 3 shows the FRDM-MCXC444 board connectors.

Table 3 describes the connectors available on the FRDM-MCXC444 board.

Table 3. FRDM-MCXC444 connectors

Part identifier	PCB label	Connector type	Description	Reference section
J1		2x8-pin header	Arduino socket connectors	Section 2.10
J2		2x10-pin header		
J3		2x8-pin header		
J4		2x6-pin header		
J5		1x8-pin header	mikroBUS socket connectors	Section 2.11
J6		1x8-pin header		
J7 (DNP)	PMOD	2x6-pin header	Pmod connector	Section 2.12
J0	SWD	2x5-pin header	Target MCU (MCXC444) external debugger connector	Section 3.2
J11 (DNP)	5VDC VR SUPPORT	1x3-pin connector	5 V DC voltage regulator	Section 2.1
J12	MCU USB	USB Type-C connector	MCXC444 full-speed USB connector	Section 2.3
J13	MCU-Link USB	USB Type-C connector	MCU-Link USB connector	Section 3.6

1.6 Jumpers

Figure 3 shows the FRDM-MCXC444 board jumpers.

Table 4 describes the FRDM-MCXC444 board jumpers.

Table 4. FRDM-MCXC444 jumpers

Part identifier	Name	Jumper type	Description	Reference section
JP1	POW_BRD	1x2-pin header	Shorted (default setting): P3V3_MCU is sourced from the P3V3 power supply. ***Note:*** *This jumper is also used for MCU current measurement. Connect a current meter between the jumper to measure the current and power consumption of the MCU. For details, see Section 2.1.1.*	Section 2.1
JP2	ISP_EN_SWD_ ACT	1x2-pin header	MCU-Link mode control jumper: • Open (default setting): MCU-Link (LPC55 S69) follows the normal boot sequence (MCU-Link boots from its internal flash if a boot image is found). With the internal flash erased, the MCU-Link normal boot sequence falls through to In-System Programming (ISP) boot mode. • Shorted: MCU-Link is forced to ISP mode (USB1). Use this setting to reprogram the MCU-Link internal flash with a new image or	Section 3.4

Table 4. FRDM-MCXC444 jumpers*...continued*

Part identifier	Name	Jumper type	Description	Reference section
			use the MCUXpresso IDE with the CMSIS-DAP protocol. ***Note:*** *By default, MCU-Link internal flash is preprogrammed with a version of CMSIS-DAP firmware.*	
JP3	VCOM_DIS	1x2-pin header	MCU-Link VCOM port disable jumper: • Open (default setting): MCU-Link VCOM port (USB-to-UART bridge) is enabled. • Shorted: MCU-Link VCOM port (USB-to-UART bridge) is disabled.	Section 3.7
JP4	SWD_DIS	1x2-pin header	MCU-Link SWD disable jumper: • Open (default setting): MCU-Link SWD feature is enabled. MCU-Link can be used to drive SWD of the target MCU. • Shorted: MCU-Link SWD feature is disabled. This jumper setting can be used for debugging the target MCU, using an external debugger connected through connector J10.	Section 3.2
JP5	SWDCLK	1x2-pin header	MCU-Link SWD clock enable jumper: • Open: MCU-Link SWD clock is disabled. • Shorted (default setting): MCU-Link SWD clock is enabled. MCU-Link drives SWD of the target MCU.	For more information on these jumpers, see FRDM-MCXC444 board schematics.
JP6 (DNP)	-	1x2-pin header	• Open (default setting): Use internal 1.2 V voltage reference (VREF). • Shorted: Use external reference source or MCU 3.3 V supply as voltage reference. ***Note:*** *When the internal 1.2 V VREF module is enabled, VREFH supply produced on the board should not be supplied to the VREFH pin, so the jumper JP6 should be kept open. For details, refer to the "MCX C44X Sub-Family Reference Manual".*	Section 2.1
JP7	-	1x2-pin header	• Open: The P3V3_MCU supply disconnects from the VLL3 pin. • Shorted (default setting): The P3V3_MCU supply connects to the VLL3 pin. ***Note:*** *When the internal LCD charge pump is enabled, the jumper JP7 must be kept open. For details, refer to the "MCX C44X Sub-Family Reference Manual".*	Section 2.1

1.7 Push buttons

Figure 3 shows the FRDM-MCXC444 board push buttons.

Table 5 describes the FRDM-MCXC444 board push buttons.

Table 5. FRDM-MCXC444 push buttons

Part identifier	PCB label	Name/function	Description
SW1	RESET	Reset button	When pressed, resets the MCXC444 MCU. SW1 is connected to the PTA20 pin on MCXC444. ***Note:*** *Apart from SW1, other sources supplying reset signal to MCXC444 are MCU-LINK through the DBGIF_RESET signal and an arduino shield connected to the J3 header that sends a reset signal (RESET_B) through pin 6.*
SW2	WAKEUP	Wakeup button	Pressing SW2 asserts the MCXC444 MCU pin PTC3/ LLWU_P7, which wakes the Low-Leakage Wakeup Unit (LLWU) module of the MCU.
SW3	ISP	In-system programming (ISP) / non-maskable interrupt (NMI) button	Forces the MCXC444 MCU to boot from the ROM bootloader, instead of booting from internal flash memory. To boot the MCU from the ROM bootloader, hold down SW3 while pressing SW1 (reset button) or while supplying power to the board.

1.8 LEDs

The FRDM-MCXC444 board provides light-emitting diodes (LEDs) for monitoring system status. The information collected from the LEDs can be used for debugging purposes.

Figure 3 shows the FRDM-MCXC444 board LEDs.

Table 6 describes the FRDM-MCXC444 board LEDs that correspond to the target MCU. The board also has some MCU-Link-specific LEDs, which are described in Section 3.9.

Table 6. FRDM-MCXC444 LEDs

Part identifier	PCB label	LED color	LED name/function	Description (when LED is ON)
D1	RESET	Red	Reset LED	Indicates system reset activity. When board reset is initiated, for example, by pressing the reset button (SW1), D1 turns ON.
D3	RGB	Red/green/blue	RGB LED	User application LED
D5	POWER	Green	Power-on indicator LED	Indicates that P3V3 supply is available and the FRDM-MCXC444 board is powered up.

2 Functional description

This section describes the features and functions of the FRDM-MCXC444 board. The functionality described in this section can be used as a reference while designing your own target board.

Note:

For details on the MCXC444VLH MCU features, see MCX C44X Microcontroller Data Sheet and MCX C44X Sub-Family Reference Manual Reference Manual.

This section contains the following subsections:

- Section 2.1 "Power supplies"
- Section 2.2 "Clocks"
- Section 2.3 "USB interface"
- Section 2.4 "LPUART interface"

- Section 2.5 "SPI interface"
- Section 2.6 "I2C interface"
- Section 2.7 "LCD interface"
- Section 2.8 "Accelerometer"
- Section 2.9 "Visible light sensor interface"
- Section 2.10 "Arduino socket"
- Section 2.11 "mikroBUS socket"
- Section 2.12 "Pmod connector"

2.1 Power supplies

The FRDM-MCXC444 board is powered with a P5V0 (5 V) power supply using one of the following source options:

- P5V_USB_FS supply from full-speed (FS) USB2.0 Type-C connector (J12)
- P5V_HDR_IN supply from 5 V regulator populated at 3-pin connector (J11) (Not populated by default)
- P5V_MCU_LINK_USB supply from MCU-Link USB2.0 Type-C connector (J13)

The P5V0 supply is an input power supply on the board and is a source for secondary power supplies. The secondary power supplies provide power to board components, including the MCXC444 MCU, MCU-Link, accelerometer, Arduino socket, mikroBUS socket, Pmod connector, and external debugger connector.

Table 7 describes the FRDM-MCXC444 board power supplies.

Table 7. FRDM-MCXC444 power supplies

Power source	Manufacturer and part number	Power supply	Description
External supply through MCU-Link USB Type-C connector J13	-	P5V_MCU_LINK_USB (5 V)	• One of the three power source options (default option) for the P5V0 supply • Provides the USB1_VBUS power to the LPC55S69 MCU (MCU-Link)
External supply through Arduino socket connector J3, pin 16	-	P5V0-9V0_VIN (5 V – 9 V)	Supplies power to 5 V DC voltage regulator J11 (DNP)
DC voltage regulator J11 (DNP)	-	P5V_HDR_IN (5 V)	• One of the three power source options for the P5V0 supply
External supply through MCU USB Type-C connector J12	-	P5V_USB_FS (5 V)	• One of the three power source options for the P5V0 supply
From the P5V_MCU_LINK_USB / P5V_USB_FS / P5V_HDR_IN supply ***Note:*** *By default, the option to produce the P5V0 supply from the P5V_HDR_IN supply is disabled.*	-	P5V0 (5 V)	Supplies power to LDO voltage regulator U3, Arduino socket connector J3 (pin 10), mikroBUS socket connector J5 (pin 7), and MCU VREGIN pin (USB VREG supply)
LDO voltage regulator U3	Torex Semiconductor XC6227C331PR-G	P3V3 (3.3 V)	• Supplies power to: – Accelerometer U2 – Light sensor Q1 – Arduino socket connector J3 (pins 4 and 8) – mikroBUS socket connector J6 (pin 7) – Pmod connector J7 (DNP) – Push buttons SW1, SW2, and SW3

Table 7. FRDM-MCXC444 power supplies*...continued*

Power source	Manufacturer and part number	Power supply	Description
			– LEDs D1, D3, and D5
P3V3	-	VDD_BOARD	• Produces the MCU_LINK_3V3 and VREF_ MCULINK supplies • Produces the P3V3_MCU supply through jumper JP1 • Source of MCU_Link supplies (MCU_LINK_3 V3, VBAT_DCDC/VBAT_PMU, MCULink_VDDA, VREF_MCULINK, DBGIF_VREF) • MCU-Link LEDs (D8, D9, and D10)
From the VDD_BOARD supply through jumper JP1	-	P3V3_MCU (3.3 V)	Supplies to: • MCXC444 MCU digital and analog power pins (VDD1, VDD2, VDDA) • MCXC444 MCU VLL3 pin through jumper JP7 • External debugger connector J10 • Source of VREFH_IN supply

2.1.1 Current measurement

The FRDM-MCXC444 board supports measurement of the digital supply current of the MCXC444 MCU, using an ampere meter (ammeter). The current measurement steps are as follows:

1. Open the 2-pin jumper JP1 (shorted by default).
2. Connect an ammeter through the pins of JP1.

2.2 Clocks

Table 8 describes the clocks available on the FRDM-MCXC444 board.

Table 8. FRDM-MCXC444 clocks

Clock generator	Manufacturer and part number	Clock	Frequency	Destination
Crystal Y1	Abracon LLC ABS07-32.768KHZ-T	[XTAL, EXTAL]_32 KHZ	32.768 kHz	MCXC444 MCU (System oscillator, low frequency range mode)
Crystal Y2	KYOCERA AVX CX3225GA16000 D0PTVCC	MCU_LINK_[P, N]_16 MHz	16 MHz	LPC55S69 MCU

2.3 USB interface

The target MCU (MCXC444) features one full-speed (FS) USB 2.0 device controller. On the FRDM-MCXC444 board, the FS USB controller connects to the USB Type-C connector (J12). This connector works in device mode and is used to provide the 5 V power supply (P5V_USB_FS) to the board.

2.4 LPUART interface

The MCXC444 MCU features two low-power UART modules (LPUART0 and LPUART1) supporting asynchronous operation in low-power modes.

The FRDM-MCXC444 board allows communication with both the LPUART0 and LPUART1 modules.

Figure 5 shows the FRDM-MCXC444 LPUART interface diagram.

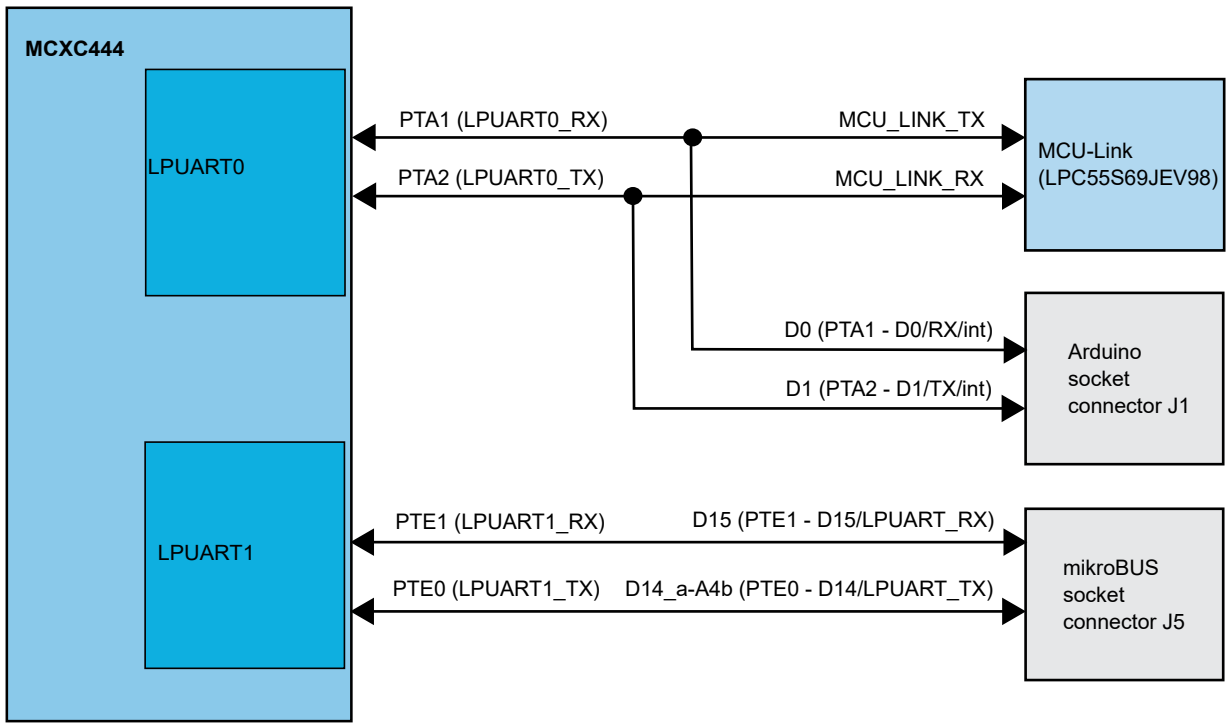

Figure 5. LPUART interface diagram

Table 9 describes the FRDM-MCXC444 LPUART connections.

Table 9. LPUART connections

LPUART module	Peripheral devices		
	Part identifier	Manufacturer and part number	Description
LPUART0	U5	NXP LPC55S69JEV98	Onboard MCU-Link — A 32-bit MCU based on the Arm Cortex-M33 core with speeds of up to 150 MHz. MCU-Link acts as a USB-to-UART bridge between the host computer and the target MCU (MCXC444) for debugging the target MCU. For more details, see Section 3.7.
	J1	-	One of the four Arduino socket connectors that allows the plugged-in Arduino board to communicate with the LPUART0 module of the MCXC444 MCU.
LPUART1	J5	-	One of the two mikroBUS socket connectors that allows the plugged-in mikroBUS click board to communicate with the LPUART1 module of the MCXC444 MCU.

2.5 SPI interface

The MCXC444 MCU features two 16-bit SPI modules, SPI0 and SPI1. These modules support the following four signals:

- Serial clock (SCK)
- Master input / slave output (MISO)
- Master output / slave input (MOSI)

- Slave select (SS)

The FRDM-MCXC444 board allows communication with both SPI0 and SPI1 modules.

Figure 6 shows the FRDM-MCXC444 SPI interface diagram.

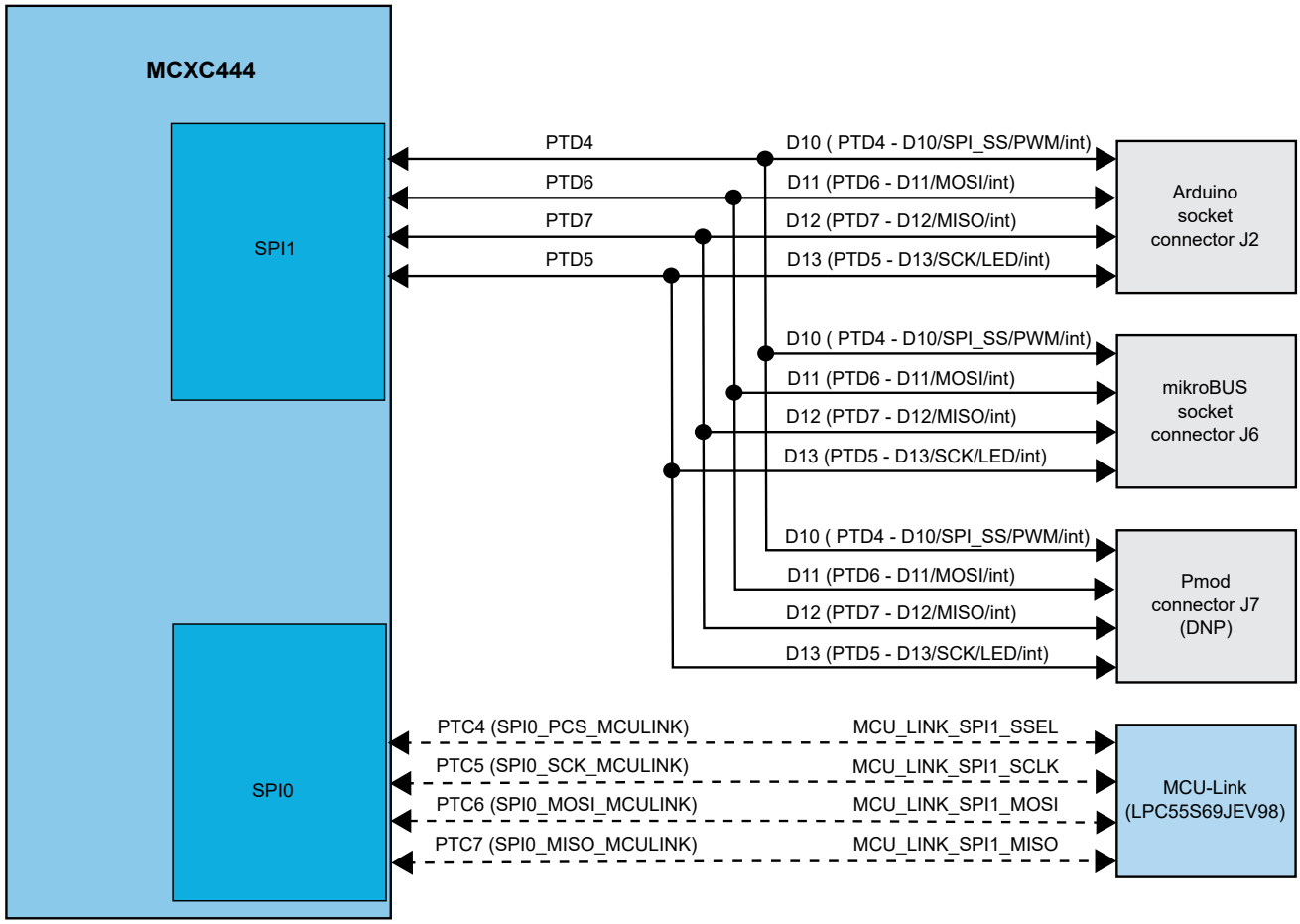

Figure 6. SPI interface diagram

Table 10 describes the FRDM-MCXC444 SPI connections.

Table 10. SPI connections

SPI module	Peripheral devices		
	Part identifier	Manufacturer and part number	Description
SPI1	J2		One of the four Arduino socket connectors that allows the plugged-in Arduino board to communicate with the SPI1 module of the MCXC444 MCU.
	J6		One of the two mikroBUS socket connectors that allows the plugged-in mikroBUS click board to communicate with the SPI1 module of the MCXC444 MCU.
	J7 (DNP)	Sullins Connector Solutions PPPC062LJBN-RC	Pmod connector

Table 10. SPI connections*...continued*

SPI module	Peripheral devices		
	Part identifier	Manufacturer and part number	Description
SPI0	U5	NXP LPC55S69JEV98	Onboard MCU-Link — A 32-bit MCU based on the Arm Cortex-M33 core with speeds of up to 150 MHz. MCU-Link can act as a USB-to-SPI bridge between the host computer and the target MCU (MCXC444) for debugging the target MCU. By default, the SPI connection between MCU-Link and the target MCU is disabled. It can be enabled by populating the following 0 Ω resistors: • R72 • R73 • R74 • R75

2.6 I2C interface

The MCXC444 MCU has two inter-integrated circuit (I2C) modules, I2C0 and I2C1, which support serial I2C communication through a pair of clock and data signals.

The FRDM-MCXC444 board allows communication with both I2C0 and I2C1 modules.

Figure 7 shows the FRDM-MCXC444 I2C diagram.

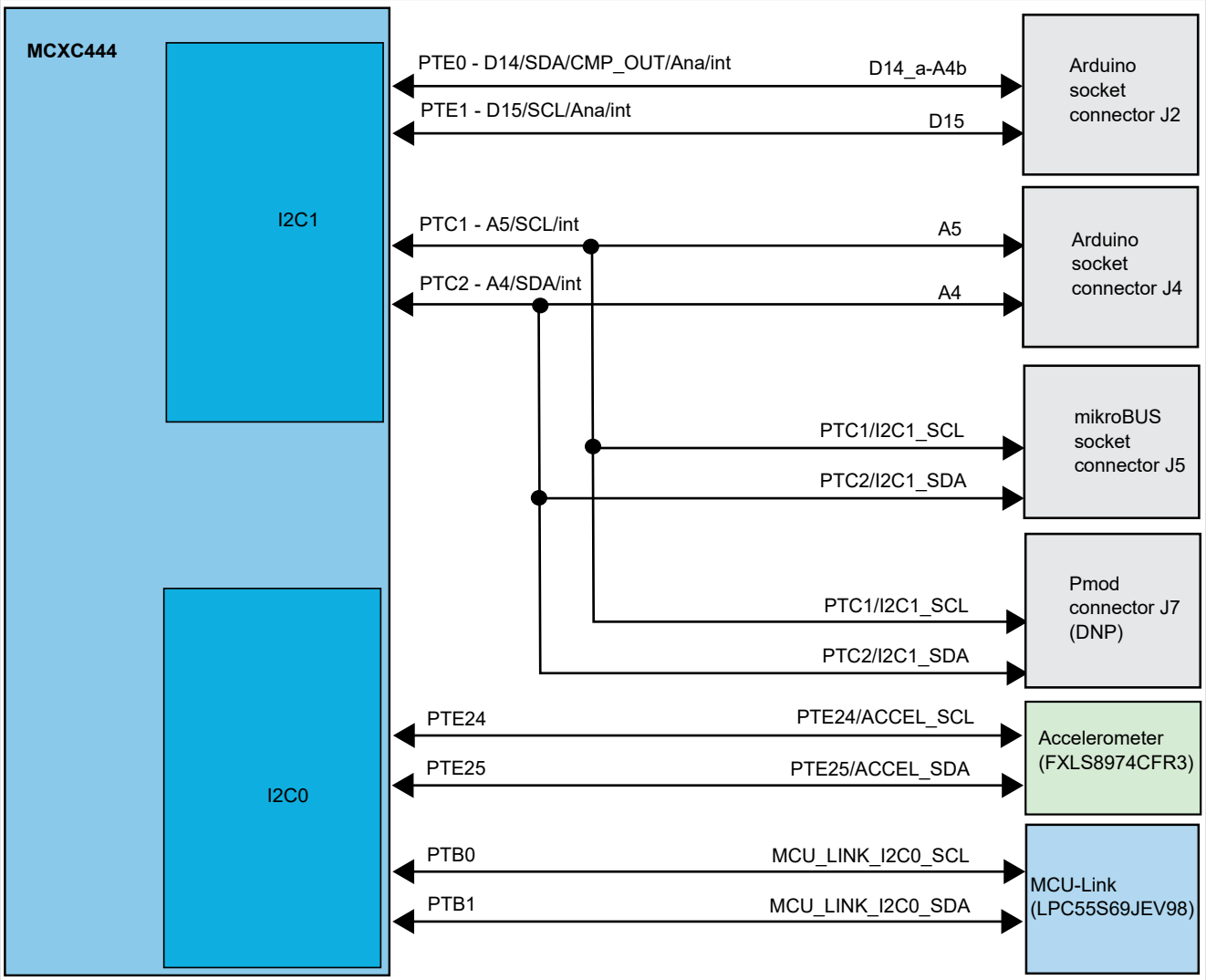

Figure 7. I2C diagram

Table 11 shows the FRDM-MCXC444 I2C bus device map.

Table 11. I2C bus device map

I2C bus	8-bit I2C address	Device	Description
I2C0, I2C1	-	NXP MCXC444 (U1)	Target MCU. It acts as the I2C master for all I2C connections on the board except for I2C connection with MCU-Link.
I2C0	0x30 [1]	NXP FXLS8974CFR3 (U2)	Accelerometer
I2C1	The I2C address depends on the plugged-in Arduino board.	Arduino board	Board/module attached to the Arduino socket comprising connectors J1, J2, J3, and J4. The I2C signals connect through J4 connector pins: 10 (SDA) and 12 (SCL) and J2 connector pins: 20 (SCL) and 18 (SDA).
I2C1	The I2C address depends on the	mikroBUS click board	Board/module attached to the mikroBUS socket comprising connectors J5 and J6. The I2C signals

Table 11. I2C bus device map...*continued*

I2C bus	8-bit I2C address	Device	Description
	plugged-in mikro BUS click board.		connect through J5 connector pins: 5 (SCL) and 6 (SDA).
I2C1	The I2C address depends on the plugged-in Pmod board.	Pmod board	Board/module attached to the Pmod connector J7 (not populated by default). The I2C signals connect through J7 pins 6 (SDA) and 8 (SCL).
I2C0	-	NXP LPC55S69JEV98 (U5)	Onboard MCU-Link — A 32-bit MCU based on the Arm Cortex-M33 core with speeds of up to 150 MHz. MCU-Link can act as a USB-to-I2C bridge between the host computer and the target MCU (MCXC444) for debugging the target MCU. By default, the I2C0 connection between MCU-Link and the target MCU is disabled. It can be enabled by populating the following 0 Ω resistors: • R78 • R79

[1] For other I2C addresses details, see Section 2.8.

2.7 LCD interface

The MCXC444 MCU features a one segment liquid crystal display (SLCD) module that can support up to 24x8 or 28x4 segments.

The FRDM-MCXC444 board supports one 7-segment LCD panel that uses seven different segments to form different numbers, letters, and some special characters. The MCU SLCD module controls the illumination of these segments through front plane/back plane signaling.

Table 12 describes the SCLD module available on the FRDM-MCXC444 board.

Table 12. SCLD module description

Part identifier	Part number and manufacturer name	Description
DS1	LCD-S401M16KR from Lumex	12-pin, 4 Digit, 7 Segment LCD Panel

Figure 8 shows the FRDM-MCXC444 SLCD diagram.

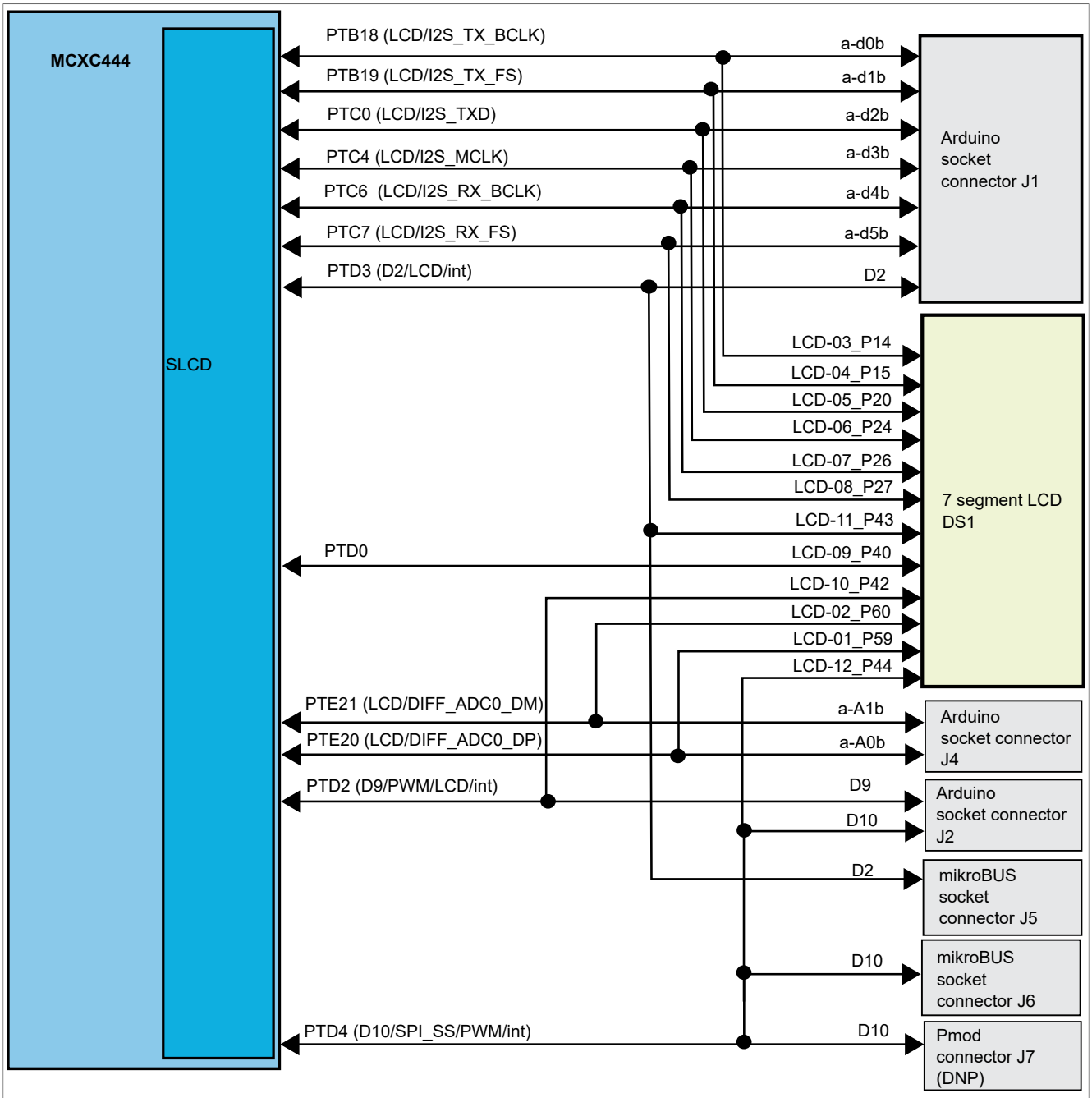

Figure 8. SLCD diagram

Table 13 describes the FRDM-MCXC444 SLCD module connections.

Table 13. SCLD module (DS1) pinout

Pin number	GPIO	LCD signal	Description
1	PTE20	LCD-01_P59	Common anode (COM0)
2	PTE21	LCD-02_P60	Common anode (COM1)
3	PTB18	LCD-03_P14	Common anode (COM2)
4	PTB19	LCD-04_P15	Common anode (COM3)

Table 13. SCLD module (DS1) pinout ...continued

Pin number	GPIO	LCD signal	Description
5	PTC0	LCD-05_P20	Data signal to 1D/1E/1G/1F
6	PTC4	LCD-06_P24	Data signal to 1DP/1C/1B/1A
7	PTC6	LCD-07_P26	Data signal to 2D/2E/2G/2F
8	PTC7	LCD-08_P27	Data signal to 2DP/2C/2B/2A
9	PTD0	LCD-09_P40	Data signal to 3D/3E/3G/3F
10	PTD2	LCD-10_P42	Data signal to 3DP/3C/3B/3A
11	PTD3	LCD-11_P43	Data signal to 4D/4E/4G/4F
12	PTD4	LCD-12_P44	Data signal to COL/4C/4B/4A

2.8 Accelerometer

On the FRDM-MCXC444 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 MEMS accelerometer sensor device FXLS8974CFR3 (U2) is used.
- The sensor device is powered by the P3V3 supply.
- 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 (default setting)
  - 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 (R37 and R38) are provided to isolate the sensor from the MCXC444 device.

Figure 9 shows the FXLS8974CFR3 sensor circuit diagram.

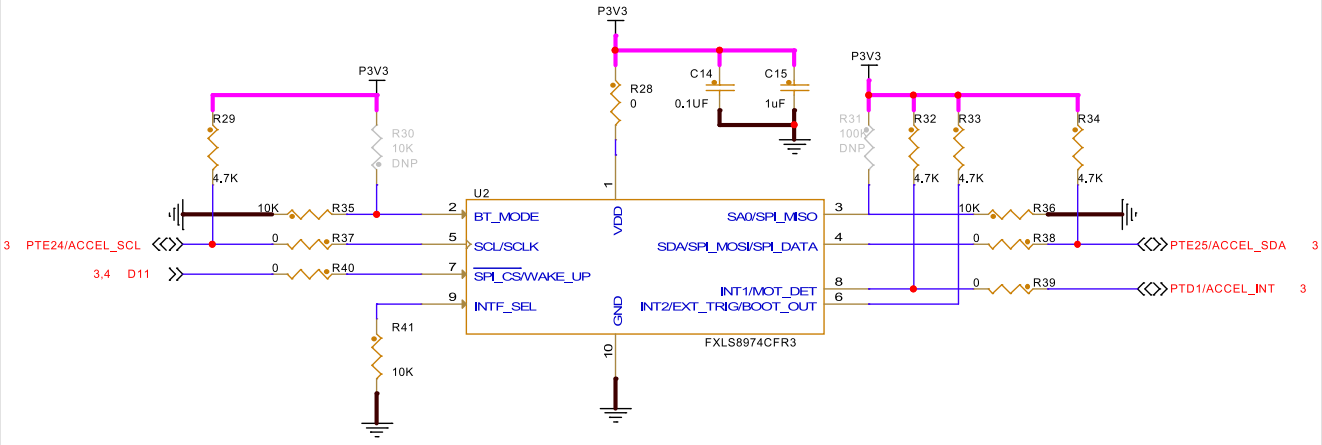

Figure 9. FXLS8974CFR3 sensor circuit

For more information on FXLS8974CFR3, visit nxp.com.

2.9 Visible light sensor interface

On the FRDM-MCXC444 board, one light sensor is provided, which connects to the PTE22 port of the target device for evaluating the ADC module.

Table 14 provides the detail of the light sensor device on the board.

Table 14. Light sensor device

Part identifier	Manufacturer and part name	Description
Q1	Everlight ALS-PT19-315C/L177/TR8	It is a low-cost ambient light sensor, consisting of phototransistor in miniature SMD.

Figure 10 shows the schematic diagram of the visible light sensor on the board.

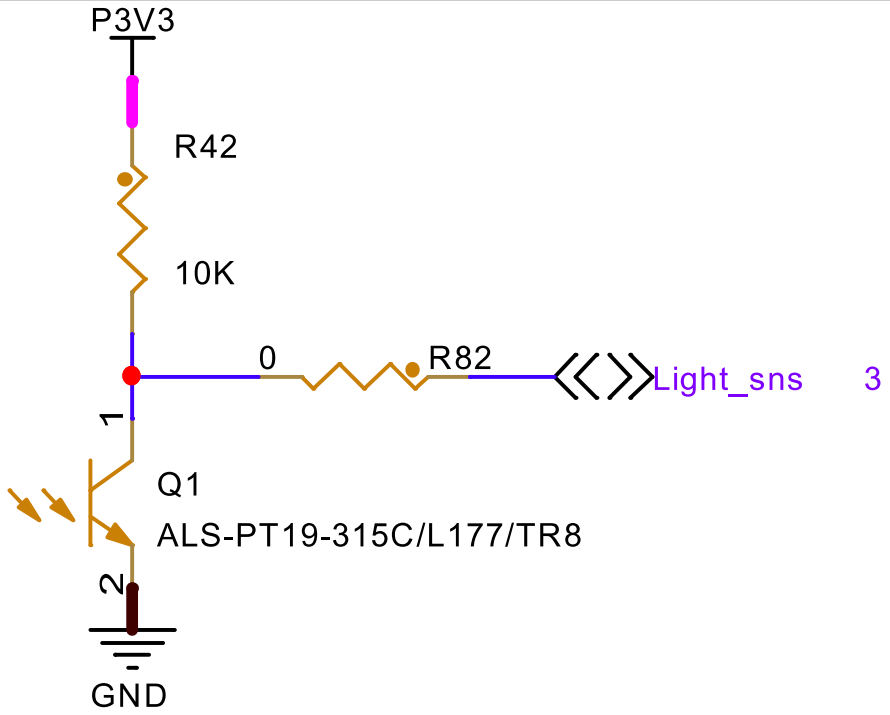

Figure 10. Visible light sensor schematic diagram

The zero-ohm resistor R82 is provided to isolate the light sensor from the MCXC444 device. When no light reaches the sensor, the Light_sns signal is at low level and when proper light reaches the sensor, the Light_sns signal is at high level.

2.10 Arduino socket

FRDM-MCXC444 supports I/O headers that 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
- Motor control: FRDM-MC-LVBLDC, FRDM-MC-LVPMSM
- Audio: ARD-AUDIO-DA7212

Table 15 describes the connectors of the Arduino socket.

Table 15. Arduino socket connectors

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

The Arduino socket allows communication with the following modules of the MCXC444 MCU:

- Low-power universal asynchronous receiver/transmitter (LPUART)
- Serial peripheral interface (SPI)
- Inter-integrated circuit (I2C)
- Analog-to-digital converter (ADC)
- Timer/PWM module (TPM)
- Inter-IC sound (I2S)

The Table 16, Table 17, Table 18, and Table 19 explain the I/O headers pinout on FRDM-MCXC444.

Table 16. J1 connector (at right-upper side) pinout

I/O header pin	MCXC444 pin	Arduino / FRDM functions	Potential conflict
J1[1]	PTB18	I2S_TX_BCLK	• 7-segment LCD (LCD-03_P14)
J1[2]	PTA1	D0 / RX / int	• MCU-Link VCOM port (UART_RX_TGTMCU)
J1[3]	PTB19	I2S_TX_FS	• 7-segment LCD (LCD-04_P15)
J1[4]	PTA2	D1 / TX / int	• MCU-Link VCOM port (UART_TX_TGTMCU)
J1[5]	PTC0	I2S_TXD	7-segment LCD (LCD-05_P20)
J1[6]	PTD3	D2 / int	• 7-segment LCD (LCD-11_P43) • mikroBUS header J5 pin 2
J1[7]	PTC4	I2S_MCLK	• MCU-Link USBSIO port (PTC4/SPI0_PCS_MCULINK) • 7-segment LCD (LCD-06_P24)
J1[8]	PTA12	D3 / PWM / int	• mikroBUS header J5 pin 1
J1[9]	PTC6	I2S_RX_BCLK	• MCU-Link USBSIO port (PTC6/SPI0_MOSI_MCULINK) • 7-segment LCD (LCD-07_P26)
J1[10]	PTA4	D4 / int	• SW3 button (PTA4/NMI)
J1[11]	PTC7	I2S_RX_FS/SOF_OUT	• MCU-Link USBSIO port (PTC7/SPI0_MISO_MCULINK) • 7-segment LCD (LCD-08_P27)
J1[12]	PTA5	D5 / PWM / int	• Pmod connector J7 (DNP) pin 2
J1[13]	-	-	-
J1[14]	PTE29	D6 / PWM / CMP / int	• Pmod connector J7 (DNP) pin 4
J1[15]	PTC5	I2S_RXD	• MCU-Link USBSIO port (PTC5/SPI0_SCK_MCULINK)

Table 16. J1 connector (at right-upper side) pinout*...continued*

I/O header pin	MCXC444 pin	Arduino / FRDM functions	Potential conflict
J1[16]	PTE30	D7 / CMP / int	• Arduino header J4 pin 11 • mikroBUS header J6 pin 2

Table 17. J2 connector (at left-upper side) pinout

I/O header pin	MCXC444 pin / function	Arduino / FRDM functions	Potential conflict
J2[1]	-	-	-
J2[2]	PTA13	D8 / InputCapture / int	-
J2[3]	-	-	-
J2[4]	PTD2	D9 / PWM / int	7-segment LCD (LCD-10_P42)
J2[5]	-	-	-
J2[6]	PTD4	D10 / SPI_SS / PWM / int	• 7-segment LCD (LCD-12_P44) • Pmod connector J7 (DNP) pin 1 • mikroBUS header J6 pin 3
J2[7]	-	-	-
J2[8]	PTD6	D11 / MOSI / int	• Pmod connector J7 (DNP) pin 3 • mikroBUS header J6 pin 6
J2[9]	-	-	-
J2[10]	PTD7	D12 / MISO / int	• Pmod connector J7 (DNP) pin 5 • mikroBUS header J6 pin 5
J2[11]	-	-	-
J2[12]	PTD5	D13 / SCK / LED / int	• Pmod connector J7 (DNP) pin 7 • mikroBUS header J6 pin 4
J2[13]	-	-	-
J2[14]	-	GND	-
J2[15]	-	-	-
J2[16]	-	AREF	-
J2[17]	PTB17	-	-
J2[18]	PTE0	D14 / SDA / Ana / int	• Arduino header J4 pin 9 • mikroBUS header J5 pin 4
J2[19]	PTB16	-	-
J2[20]	PTE1	D15 / SCL / Ana / int	• mikroBUS header J5 pin 3

Table 18. J3 connector (at left-lower side) pinout

I/O header pin	MCXC444 pin / function	Arduino / FRDM functions	Potential conflict
J3[1]	-	-	-

Table 18. J3 connector (at left-lower side) pinout*...continued*

I/O header pin	MCXC444 pin / function	Arduino / FRDM functions	Potential conflict
J3[2]	-	-	-
J3[3]	-	-	-
J3[4]	-	GND	-
J3[5]	-	-	-
J3[6]	-	RESET_B	-
J3[7]	-	-	-
J3[8]	-	P3V3	-
J3[9]	-	-	-
J3[10]	-	P5V0	-
J3[11]	-	-	-
J3[12]	-	GND	-
J3[13]	-	-	-
J3[14]	-	GND	-
J3[15]	-	-	-
J3[16]	-	P5-9V_VIN	-

Table 19. J4 connector (at right-lower side) pinout

I/O header pin	MCXC444 pin / function	Arduino / FRDM functions	Potential conflict
J4[1]	PTE20	DIFF_ADC0_DP	• 7-segment LCD (LCD-01_P59)
J4[2]	PTB0	A0 / Int	• MCU-Link USBSIO port (PTB0/I2C0_SCL-MCULINK) • mikroBUS header J6 pin 1
J4[3]	PTE21	DIFF_ADC0_DM	• 7-segment LCD (LCD-02_P60)
J4[4]	PTB1	A1 / Int	• MCU-Link USBSIO port (PTB1/I2C0_SDA-MCULINK)
J4[5]	PTE22	DIFF_ADC1_DP	• Ambient light sensor (Light_sns)
J4[6]	PTB2	A2 / int	-
J4[7]	PTE23	DIFF_ADC1_DM	-
J4[8]	PTB3	A3 / int	-
J4[9]	PTE0	CMP_OUT	• Arduino header J2 pin 18 • mikroBUS header J5 pin 4
J4[10]	PTC2	A4 / SDA / int	• Pmod connector J7 (DNP) pin 8 • mikroBUS header J5 pin 6
J4[11]	PTE30	DAC_OUT	• Arduino header J1 pin 16 • mikroBUS header J6 pin 2
J4[12]	PTC1	A5 / SCL / int	• Pmod connector J7 (DNP) pin 6

Table 19. J4 connector (at right-lower side) pinout...*continued*

I/O header pin	MCXC444 pin / function	Arduino / FRDM functions	Potential conflict
			• mikroBUS header J5 pin 5

2.11 mikroBUS socket

A mikroBUS socket is a pair of 1x8-position receptacles (connectors) with a proprietary pin configuration and silkscreen markings. It provides many hardware expansion options with few pins. An add-on board called a *click board*, can be installed on a mikroBUS socket. A click board provides a plug-and-play solution for adding new functionality to a board design.

A click board has a pair of 1x8-pin headers that connects to the pair of receptacles on a mikroBUS socket. MikroElektronika (MIKROE) is one of the manufacturers of click boards. To find some example click boards for the FRDM-MCXC444 mikroBUS socket, visit MIKROE website.

The FRDM-MCXC444 board has a mikroBUS socket with a pair of 1x8-position receptacles, J5 and J6. Figure 11 shows the pinouts of the mikroBUS socket connectors.

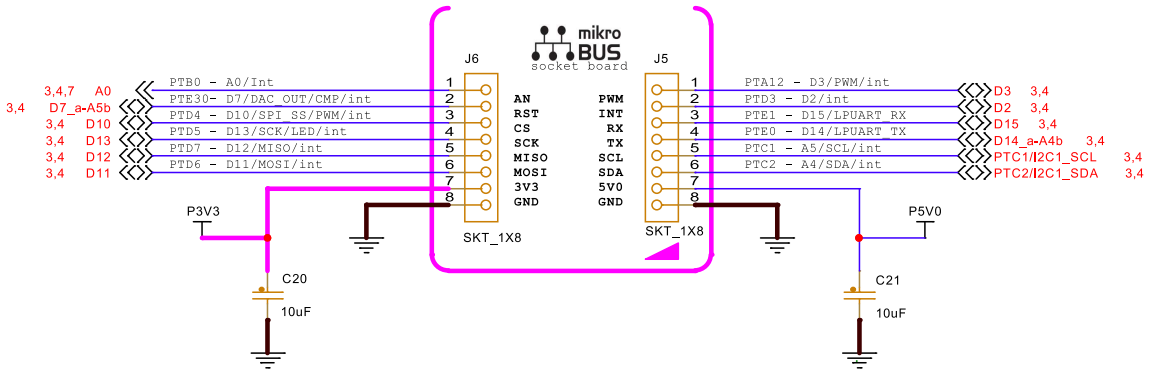

Figure 11. mikroBUS socket connector pinout

The mikroBUS socket allows communication with the following modules of the MCXC444 MCU:

- Low-power universal asynchronous receiver/transmitter (LPUART)
- Serial peripheral interface (SPI)
- Inter-integrated circuit (I2C)
- Analog-to-digital converter (ADC)

Table 20 explains the pinout of the mikroBUS headers J5 and J6 on FRDM-MCXC444.

Table 20. mikroBUS header pinout

I/O header pin	MCXC444 pin / function	mikroBUS function	Potential conflict
J5 header pinout			
J5[1]	PTA12 - D3/PWM/int	PWM	• Arduino header J1 pin 8
J5[2]	PTD3 - D2/int	INT	• Arduino header J1 pin 6 • 7-segment LCD (LCD-11_P43)
J5[3]	PTE1 - D15/LPUART_RX	RX	• Arduino header J2 pin 20
J5[4]	PTE0 - D14/LPUART_TX	TX	• Arduino header J2 pin 18 • Arduino header J4 pin 9
J5[5]	PTC1 - A5/SCL/int	SCL	• Arduino header J4 pin 12

Table 20. mikroBUS header pinout...*continued*

I/O header pin	MCXC444 pin / function	mikroBUS function	Potential conflict
			• Pmod header J7 pin 6
J5[6]	PTC2 - A4/SDA/int	SDA	• Arduino header J4 pin 10 • Pmod header J7 pin 8
J5[7]	P5V0	5V0	-
J5[8]	GND	GND	-
J6 header pinout			
J6[1]	PTB0 - A0/Int	AN	• Arduino header J4 pin 2 • MCU-Link USBSIO port (PTB0/I2C0_SCL-MCULINK)
J6[2]	PTE30- D7/DAC_OUT/CMP/int	RST	• Arduino header J1 pin 16 (PTE30-D7/CMP/int) • Arduino header J4 pin 11 (PTE30 - DAC_OUT)
J6[3]	PTD4 - D10/SPI_SS/PWM/int	CS	• Arduino header J2 pin 6 • 7-segment LCD (LCD-12_P44) • Pmod header J7 pin 1
J6[4]	PTD5 - D13/SCK/LED/int	SCK	• Arduino header J2 pin 12 • Pmod header J7 pin 7
J6[5]	PTD7 - D12/MISO/int	MISO	• Arduino header J2 pin 10 • Pmod header J7 pin 5
J6[6]	PTD6 - D11/MOSI/int	MOSI	• Arduino header J2 pin 8 • Pmod header J7 pin 3
J6[7]	P3V3	3V3	-
J6[8]	GND	GND	-

2.12 Pmod connector

Digilent Pmod (peripheral module) devices are small input/output interface boards that can be easily integrated with embedded control boards for expanding their capabilities.

The FRDM-MCXC444 board supports a Digilent PPPC062LJBN-RC Pmod connector J7 (not populated) for expanding the board capabilities.

Table 21 shows the pinout of the Pmod connector J7.

Table 21. Pmod connector pinout

Pin number	MCXC444 pin name	Function	Potential conflict
1	PTD4	D10/LCD/SPI_SS/PWM/int	• Arduino header J2 pin 6 • mikroBUS header J6 pin 3 • 7-segment LCD (LCD-12_P44)
2	PTA5	D5/PWM/int	• Arduino header J1 pin 12
3	PTD6	D11/MOSI/int	• Arduino header J2 pin 8 • mikroBUS header J6 pin 6
4	PTE29	D6/PWM/CMP/RESET	• Arduino header J1 pin 14
5	PTD7	D12/MISO/int	• Arduino header J2 pin 10

Table 21. Pmod connector pinout...*continued*

Pin number	MCXC444 pin name	Function	Potential conflict
			• mikroBUS header J6 pin 5
6	PTC1	A5/SCL/int	• Arduino header J4 pin 12 • mikroBUS header J5 pin 5
7	PTD5	D13/SCK/LED/int	• Arduino header J2 pin 12 • mikroBUS header J6 pin 4
8	PTC2	A4/SDA/int	• Arduino header J4 pin 10 • mikroBUS header J5 pin 6
9, 10	GND	-	-
11, 12	P3V3	-	-

3 MCU-Link OB debug probe

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 for onboard debug probes in evaluation boards (such as FRDM-MCXC444). The onboard implementation of MCU-Link is referred to as *MCU-Link OB*.

The FRDM-MCXC444 board implements a subset of the MCU-Link architecture features, as mentioned in Section 3.1. For more details on the MCU-Link architecture, visit the MCU-Link Debug Probe Architecture page.

The MCU-Link OB on the FRDM-MCXC444 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 3.4.

3.1 Supported MCU-Link features

MCU-Link includes several mandatory and optional features. Table 22 summarizes the MCU-Link features supported on the FRDM-MCXC444 board.

Table 22. Supported MCU-Link features

Feature	Description
Serial wire debug (SWD) / serial wire debug trace output (SWO)	MCU-Link allows SWD-based debugging with SWO for profiling and/or low overhead debug standard I/O communication.
Virtual communication (VCOM) serial port	MCU-Link adds a serial COM port on the host computer and connects it to the target MCU, while acting as a USB-to-UART bridge.
USB serial input/output (USBSIO[1])	MCU-Link adds a USB serial I/O port on the host computer and connects it to the target MCU, while acting as a USB-to-SPI or USB-to-I2C bridge.
External debug probe support	The MCU-Link interface supports debugging the target MCU (MCXC444) using an external debug probe, instead of MCU-Link. Support for an external debug probe is enabled by disabling the SWD feature.

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

3.2 Supported debug scenarios

Table 23 describes the debug scenarios supported on the FRDM-MCXC444 board.

Table 23. Supported debug scenarios

Debug scenario	Feature support	Required jumper / connector settings
Use MCU-Link for debugging the MCXC444 MCU	SWD: Enabled	MCU-Link SWD disable jumper JP4 is open.
		Target MCU SWD connector J10 is not used for external connection.
	VCOM: Enabled	MCU-Link VCOM port disable jumper JP3 is open.
	USBSIO: Enabled	Do not populate (DNP) R64 resistor (1 kΩ).
Use an external debugger for debugging the MCXC444 MCU	SWD: Disabled	Short JP4.
		Connect the external debugger to J10.
	VCOM: Enabled	JP3 is open.
	USBSIO: Enabled	DNP R64 resistor (1 kΩ).

3.3 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 3.4.*

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 3.4. 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 24. 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 24. Compatibility check between MCUXpresso IDE and MCU-Link firmware

MCUXpresso IDE version	Supported MCU-Link firmware version	USB driver type	CMSIS-SWO support	FreeMASTER support via	
				SWD / JTAG	USB bridge
MCUXpresso 11.3 or later	V1.xxx and V2.xxx	HID	No	Yes	Yes
MCUXpresso 11.7.0 or later	V3.xxx (up to and including V3.108)	WinUSB	No	Yes	FreeMASTER V3.2.2 or later
MCUXpresso 11.7.1 or later	V3.117 and later	WinUSB	Yes	Yes	FreeMASTER V3.2.2 or later

3.4 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 JP2, and reconnect the board. The red MCU-Link status LED D9 lights up and stays on. For more details on MCU-Link LEDs, see Section 3.9.
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 CMSIS-DAP or J-Link firmware version.
5. Disconnect the board from the host computer, open jumper JP2, and reconnect the board. The board enumerates on the host computer as a WinUSB or HID device (depending on the firmware version, see Table 24).

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.*

Note: *To enable SWO-related features in non-NXP IDEs, CMSIS-SWO support was introduced in firmware version V3.117.*

3.5 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.

3.5.1 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 3.4.

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 installed on your host computer. To check the compatibility of the MCU-Link firmware you want to use with your MCUXpresso IDE, see Table 24.

3.5.2 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 3.3. For more details on MCUXpresso for Visual Studio Code, visit the MCUXpresso for Visual Studio Code page.

3.5.3 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).

3.6 MCU-Link USB connector

The FRDM-MCXC444 board has a USB Type-C connector J13, which allows you to connect MCU-Link with your host computer. It can also be used to supply 5 V power to the board.

3.7 VCOM port (USB to target UART bridge)

MCU-Link supports a feature known as *virtual communication (VCOM) serial port*. This feature allows MCU-Link to add a serial COM port on the host computer. The, MCU-Link acts as a USB-to-UART bridge between the host computer and the target MCU.

In the FRDM-MCXC444 board, MCU-Link is connected to the LPUART0 module of the target MCU. To use MCU-Link as a USB-to-UART bridge, verify the following jumper settings and connect the MCU-Link USB connector J13 to the USB port of the host computer:

- Jumper JP2 is open (MCU-Link boots normally)
- Jumper JP3 is open (MCU-Link VCOM port is enabled)

When you boot the FRDM-MCXC444 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 VCOM function can be disabled by shorting jumper JP3 before powering up the board. Changing the JP3 setting (open/short) after powering up the board has no impact on the MCU-Link VCOM function.

3.8 USBSIO port (USB to target SPI/I2C bridge)

MCU-Link supports a feature known as *USB serial input/output (USBSIO) port*. This feature allows MCU-Link to add a USB serial I/O port on the host computer. Then, MCU-Link acts as a USB-to-SPI or USB-to-I2C bridge between the host computer and the target MCU.

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.

The FRDM-MCXC444 board supports connecting MCU-Link to the SPI0 module of the target MCU. By default, this SPI connection is disabled. It can be enabled by populating the following 0 Ω resistors:

- R72
- R73
- R74
- R75

Similarly, the FRDM-MCXC444 board supports connecting MCU-Link to the I2C0 module of the target MCU. By default, this I2C connection is disabled. It can be enabled by populating the following 0 Ω resistors:

- R78
- R79

To use MCU-Link as a USB-to-SPI or USB-to-I2C bridge, the board must be connected to the host computer by connecting a USB cable from its J13 connector. A USB-to-SPI bridge can be used to emulate the host system. A USB-to-I2C bridge can be used to emulate the host system / board peripherals.

By default, the USBSIO feature is disabled for SPI and I2C on the FRDM-MCXC444 board, allowing the target MCU SPI/I2C port to be used for other purposes. Disabling the USBSIO 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 feature and energy measurement feature (not supported on this board) of MCU-Link.

3.9 MCU-Link status LEDs

The FRDM-MCXC444 board has three status indicator LEDs for MCU-Link. Table 25 lists these LEDs and describes how each LED behaves in different MCU-Link modes.

Table 25. MCU-Link LEDs

Part identifier	PCB label	LED color	LED function		
			Normal operation (with CMSIS-DAP)	Normal operation (with J-Link)	ISP (firmware update) mode
D8	USB_ ACTIVE	Green	Indicates USB communication. The LED lights up after successful USB enumeration at startup, and then stays ON.	The LED remains OFF.	The LED remains OFF.
D9	ISP_EN	Red	Indicates MCU-Link status / SWD activity. It acts as a heartbeat LED (fades in/ out repeatedly), with SWD activity overlaid. If an error occurs at startup, the D8 LED blinks rapidly.	The LED remains OFF.	The LED lights up when MCU-Link (LPC55S69) boots in ISP mode.
D10	VCOM_ ACTIVE	Green	Indicates if the VCOM port is receiving/sending data. The LED lights up when MCU-Link boots, and then blinks when debug activity happens.	Indicates if the VCOM port is receiving/sending data. The LED lights up when MCU-Link boots, and then blinks when debug activity happens.	The LED remains OFF.

4 Board errata

Not applicable for the current board revision.

5 Related documentation

Table 26 lists some additional documents and resources that you can refer to for more information on the FRDM-MCXC444 board. Some of these documents may be available only under a non-disclosure

agreement (NDA). To access such a document, contact a local NXP field applications engineer (FAE) or sales representative.

Table 26. Related documentation

Document	Description	Link / how to obtain
MCX C44X Sub-Family Reference Manual	Provides a detailed description about the MCXC444 MCU and its features, including memory maps, power supplies, and clocks.	MCXC444RM
MCX C44X Microcontroller Data Sheet	Provides information about the MCXC444 electrical characteristics, hardware design considerations, and ordering information.	MCXC44XP64M48SF6
FRDM-MCXC444 design file	A zip file including *.DSN, *.brd, schematic files, and so on.	FRDM-MCXC444-DESIGNFILES

6 Acronyms

Table 27 lists the acronyms used in this document.

Table 27. Acronyms

Acronym	Description
ADC	Analog-to-digital converter
DNP	Do not populate / do not place
I2C	Inter-integrated circuit
ISP	In-system programming
LDO	Low-dropout regulator
LED	Light-emitting diode
LLWU	Low-leakage wakeup unit
LPUART	Low-power universal asynchronous receiver/transmitter
MCU	Microcontroller unit
NMI	Non-maskable interrupt
OB	Onboard
OS	Operating system
PWM	Pulse width modulator
RAM	Random-access memory
ROM	Read-only memory
RTC	Real-time clock
RX	Receive
SLCD	Segment liquid crystal display
SPI	Serial peripheral interface
SWD	Serial wire debug
SWO	Serial wire debug trace output
TX	Transmit

Table 27. Acronyms...*continued*

Acronym	Description
UART	Universal asynchronous receiver/transmitter
USB	Universal serial bus
USBSIO	USB serial input/output
VCOM	Virtual communication

7 Revision history

Table 28 summarizes the revisions to this document.

Table 28. Revision history

Document ID	Release date	Description
UM12120 v.1	15 July 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.

Contents