

# UM12049

## FRDM-KE17Z512 Board User Manual

Rev. 1.0 — 17 April 2024

User manual

### Document information

Information	Content
Keywords	UM12049, FRDM-KE17Z512, MKE17Z512VLL9
Abstract	This document provides detailed information about the FRDM-KE17Z512 board interfaces, power supplies, clocks, LEDs, sensors, and other interfaces.



# 1 Overview

The Kinetis KE17Z512 freedom (FRDM-KE17Z512) board is a standalone development platform that supports two microcontrollers (MCUs): the target MCU and an onboard debugger MCU. The target MCU is MKE17Z512VLL9, which is a part of the Kinetis E series of Arm Cortex-M0+ MCUs product family. The onboard debugger MCU is an LPC55S6x MCU family device, LPC55S69JEV98.

The FRDM-KE17Z512 board is compatible with the Arduino shields, NXP FRDM-TOUCH board, NXP FRDM-MC-LVBLDC board, Pmod connector board, and Mikroe click board.

The FRDM-KE17Z512 board comes preloaded with the LED blinky demo. The demo is available at the `boards\frdmke17z512\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-KE17Z512 board including power supplies, clocks, LEDs, accelerometer sensor, MCU-Link debug probe circuit, and other interfaces on the board.

## 1.1 Block diagram

The [Figure 1](#) shows the FRDM-KE17Z512 block diagram.

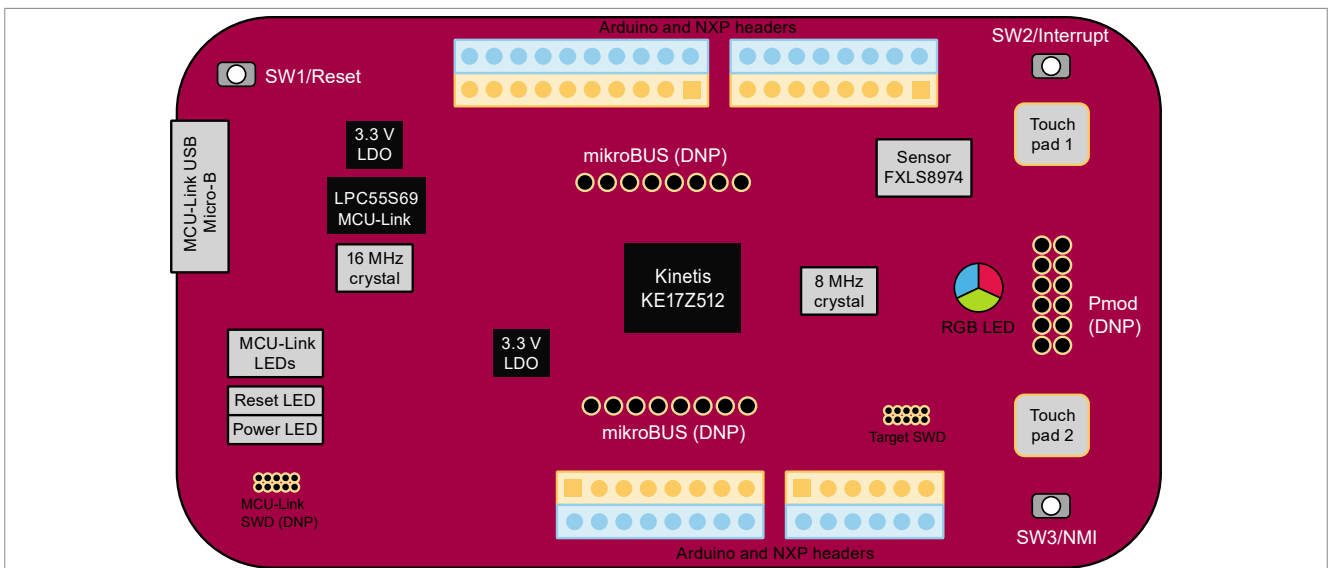


Figure 1. FRDM-KE17Z512 block diagram

## 1.2 Board features

[Table 1](#) lists the features of the FRDM-KE17Z512 board.

Table 1. FRDM-KE17Z512 features

FRDM-KE17Z512 feature	Target MCU features used	Description
MKE17Z512VLL9 MCU (Target MCU)		MKE17Z512VLL9, 32-bit MCU core from the Arm Cortex-M0+ class, supports up to 96 MHz CPU frequency, 512 KB Flash, and 96 KB RAM
Power supply		<ul style="list-style-type: none"> <li>5 V input power via an external DC power supply through MCU-Link USB micro-B connector J10</li> </ul>

Table 1. FRDM-KE17Z512 features...continued

FRDM-KE17Z512 feature	Target MCU features used	Description
		<ul style="list-style-type: none"> <li>3.3 V or 5 V for the MKE17Z512VLL9 MCU by power selection jumper J8</li> <li>3.3 V for the LPC55S69JEV98 MCU-Link</li> </ul>
Clock		<ul style="list-style-type: none"> <li>8 MHz crystal clock for the MKE17Z512VLL9 MCU</li> <li>16 MHz crystal clock for the LPC55S69JEV98 MCU-Link</li> </ul>
TSI touch pad	Touch sensing input (TSI) module	<ul style="list-style-type: none"> <li>Supports two touch electrodes to support TSI functions</li> <li>Electrode 1 is connected to TSI0 channel 19 and electrode 2 is connected to TSI1 channel 3</li> </ul>
3-axis MEMS accelerometer	Low-power inter-integrated circuit (LPI2C) module	Includes one NXP FXLS8974CFR3 device supporting motion sensing
Debug		<ul style="list-style-type: none"> <li>Onboard MCU-Link circuit provides the onboard debugger for debugging the MKE17Z512VLL9 MCU</li> <li>Supports a micro-B USB port to access serial port as console for debug</li> <li>Supports an SWD connector to use an external debugger to program and debug the MKE17Z512VLL9 MCU</li> <li>Supports an SWD connector to use an external debugger to program and debug the LPC55S69JEV98 MCU</li> </ul>
Buttons		Supports two interrupt push buttons and one MCU reset button: <ul style="list-style-type: none"> <li>SW1 is used to reset the MKE17Z512VLL9 MCU</li> <li>SW2 is used to issue an asynchronous wake-up interrupt controller (AWIC) interrupt signal to the MKE17Z512VLL9 MCU</li> <li>SW3 is used to issue a non-maskable interrupt (NMI) signal to the MKE17Z512VLL9 MCU</li> </ul>
I/O headers		Headers compatible with: <ul style="list-style-type: none"> <li>Arduino shields (outer rows) and FRDM header (inner rows)</li> <li>Mikroe click boards</li> <li>Peripheral module (Pmod)</li> </ul>
RGB LED	Pulse width modulation (PWM)	Supports RGB LED controlled by the embedded software application
PCB		8.2 cm x 5.3 cm
Orderable part number		FRDM-KE17Z512

### 1.3 Kit contents

Table 2 lists the items included in the FRDM-KE17Z512 board kit.

Table 2. Hardware kit contents

Item	Quantity
FRDM-KE17Z512 board hardware assembly	1
USB Type A to USB micro-B cable	1
FRDM-KE17Z512 Quick Start Guide	1

### 1.4 Board pictures

Figure 2 shows the top view of the FRDM-KE17Z512 board.

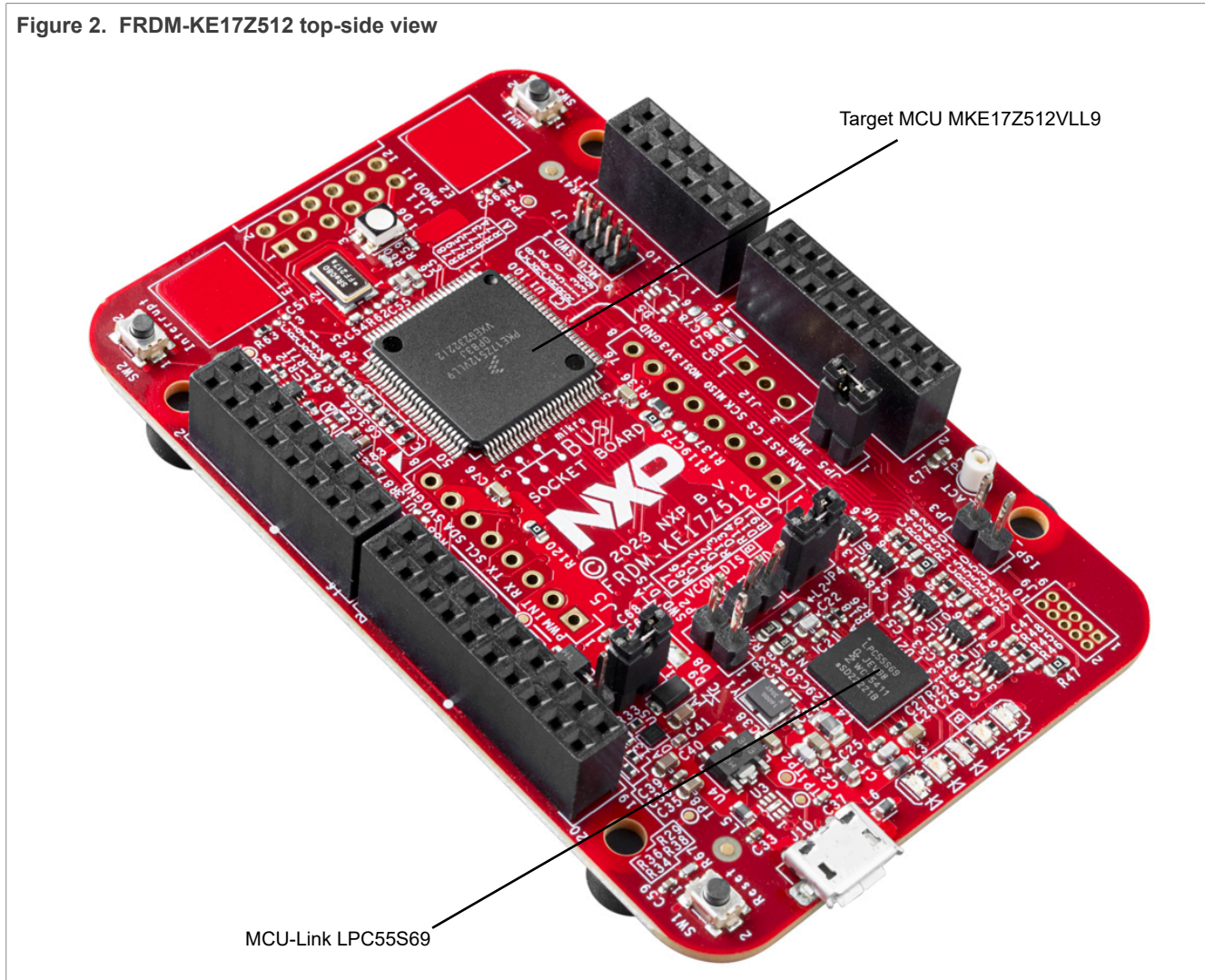


Figure 3 shows the top-side view of the FRDM-KE17Z512 board, with onboard connectors, jumpers, push buttons, and LEDs highlighted.

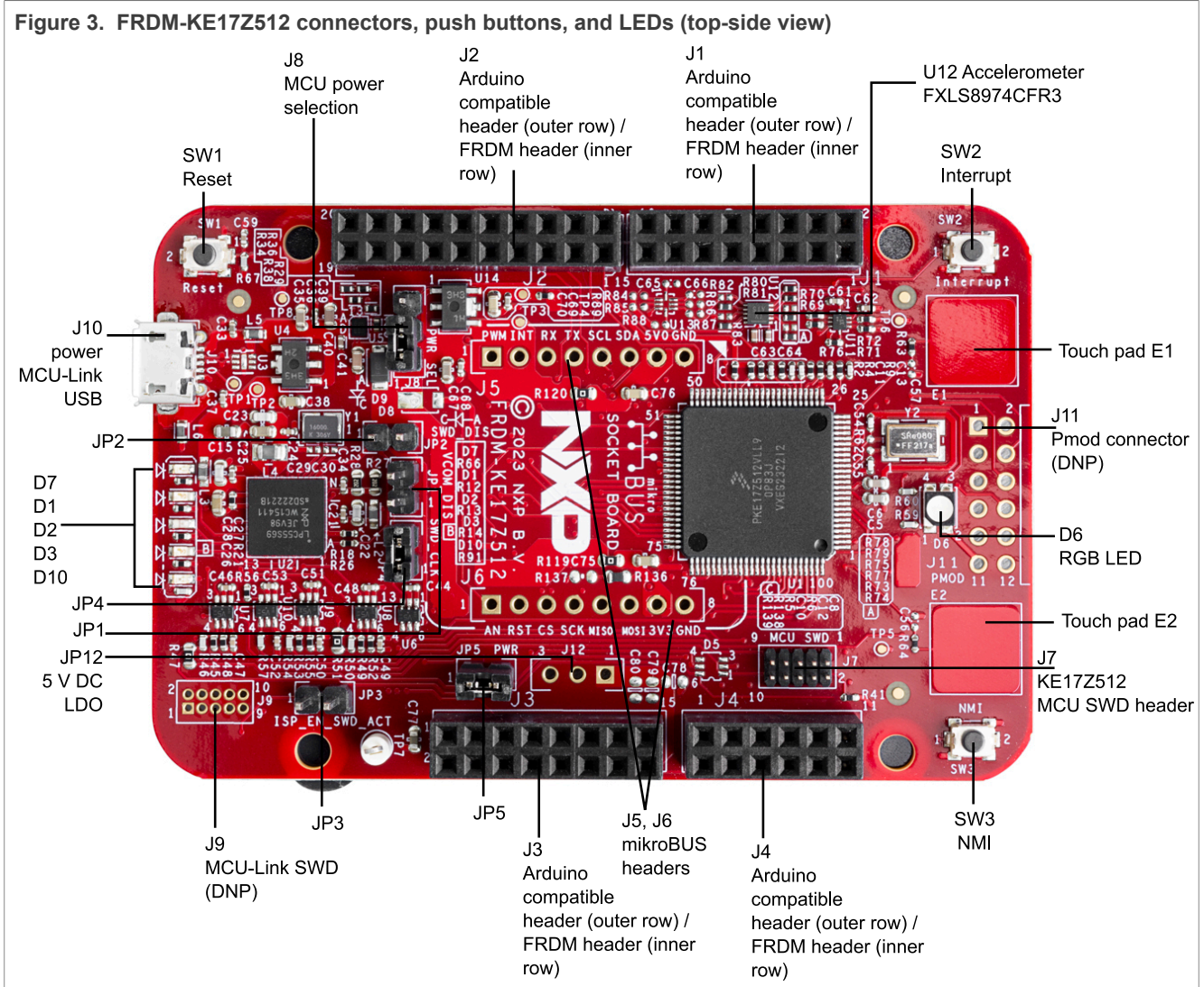


Figure 4 shows the bottom view of the FRDM-KE17Z512 board.

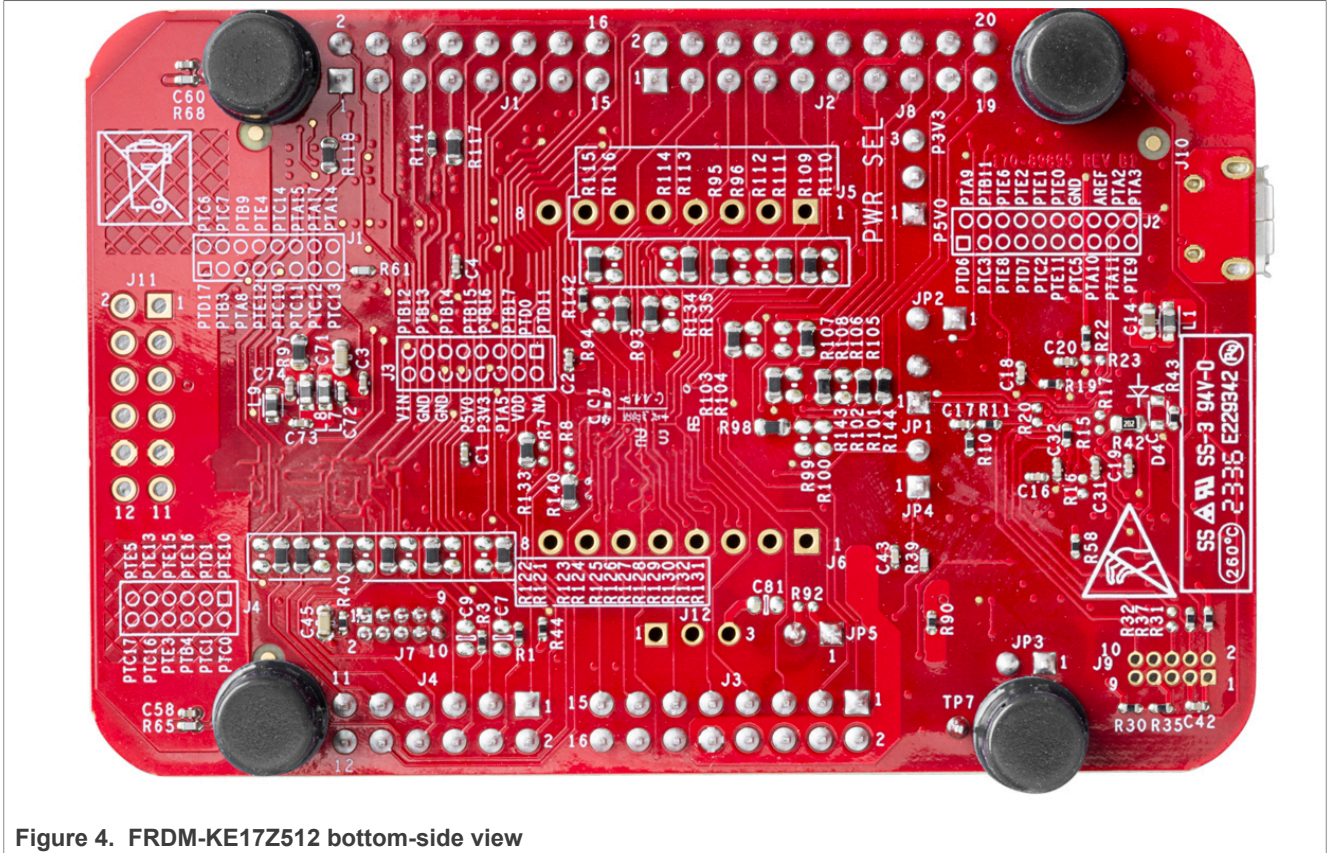


Figure 4. FRDM-KE17Z512 bottom-side view

### 1.5 Connectors

Connectors are onboard devices that allow you to connect external devices to the board. [Figure 3](#) shows the FRDM-KE17Z512 connectors. [Table 3](#) describes the connectors.

Table 3. FRDM-KE17Z512 connectors

Part identifier	Connector type	Description	Reference section
J1	2x8 connector	I/O headers compatible with the Arduino shields, the NXP FRDM-TOUCH board, and the NXP FRDM-MC-LVBLDC board	<a href="#">Section 2.5</a>
J2	2x10 connector		
J3	2x8 connector		
J4	2x6 connector		
J5	1x8 connector	mikroBUS interface	<a href="#">Section 2.6</a>
J6	1x8 connector		
J7	2x5 connector	MCU SWD interface	<a href="#">Section 3.3</a>
J9 (DNP)	2x5 connector	MCU-Link SWD interface (for development of custom debug probe code only)	For details, see FRDM-KE17Z512 schematics
J10	Micro-B USB connector	MCU-Link USB interface	<a href="#">Section 3.7</a>
J11 (DNP)	2x6 connector	Pmod interface	<a href="#">Section 2.7</a>
J12 (DNP)	1x3 header	For P5V_LDO_OUT output supply	<a href="#">Section 2.1</a>

## 1.6 Jumpers

Jumpers (or shorting headers) are small connectors that allow you to choose from two or more options available. Jumpers are usually installed during board assembly and do not require any changes. In FRDM-KE17Z512, all jumpers are 2/3-pin connectors with two settings: open and shorted. [Figure 3](#) shows the FRDM-KE17Z512 jumpers.

[Table 4](#) lists all the jumpers available on the FRDM-KE17Z512 board.

**Table 4. FRDM-KE17Z512 jumpers**

Part identifier	Jumper type	Description	Jumper settings
J8	1x3 header	MCU power select	<ul style="list-style-type: none"> <li>Pin 1-2 shorted (<b>Default setting</b>): VDD is 5 V powered.</li> <li>Pin 2-3 shorted: VDD is 3.3 V powered.</li> </ul>
JP1	1x2 header	Primary VCOM enable / disable	<ul style="list-style-type: none"> <li>Open (<b>default setting</b>): Primary MCU-Link USB to the MCU UART bridge (VCOM) is enabled.</li> <li>Shorted: Primary MCU-Link USB to the MCU UART bridge (VCOM) is disabled.</li> </ul>
JP2	1x2 header	SWD function enable/ disable	<ul style="list-style-type: none"> <li>Open (<b>default setting</b>): MCU-Link serial wire debug (SWD) feature is enabled. MCU-Link can be used to drive SWD of the target MCU.</li> <li>Shorted: MCU-Link SWD feature is disabled. This setting of JP2 can be used when connecting an external debugger through connector J7 for debugging the target MCU.</li> </ul>
JP3	1x2 header	MCU-Link ISP mode enable / disable	<ul style="list-style-type: none"> <li>Open (<b>default setting</b>): 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.</li> <li>Shorted: MCU-Link is forced to ISP mode. Use this setting to update the MCU-Link firmware.</li> </ul> <p><b>Note:</b> By default, MCU-Link flash is preprogrammed with a version of CMSIS-DAP firmware.</p>
JP4	1x2 header	SWD_CLK configuration jumper	<ul style="list-style-type: none"> <li>Open: MCU-Link SWD clock is disabled.</li> <li>Shorted (<b>default setting</b>): MCU-Link SWD clock is enabled. MCU-Link drives SWD of the target MCU MKE17Z512VLL9. The SWD clock connects to the PTC4 pin on MKE17Z512VLL9.</li> </ul>
JP5	1x2 header	MCU power enable/ disable	<ul style="list-style-type: none"> <li>Open: The MKE17Z512VLL9 MCU is powered off.</li> <li>Shorted (<b>default setting</b>): The MKE17Z512VLL9 MCU is powered on.</li> </ul>

## 1.7 Push buttons

In addition to a Reset button for manually triggering a system reset, the FRDM-KE17Z512 board supports two interrupt push buttons. [Figure 3](#) shows the FRDM-KE17Z512 push buttons.

[Table 5](#) explains the push buttons used on the FRDM-KE17Z512 board.

**Table 5. Reset and interrupt push buttons**

Part identifier	Switch type	Description
SW1	Push button	<p>When pressed, resets the MKE17Z512VLL9 MCU. SW1 is connected to the PTA5 pin on MKE17Z512VLL9.</p> <p><b>Note:</b> Apart from SW1, MCU-LINK also sends a reset signal to MKE17Z512VLL9 through the DBGIF_RESET signal, which connects to the PTA5 pin through the bidirectional voltage translator U8. An arduino shield connected to</p>

Table 5. Reset and interrupt push buttons...continued

Part identifier	Switch type	Description
		the J3 header also sends a reset signal (PTA5/Arduino_RESET) to the MKE17Z512VLL9 MCU through pin 6.
SW2	Push button	When pressed, issues an asynchronous wakeup interrupt controller (AWIC) interrupt signal to MKE17Z512VLL9. SW2 is connected to the PTE14 pin on the MKE17Z512VLL9 MCU.
SW3	Push button	When pressed, generates a non-maskable interrupt (NMI) signal to the MKE17Z512VLL9 MCU. SW3 is connected to the PTD3 pin on the MKE17Z512VLL9 MCU.

### 1.8 LEDs

FRDM-KE17Z512 has light-emitting diodes (LEDs) to monitor system functions, such as power-on, reset, and board faults. The information collected from LEDs can be used for debugging purposes. [Figure 3](#) shows the FRDM-KE17Z512 LEDs.

[Table 6](#) describes the LEDs that correspond to the target MCU.

**Note:** The FRDM-KE17Z512 board also has three status indicator LEDs for MCU-Link. For details, see [Section 3.9](#).

Table 6. FRDM-KE17Z512 LEDs

Part identifier	LED color	LED name	Description (When LED in ON)
D6	RGB	RGB LED	User application LEDs. Each of these LEDs can be controlled through a user application. <ul style="list-style-type: none"> <li>Red LED connects to target MCU pin PTD10</li> <li>Green LED connects to target MCU pin PTD5</li> <li>Blue LED connects to target MCU pin PTD12</li> </ul>
D7	Red	Reset LED	Indicates that the Reset command is sent to the target MCU via MCU-Link or when SW1 is pressed
D10	Green	Power LED	Indicates that VDD supply is On

## 2 Functional description

This section describes the features and functions of the FRDM-KE17Z512 board. For details of the MKE17Z512VLL9 MCU features, see Kinetis KE17Z/13Z/12Z with up to 512 KB Flash Reference Manual.

The section is divided into the following subsections:

- [Section "Power supplies"](#)
- [Section "Clock"](#)
- [Section "TSI"](#)
- [Section "3-axis digital sensor"](#)
- [Section "Input/output headers"](#)
- [Section "mikroBUS interface"](#)
- [Section "Pmod interface"](#)
- [Section "Board operating conditions"](#)



## 2.1 Power supplies

The FRDM-KE17Z512 board is powered with a 5 V power supply. The 5 V supply can be sourced from one of the following source options:

- P5V\_MCU\_LINK\_USB: External power supply from MCU-Link USB2.0 micro-B connector J10 (**default setting**).
- P5V\_LDO\_OUT: External power supply from the 5 V regulator output populated at header J12 (DNP by default).

**Note:** The input voltage P5-9V\_VIN of the 5 V regulator is from J3, pin16.

The onboard power supply devices use the 5 V power to generate the required power supplies for the MKE17Z512VLL9 MCU and different peripheral and interfaces on the board, such as SWD interface, I/O headers for Arduino shields, NXP FRDM-TOUCH board, and NXP FRDM-MC-LVBLDC.

The MKE17Z512VLL9 MCU can be powered by a supply voltage ranging from 2.7 V to 5.5 V. The J8 jumper on the board is provided to power the MCU with either 3.3 V or 5 V. For details of the J8 setting, see [Section 1.6](#). However, the MCU-Link debugger MCU LPC55S69JEV98 is always 3.3 V powered.

[Table 7](#) describes the power devices and generated power supplies on the board.

Table 7. Power supplies

Part identifier	Device / power source	Output power supply	Description
J10	MCU-Link USB2.0 micro-B connector	P5V_MCU_LINK_USB	<ul style="list-style-type: none"> <li>• Input for MCU-LINK USB power switch U5, providing P5V_MCU_LINK_PSW</li> <li>• Input for MCU-Link voltage regulator U4</li> <li>• Input for USB1_VBUS of MCU-Link microcontroller LPC55S69</li> </ul>
J12	5 V power regulator populated at J12	P5V_LDO_OUT	One of the sources for P5V0 supply
-	P5V_MCU_LINK_PSW ( <b>default setting</b> ) / P5V_LDO_OUT	P5V0	<ul style="list-style-type: none"> <li>• Input power supply for XC6227C331PR-G LDO voltage regulator (U14)</li> <li>• Input power supply for mikroBUS connector J5 (DNP)</li> <li>• One of the sources (default option) for VDD_KE17Z512</li> </ul>
U4	XC6227C331PR-G LDO	3V3	Power supply for: <ul style="list-style-type: none"> <li>• MCU-Link LEDs (D1, D2, and D3)</li> <li>• MCU-Link microcontroller LPC55S69</li> <li>• MCU-Link SWD connector J9</li> </ul>
U14	XC6227C331PR-G (TOREX)	P3V3	<ul style="list-style-type: none"> <li>• Power supply for mikroBUS connector J6 (DNP)</li> <li>• Power supply for accelerometer sensor U12</li> <li>• One of the sources for VDD_KE17Z512</li> </ul>
	P3V3 or P5V0 through J8	VDD	<ul style="list-style-type: none"> <li>• Power supply for:                             <ul style="list-style-type: none"> <li>– Pmod connector J11 (DNP)</li> <li>– MCU SWD connector J7</li> <li>– RGB LED D6</li> </ul> </li> <li>• Power source for:                             <ul style="list-style-type: none"> <li>– VCC_IF_LINK supply</li> <li>– VDD_KE17Z512 supply</li> </ul> </li> </ul>
L7	VDD_KE17Z512	VDDA	<ul style="list-style-type: none"> <li>• Power supply for MCU KE17Z512 analog circuits</li> <li>• Power source for VREFH</li> </ul>
L8	VDDA	VREFH	<ul style="list-style-type: none"> <li>• Voltage reference for MCU KE17Z512 analog modules</li> </ul>

Table 7. Power supplies...continued

Part identifier	Device / power source	Output power supply	Description
			• Power source for AREF
R97	VREFH	AREF	Power supply for I/O header J2, pin 16

## 2.2 Clock

FRDM-KE17Z512 has the capability to run up to 96 MHz.

[Table 8](#) provides details of different clocks of FRDM-KE17Z512.

Table 8. FRDM-KE17Z512Z clocks

Clock generator	Clock frequency	Destination	Description
Crystal oscillator, Y1 (CX3225 GA16000 D0PTVCC, KYOCERA)	16 MHz	XTAL32M_N/P pins of LPC55S69 MCU-Link	Option for external clock input to MCU-Link
Crystal oscillator, Y2 (FY0800027, Diodes Incorporated)	8 MHz	<ul style="list-style-type: none"> <li>PTB6 pin of target MCU MKE17Z512 VLL9</li> <li>PTB7 pin of target MCU MKE17Z512 VLL9</li> </ul>	Option for external clock input to MKE17Z512VLL9 MCU

## 2.3 TSI

MKE17Z512VLL9 includes the Touch Sensing Input (TSI) module to detect capacitive touch sensor. TSI is an electrode capacitive scan method based on the hardware. The basic element in capacitive touch sensing is the electrode.

In this case, the electrode is an area of conductive material with dielectric material on the top, usually plastic or glass. This is what the user touches. This conductive area plus the dielectric material effectively create a capacitor referenced to the system ground.

By touching the dielectric on top of the electrode, the user effectively changes the electrode capacitance both by adding a second conductive area that is grounded and by increasing the dielectric of the original capacitor. The sensor, or the TSI module in this case, uses a capacitive sensing method to measure changes in the electrode capacitance.

[FRDM-TOUCH](#) is a touch shield board for FRDM-KE17Z512 with capacitive touch buttons, slider, and rotary. This board can connect with FRDM-KE17Z512 and control the onboard RGB LED using keys, slider, and rotary touches.

FRDM-KE17Z512 includes two touch electrodes to support TSI functions in self-capacitive mode. See [Table 9](#).

The touch electrode 1 is connected to TSI0 channel 19 and touch electrode 2 is connected to TSI1 channel 3.

Table 9. TSI touch pad connections

Touch electrode	Touch pad	Pin / TSI channel
Electrode 1	TouchPad_0	PTD16/TSI0_CH19

Table 9. TSI touch pad connections...continued

Touch electrode	Touch pad	Pin / TSI channel
Electrode 2	TouchPad_1	PTA1/TSI1_CH3

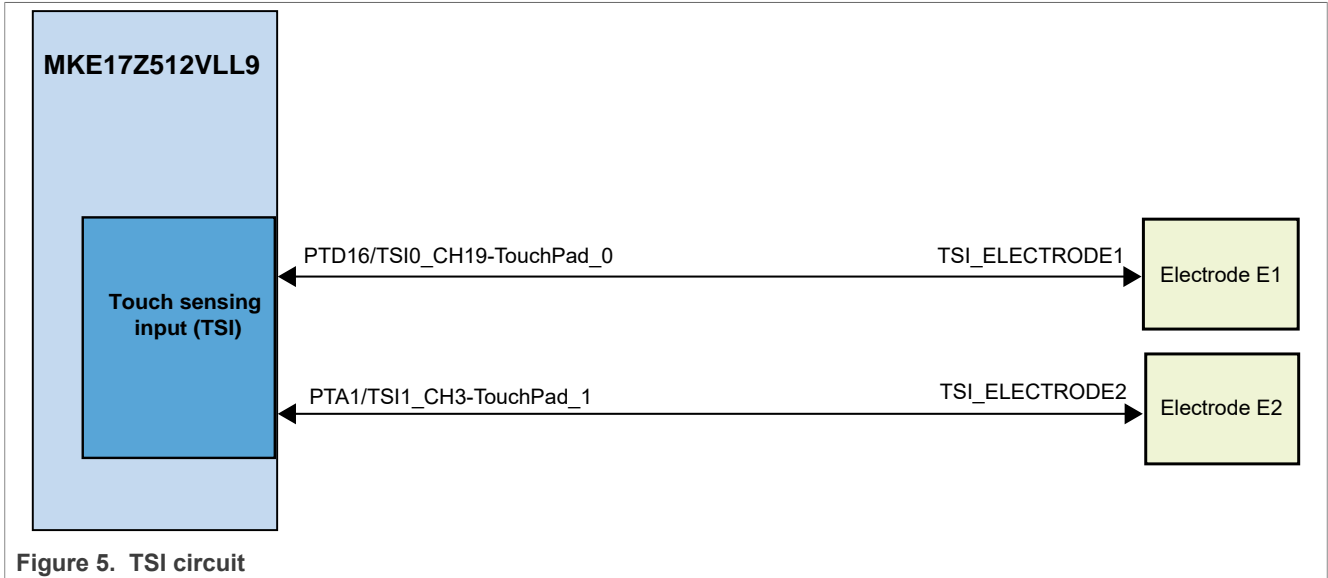


Figure 5. TSI circuit

FRDM-KE17Z512 outputs TSI signals to headers J2 and J4 to support the FRDM-TOUCH board. [FRDM-TOUCH](#) includes four touch keys in a mutual-capacitive mode, touch slider, and rotary.

Table 10. TSI channel connections

Pin / TSI channel	I/O header	FRDM-TOUCH connection
PTB5/TSI0_CH23 (DNP)	J2[3]	-
PTE4/TSI0_CH17 (DNP)	J2[5]	-
PTA12/TSI0_CH3 (DNP)	J2[7]	-
PTA13/TSI0_CH2 (DNP)	J2[9]	-
PTE11/TSI0_CH9	J2[11]	TOUCH_M_RX_0
PTD4/TSI1_CH7 (DNP)		
PTC5/TSI0_CH8	J2[13]	TOUCH_M_RX_1
PTD2/TSI1_CH6 (DNP)		
PTA10/TSI0_CH5	J2[15]	TOUCH_M_TX_0
PTA3/TSI1_CH5 (DNP)		
PTA11/TSI0_CH4	J2[17]	TOUCH_M_TX_1
PTA2/TSI1_CH4 (DNP)		
PTE9/TSI0_CH21	J2[19]	-
PTA7/TSI1_CH12 (DNP)		
GND (DNP)		
PTE10/TSI0_CH10	J4[1]	TOUCH_S_0
PTE7/TSI1_CH10 (DNP)		
PTD1/TSI0_CH11	J4[3]	TOUCH_S_1

Table 10. TSI channel connections...continued

Pin / TSI channel	I/O header	FRDM-TOUCH connection
PTA6/TSI1_CH11 (DNP)		
PTE16/TSI0_CH13	J4[5]	TOUCH_S_2
PTC8/TSI1_CH13 (DNP)		
PTE15/TSI0_CH14	J4[7]	TOUCH_S_3
PTC9/TSI1_CH14 (DNP)		
PTE13/TSI0_CH15	J4[9]	TOUCH_S_4
PTB0/TSI1_CH15 (DNP)		
PTE5/TSI0_CH16	J4[11]	TOUCH_S_5
PTB1/TSI1_CH16 (DNP)		
PTC1/ADC0_SE8/TSI1_CH24	J4[4]	-
PTB4/ADC0_SE5/TSI0_CH24	J4[6]	-

## 2.4 3-axis digital sensor

The FRDM-KE17Z512 board supports an accelerometer sensor 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 (U12) is used.
- FXLS8974CFR3 is connected to MKE17Z512VLL9 through an I2C interface, LPI2C0.
- 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
  - SA0:1 → 8-bit I2C read address: 0x33, 8-bit I2C write address: 0x32
- Voltage level translators (U11, U13 (DNP)) are connected between MKE17Z512VLL9 and FXLS8974CFR3. These translators translate 5 V (default) I2C signals from MCU to 3.3 V I2C signals for FXLS8974CFR3 and vice versa.
- Series zero ohm resistors (R78 and R79) are provided to isolate the sensor from the MKE17Z512VLL9 device.

[Figure 6](#) shows the FXLS8974CFR3 sensor circuit diagram.

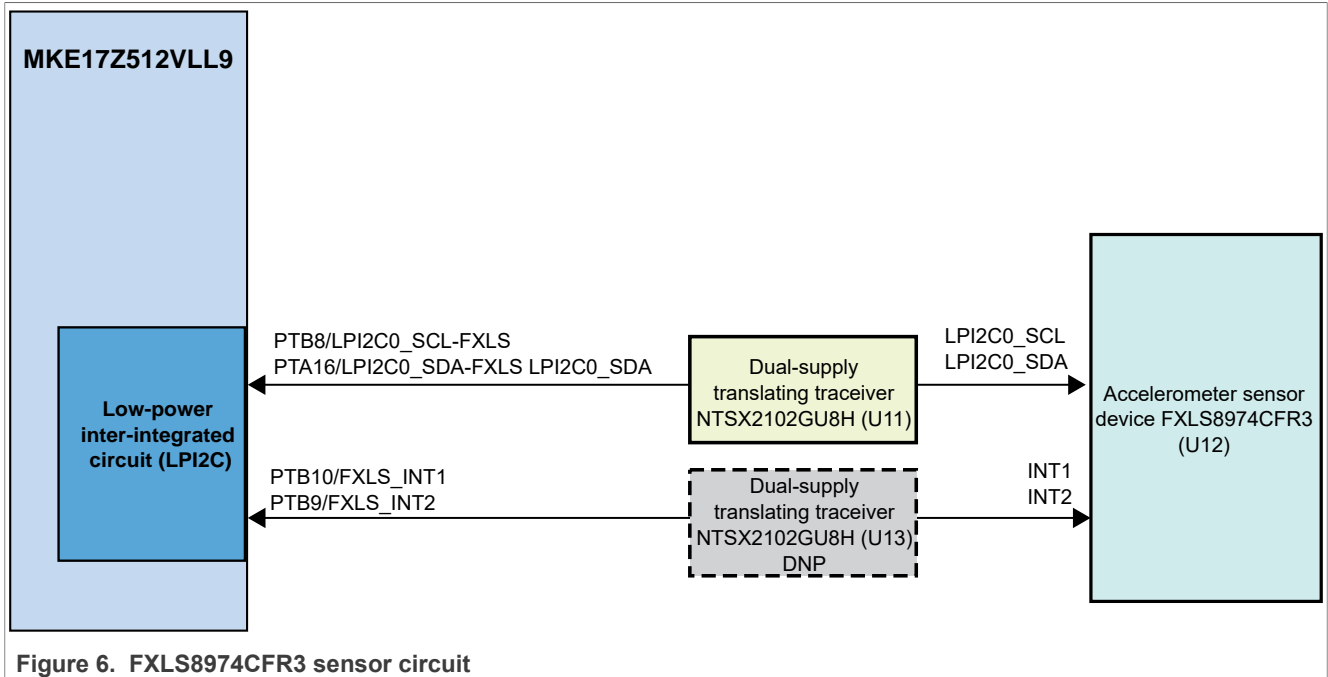


Figure 6. FXLS8974CFR3 sensor circuit

### 2.5 Input/output headers

FRDM-KE17Z512 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:

- Touch sensor: [NXP FRDM-TOUCH board](#)
- Motor control: [NXP FRDM-MC-LVBLDC board](#)
- Arduino shields

[Table 11](#) describes the connectors of the Arduino socket.

Table 11. Arduino socket connectors

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

The [Table 12](#), [Table 13](#), [Table 14](#), and [Table 15](#) explain the I/O headers pinout on FRDM-KE17Z512.

Table 12. J1 connector (at right-upper side) pinout

I/O header pin	MKE17Z512VLL9 pin / function	Arduino / FRDM functions	Potential conflict
J1[1]	PTD17/LPUART2_RX_TGTMCU	-	-
J1[2]	PTC6/LPUART1_RX	Arduino_D0	-
J1[3]	PTB3/FTM1_CH1	MC_ENC_I	-
J1[4]	PTC7/LPUART1_TX	Arduino_D1	-

Table 12. J1 connector (at right-upper side) pinout...continued

I/O header pin	MKE17Z512VLL9 pin / function	Arduino / FRDM functions	Potential conflict
J1[5]	PTA8/FXIO_D6	-	mikroBUS header J5 pin 2 (PTA8/GPIO-MIKROE_INT)
J1[6]	PTB9	Arduino_D2	Accelerometer sensor voltage translator U13 (PTB9/FXLS_INT2)
J1[7]	PTE12/LPUART2_TX_TGTMCU	-	-
J1[8]	PTE4/FTM2_CH2	Arduino_D3	-
J1[9]	PTC10/LPSPI0_SCK	-	-
J1[10]	PTC14	Arduino_D4	Pmod connector J11 pin 4 (PTC14/GPIO-PMOD_RST)
J1[11]	PTC11/LPSPI0_SIN	-	-
J1[12]	PTA15/FTM1_CH2	Arduino_D5	-
J1[13]	PTC12/LPSPI0_SOUT	-	-
J1[14]	PTA17/FTM0_CH6	Arduino_D6	-
J1[15]	PTC13/LPSPI0_PCS0	-	-
J1[16]	PTA14	Arduino_D7	-

Table 13. J2 connector (at left-upper side) pinout

I/O header pin	MKE17Z512VLL9 pin / function	Arduino / FRDM functions	Potential conflict
J2[1]	PTD6/ADC0_SE11	MC_BEMF_A	-
J2[2]	PTA9	Arduino_D8	mikroBUS header J6 pin 2 (PTA9/GPIO-MIKROE_RST)
J2[3]	PTC3/ADC0_SE7	MC_BEMF_B	-
	PTB5/TSI0_CH23 (DNP)		mikroBUS header J6 pin 3 (PTB5/LPSPI0_PCS1-MIKROE)
J2[4]	PTB11/FTM0_CH1	Arduino_D9	-
J2[5]	PTE8/ADC0_SE1	MC_BEMF_C	-
	PTE4/TSI0_CH17 (DNP)		-
J2[6]	PTE6/LPSPI0_PCS2	Arduino_D10	-
	PTD15/FTM0_CH0 (DNP)		-
J2[7]	PTD7/ADC0_SE13	MC_VOLT_DCB/GES_R0	-
	PTA12/TSI0_CH3 (DNP)		-
J2[8]	PTE2/LPSPI0_SOUT	Arduino_D11	-
	PTB2/FTM1_CH0 (DNP)		-
J2[9]	PTC2/ADC0_SE15	MC_CUR_DCB / GES_R1	-
	PTA13/TSI0_CH2 (DNP)		-

Table 13. J2 connector (at left-upper side) pinout...continued

I/O header pin	MKE17Z512VLL9 pin / function	Arduino / FRDM functions	Potential conflict
J2[10]	PTE1/LPSPI0_SIN	Arduino_D12	-
J2[11]	PTE11/TSI0_CH9	TOUCH_M_RX_0/GES_R2	-
	PTD4/TSI1_CH7 (DNP)		-
J2[12]	PTE0/LPSPI0_SCK	Arduino_D13	-
J2[13]	PTC5/TSI0_CH8	TOUCH_M_RX_1/GES_R3	-
	PTD2/TSI1_CH6 (DNP)		-
J2[14]	GND	-	-
J2[15]	PTA10/TSI0_CH5	TOUCH_M_TX_0/GES_R4	-
	PTA3/TSI1_CH5 (DNP)		J2 connector pin 20 (PTA3/LPI2C0_SCL-Arduino_D15)
J2[16]	AREF	-	-
J2[17]	PTA11/TSI0_CH4	TOUCH_M_TX_1/GES_R5	-
	PTA2/TSI1_CH4 (DNP)		J2 connector pin 18 (PTA2/LPI2C0_SDA-Arduino_D14)
J2[18]	PTA2/LPI2C0_SDA	Arduino_D14	J2 connector pin 17 (PTA2/TSI1_CH4 (DNP))
J2[19]	PTE9/TSI0_CH21	GND/GES_SHIELD	-
	PTA7/TSI1_CH12 (DNP)		-
	GND (DNP)		-
J2[20]	PTA3/LPI2C0_SCL	Arduino_D15	J2 connector pin 15 (PTA3/TSI1_CH5 (DNP))

Table 14. J3 connector (at left-lower side) pinout

I/O header pin	MKE17Z512VLL9 pin / function	Arduino / FRDM functions	Potential conflict
J3[1]	PTD11/FTM2_CH1	MC_ENC_B	-
J3[2]	Not connected	-	-
J3[3]	PTD0/FTM2_CH0	MC_ENC_A	-
J3[4]	VDD	-	-
J3[5]	PTB17/FTM0_CH5	MC_PWM_CB	-
J3[6]	PTA5	Arduino_RESET	-
J3[7]	PTB16/FTM0_CH4	MC_PWM_CT	-
J3[8]	P3V3	-	-
J3[9]	PTB15/FTM0_CH3	MC_PWM_BB	-
J3[10]	P5V0	-	-
J3[11]	PTB14/FTM0_CH2	MC_PWM_BT	-
J3[12]	GND	-	-

Table 14. J3 connector (at left-lower side) pinout...continued

I/O header pin	MKE17Z512VLL9 pin / function	Arduino / FRDM functions	Potential conflict
J3[13]	PTB13/FTM0_CH1	MC_PWM_AB	-
J3[14]	GND	-	-
J3[15]	PTB12/FTM0_CH0	MC_PWM_AT	-
J3[16]	P5-9V_VIN	-	-

Table 15. J4 connector (at right-lower side) pinout

I/O header pin	MKE17Z512VLL9 pin / function	Arduino / FRDM functions	Potential conflict
J4[1]	PTE10/TSI0_CH10	TOUCH_S_0/GES_C5	-
	PTE7/TSI1_CH10 (DNP)		-
J4[2]	PTC0/ADC0_SE10	Arduino_A0	-
J4[3]	PTD1/TSI0_CH11	TOUCH_S_1/GES_C4	-
	PTA6/TSI1_CH11 (DNP)		-
J4[4]	PTC1/ADC0_SE8	Arduino_A1	-
J4[5]	PTE16/TSI0_CH13	TOUCH_S_2/GES_C3	-
	PTC8/TSI1_CH13 (DNP)		-
J4[6]	PTB4/ADC0_SE5	Arduino_A2	-
J4[7]	PTE15/TSI0_CH14	TOUCH_S_3/GES_C2	-
	PTC9/TSI1_CH14 (DNP)		-
J4[8]	PTE3/ADC0_SE6	Arduino_A3	-
J4[9]	PTE13/TSI0_CH15	TOUCH_S_4/GES_C1	-
	PTB0/TSI1_CH15 (DNP)		-
J4[10]	PTC16/ADC0_SE14	Arduino_A4	-
	PTD8/LPI2C1_SDA (DNP)		-
J4[11]	PTE5/TSI0_CH16	TOUCH_S_5/GES_C0	-
	PTB1/TSI1_CH16 (DNP)		-
J4[12]	PTC17/ADC0_SE12	Arduino_A5	-
	PTD9/LPI2C1_SCL (DNP)		-

## 2.6 mikroBUS interface

FRDM-KE17Z512 supports mikroBUS interface to communicate with the mikroBUS socket board. [Table 16](#) explains the pinout of the mikroBUS headers J5 and J6 on FRDM-KE17Z512.

**Note:** In case the mikroBUS socket board is a 3.3 V device, ensure to switch the power supply of MKE17Z512VLL9 to 3.3 V using J8 jumper. For J8 details, see [Section 1.6](#).



Table 16. mikroBUS headers pinout

I/O header pin	MKE17Z512VLL9 pin / function	mikroBUS function	Potential conflict
J5 header pinout			
J5[1]	PTA0/FTM2_CH1	PWM	-
J5[2]	PTA8/GPIO	INT	J1 connector pin 5 (PTA8/FXIO_D6)
J5[3]	PTD13/LPUART1_RX	RX	-
J5[4]	PTD13/LPUART1_TX	TX	-
J5[5]	PTD9/LPI2C1_SCL	SCL	-
J5[6]	PTD8/LPI2C1_SDA	SDA	-
J5[7]	P5V0	5V0	-
J5[8]	GND	GND	-
J6 header pinout			
J6[1]	PTC15/ADC0_SE16	AN	-
J6[2]	PTA9/GPIO	RST	J2 connector pin 2 (PTA9/Arduino_D8)
J6[3]	PTB5/LPSPI0_PCS1	CS	J2 connector pin 3 (PTB5/TSIO_CH23 (DNP))
J6[4]	PTE0/LPSPI0_SCK	SCK	-
J6[5]	PTE1/LPSPI0_SIN	MISO	-
J6[6]	PTE2/LPSPI0_SOUT	MOSI	-
J6[7]	P3V3	3V3	-
J6[8]	GND	GND	-

## 2.7 Pmod interface

FRDM-KE17Z512 supports a Pmod interface to communicate with the Pmod connector board.

### Note:

If the Pmod connector board is a 3.3 V device, ensure to switch the power supply of the KE17Z512 MCU to 3.3 V using the J8 jumper. For J8 details, see [Section 1.6](#).

[Table 17](#) explains the pinouts for the Pmod interface on the FRDM-KE17Z512 board.

Table 17. J11 Pmod interface

I/O header pin	MKE17Z512VLL9 pin / function	Pmod function	Potential conflict
J11[1]	PTE6/LPSPI0_PCS2	SS	-
J11[3]	PTE2/LPSPI0_SOUT	MOSI	-
J11[5]	PTE1/LPSPI0_SIN	MISO	-
J11[7]	PTE0/LPSPI0_SCK	SCK	-
J11[9]	GND	GND	-
J11[11]	VDD	3.3V	-

Table 17. J11 Pmod interface...continued

I/O header pin	MKE17Z512VLL9 pin / function	Pmod function	Potential conflict
J11[2]	PTB10/GPIO	IO1	Accelerometer sensor voltage translator U13 (PTB10/FXLS_INT1)
J11[4]	PTC14/GPIO	IO2	J1 connector pin 10 (PTC14/Arduino_D4)
J11[6]	PTD9/LPI2C1_SCL	SCL	-
J11[8]	PTD8/LPI2C1_SDA	SDA	-
J11[10]	GND	GND	-
J11[12]	VDD	VDD	-

### 2.8 Board operating conditions

The operating temperature range for the FRDM-KE17Z512 board is -40 °C to +105 °C. The MKE17Z512VLL9 device supports up to 125 °C junction temperature. See *Kinetis KE17Z/13Z/12Z with up to 512 KB Flash* 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-KE17Z512. The onboard implementations of MCU-Link are referred to as MCU-Link OB.

The FRDM-KE17Z512 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-KE17Z512 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 18](#) summarizes the MCU-Link features supported on the FRDM-KE17Z512 board.

Table 18. 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 18. 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
External debug probe support	Allows debugging the target MCU (KE17Z512) 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 feature can be used.
External target support <sup>[1]</sup>	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-KE17Z512 board, the MCU-Link debug probe target can be either the MKE17Z512VLL9 MCU or an external target compliant with MCU-Link. The board also allows to use an external debugger for debugging the MKE17Z512VLL9 MCU, in place of the MCU-Link debug probe.

[Table 19](#) describes the debug scenarios supported on the FRDM-KE17Z512 board.

**Table 19. Supported debug scenarios**

Debug scenario	Feature support	Resistor / Jumper / connector settings
Use MCU-Link as a debugger for the target MCU (KE17Z512)	SWD is enabled	JP2 must be open
	VCOM is enabled	JP1 must be open
Use an external debugger to debug the target MCU (KE17Z512)	SWD is disabled	<ul style="list-style-type: none"> <li>JP2 must be shorted</li> <li>Connect an external debugger to the MCU SWD connector J7</li> </ul>
	VCOM is enabled	JP1 must be open
Use MCU-Link as a debugger for an external target MCU	SWD is enabled	<ul style="list-style-type: none"> <li>JP2 must be open</li> <li>JP4 must be open</li> <li>Connect an external target MCU to the MCU SWD connector J7</li> </ul>
	VCOM is not supported	JP1 must be shorted

### 3.4 MCU-Link host driver and utility installation

The MCU-Link 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 [Section 3.5](#)) 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 20](#) (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 20. 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 JP3, and reconnect the board. The D2 (Red) MCU-Link status LED lights up and stays on. For the D2 LED details, see [Section 3.9](#).
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 JP3, and reconnect the board. The board enumerates on the host computer as a WinUSB or HID device (depending on the firmware version (see [Table 20](#))).

**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 20](#) 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 LinksServer 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-KE17Z512 board has a Micro-B USB 2.0 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-KE17Z512 board, MCU-Link LPC55S69 is connected to the PTE12 and PTD17 pins of the target MCU through the bidirectional level translators U9 and U10, respectively.

To use MCU-Link as a USB-to-UART bridge, ensure that the JP1 jumper is open and connect the J10 connector on the board to the USB port of the host computer.

When you boot the FRDM-KE17Z512 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 MCU-Link status LEDs

The FRDM-KE17Z512 board has three status indicator LEDs for MCU-Link. [Table 21](#) lists these LEDs and describes how each LED behaves in different MCU-Link modes. These LEDs are shown in [Figure 3](#).

Table 21. 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
D1	USB Communication / Green	Lights up after successful USB enumeration at startup. Afterward, the LED stays ON.	Remains OFF	Remains OFF
D2	Status / SWD Activity / 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
D3	VCOM Activity / 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 22 lists and explains the additional documents and resources that you can refer to for more information on FRDM-KE17Z512. Some of the documents listed below are available only under a nondisclosure agreement (NDA). To request access to these documents, contact your local field applications engineer (FAE) or sales representative.

Table 22. Related documentation

Document Title	Description	How to access
Kinetis KE17Z/13Z/12Z with up to 512 KB Flash Reference Manual	Intended for system software and hardware developers and applications programmers who want to develop products with this device.	<a href="#">KE1XZP100M96SF0RM</a>
Kinetis KE17Z/13Z/12Z with up to 512 KB Flash	Provides information about electrical characteristics, hardware design considerations, and ordering information	<a href="#">KE1xZP100M96SF0</a>
MCUXpresso Software Development Kit (SDK)	MCUXpresso Software Development Kit (SDK) is a comprehensive software enablement package designed to simplify and accelerate application development with NXP MCUs based on Arm Cortex-M cores.	<a href="#">MCUXpresso Software Development Kit (SDK)</a>

Table 22. Related documentation...continued

Document Title	Description	How to access
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	<a href="#">UM11126.pdf</a>

## 6 Acronyms

[Table 23](#) lists and defines the acronyms and abbreviations used in this document.

Table 23. Acronyms

Term	Description
ADC	Analog-to-digital converter
AWIC	Asynchronous wake-up interrupt controller
CLK	Clock
DIO	Data input/output
DNP	Do not populate
ISP	In-system programming
LPI2C	Low-power inter-integrated circuit (I2C)
MCU	Microcontroller unit
MCU-Link	The NXP single, unified debug probe architecture for all NXP general-purpose Arm Cortex-M based MCUs
MISO	Master input slave output
MOSI	Master output slave input
NMI	Non-maskable interrupt
PWM	Pulse width modulation
SS	Slave select
SWD	Serial wire debug
TSI	Touch sensing input
UART	Universal asynchronous receiver transmitter
USB	Universal serial bus

## 7 Revision history

[Table 24](#) summarizes the revisions to this document.

**Table 24. Revision history**

Document ID	Release date	Description
UM12049 v1.0	17 April 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](mailto: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,  $\mu$ Vision, 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.

**eIQ** — is a trademark of NXP B.V.

**IAR** — is a trademark of IAR Systems AB.

**i.MX** — is a trademark of NXP B.V.

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

**Kinetis** — is a trademark of NXP B.V.

## Contents

<b>1</b>	<b>Overview</b> .....	<b>2</b>
1.1	Block diagram .....	2
1.2	Board features .....	2
1.3	Kit contents .....	3
1.4	Board pictures .....	4
1.5	Connectors .....	6
1.6	Jumpers .....	7
1.7	Push buttons .....	7
1.8	LEDs .....	8
<b>2</b>	<b>Functional description</b> .....	<b>8</b>
2.1	Power supplies .....	9
2.2	Clock .....	10
2.3	TSI .....	10
2.4	3-axis digital sensor .....	12
2.5	Input/output headers .....	13
2.6	mikroBUS interface .....	16
2.7	Pmod interface .....	17
2.8	Board operating conditions .....	18
<b>3</b>	<b>MCU-Link OB debug probe</b> .....	<b>18</b>
3.1	MCU-Link overview .....	18
3.2	Supported MCU-Link features .....	18
3.3	Supported debug scenarios .....	19
3.4	MCU-Link host driver and utility installation .....	19
3.5	Updating MCU-Link firmware using the firmware utility .....	20
3.6	Using MCU-Link with development tools .....	20
3.6.1	Using MCU-Link with MCUXpresso IDE .....	21
3.6.2	Using MCU-Link with MCUXpresso for Visual Studio Code .....	21
3.6.3	Using MCU-Link with third-party IDEs .....	21
3.7	MCU-Link USB connector .....	21
3.8	Connecting to a target through a USB-to- UART bridge .....	21
3.9	MCU-Link status LEDs .....	21
<b>4</b>	<b>Board errata</b> .....	<b>22</b>
<b>5</b>	<b>Related documentation</b> .....	<b>22</b>
<b>6</b>	<b>Acronyms</b> .....	<b>23</b>
<b>7</b>	<b>Revision history</b> .....	<b>24</b>
	<b>Legal information</b> .....	<b>25</b>

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