AN14209

How To Perform Boundary Scan On MCXA Series Using µTrace And Trace32

Rev. 1 — 12 April 2024 Application note

Document information

Information	Content
Keywords	MCXA, boundary scan, µTrace, trace32
Abstract	This document focuses on how to enter boundary scan mode and perform boundary scan on MCXA series and gives an overview of JTAG and the boundary scan technology.

1 Introduction

The MCXA series is a Cortex-M33-based microcontroller with multiple high-speed connectivity, operating up to 96 MHz, serial peripherals, timers, analog, and low power consumption. The JTAG boundary scan is supported. This document focuses on how to enter boundary scan mode and perform boundary scan on the MCXA series and gives an overview of JTAG and the boundary scan technology.

This application note applies to MCXA series MCUs but is also applicable to other MCX series like MCXN.

To better understand this document, basic knowledge of JTAG and boundary scan is required.

2 JTAG and boundary scan

This section provides general information on JTAG and boundary scan.

2.1 Introduction

JTAG/boundary scan is an interface containing four ports that allows access to the special embedded logic on most chips. The JTAG/boundary can provide several functions that can contain any or all the following:

- probe-less device connectivity test;
- logic programming for flash memory, CPLD's and FPGA's;
- debug logic in microprocessors and microcontrollers used for software debugging or testing connections with peripheral devices at speed without embedded software.

2.2 Development history

The architecture for the Test Access Port (TAP) and the boundary scan is defined in IEEE Std 1149.1. The development history of this standard is summarized as follows:

- 1985: The Joint European Test Action Group (**JETAG**) was formed.
- 1986: The Joint European Test Action Group (**JETAG**) was renamed as Joint Test Action.

Group (**JTAG**).

- 1986-1988: JTAG Technical Subcommittee developed and published a series of proposals for a standardized form of the boundary scan.
- 1988: The last of these proposals, JTAG Version 2.0, was offered to the IEEE Testability.

Bus Standards Committee (P1149) and was accepted by P1149. The JTAG proposal became the basis of the standard within the Testability Bus family.

- 1990: From 1990, JTAG developed a supplement for correction, clarification, and enhancement.
- 1993 IEEE Std 1149.1aTM-1993
- 1994 IEEE Std 1149.1b-1994
- 2001 IEEE Std 1149.1-2001
- 2013 IEEE Std 1149.1-2013

2.3 Basic principle

The boundary scan is a method for testing interconnects on PCBs and internal IC subblocks. For boundary scan tests, additional logic is added to the device. The boundary scan cells are placed between the core logic and the ports.

In the boundary scan test, each primary input and output signal on a device is supplemented with a multipurpose memory element called a boundary scan cell. These cells are connected to a shift register, which is referred to as the boundary scan register. This register can be used to read and write port states.

In normal mode, these cells are transparent, and the core is connected to those ports. In boundary scan mode, the core is isolated from the ports and the port signals are controlled by the JTAG interface.

The basic principle of boundary scan is shown in the figure below.

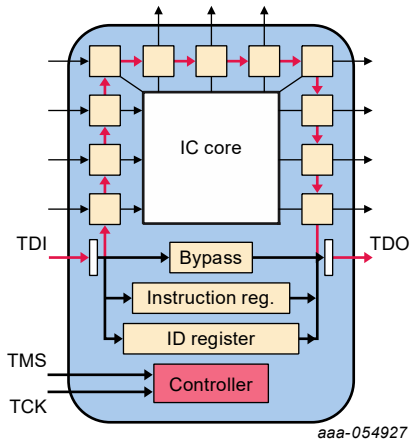

Figure 1. Boundary scan basic principle

2.4 Instruction set

Boundary scan instructions defined in the IEEE Std 1149.1 are described in Table 1

Table 1. Standard instruction set

Instruction	Mandatory/Optional	Description
BYPASS	Mandatory	TDI is connected to TDO via a single shift register.
SAMPLE	Mandatory	takes a snapshot of the normal operation of the IC (Integrated Circuit)
PRELOAD	Mandatory	loads data to the boundary scan register
EXTEST	Mandatory	apply preloaded data of the boundary scan register to the ports
INTEST	Optional	apply preloaded data of the boundary scan register to the core logics
RUNBIST	Optional	executes a self-contained self-test of the IC
CLAMP	Optional	applies the preloaded data of the boundary scan register to the ports and selects the bypass register as the serial path between TDI and TDO
IDCODE	Optional	reads the device identification register
USERCODE	Optional	reads and writes a user programmable identification register
HIGHZ	Optional	places the IC in an inactive drive state (for example, all ports are set to the high-impedance state)

2.5 JTAG Test Access Port (TAP)

TAP is a general-purpose port that can provide access to many test support functions built into a component, including the test logic. It is composed of a minimum of the three input connections that are TCK, TMS, TDI, and one output connection (TDO). An optional fourth input connection (___T___R___S___T) provides asynchronous initialization of the test logic.

The signals of TAP are described in Table 2

Table 2. TAP signal description

Signal Name	I/O Type	Description
TCK	Input	provides the clock for the test logic
TMS	Input	The value of the signal presented as TMS at the time of a rising edge at TCK determines the next state of the TAP controller, the circuit that controls test operations.
TDI	Input	Serial test instructions and data are received by the test logic at TDI.
TDO	Output	the serial output for test instructions and data from the test logic
___T___R___S___T	Input	provides asynchronous initialization and active low

2.6 BSDL

BSDL is the abbreviation of the Boundary-Scan Description Language. Although BSDL is based on the syntax and grammar of VHDL (Very high-speed integrated-circuit Hardware Description Language), it is not a general-purpose hardware description language and is intended solely as a means of describing key aspects of the implementation of the boundary scan within a particular component.

Normally, the BSDL file contains the following elements described in Table 3.

Table 3. BSDL elements

Element	Description
Entity Description	statement for device name or functionality
Generic Parameter	description for package or pin mapping
Logical Port Description	description for pin type such as in, out, inout, linkage
Standard Use Statement	references external definitions
Component Conformance Statement	standards to follow
Device Package Pin Mapping	description for pin mapping
Scan Port Identification	pin description on device for JTAG TAP including TCK, TMS, TDI, TDO
Compliance Enable Description	the pins involved in entering boundary scan mode and the level applied to the pins (it is useful when you make a chip enter boundary scan mode)
Instruction Register Description	instruction length and instruction code, sometimes includes a device-specific instruction also called a private instruction
Register Access Description	registers corresponding to specific instructions

Table 3. BSDL elements*...continued*

Element	Description
Boundary-Scan Register Description	The list records the boundary scan cells and the functionality of these boundary scan cells.

2.7 More information on JTAG and boundary scan

For more information on JTAG and boundary scan, refer to the links below:

- Home page for JTAG and boundary scan:https://www.jtag.com/
- IEEE Std 1149.1

Version 1990: https://standards.ieee.org/standard/1149_1-1990.html

Version 2001: https://standards.ieee.org/standard/1149_1-2001.html

Version 2013: https://standards.ieee.org/standard/1149_1-2013.html

3 Build boundary scan test environment

This section provides details on how to build the boundary scan test environment.

3.1 Introduction to boundary scan test tool suite

In this application note, the boundary scan test uses the tool set from Lauterbach that is an all-in-one debug and trace solution for Cortex-M. This tool set includes the following two parts:

3.1.1 µTrace for Cortex-M

µTrace for Cortex-M is one of the architecture-specific products from Lauterbach and it has the following features:

- On-chip/external flash programming, debug, trace, JTAG boundary scan
- Recommended for single-core microcontrollers with Cortex-M
- Recommended for multicore microcontrollers with solely Cortex-M (single debug port)
- 256 Mbit trace memory
- USB 3 interface to the host computer
- TRACE32 Streaming up to 150 Mbit
- TRACE32 Mixed Signal Probe supported

This application note uses LA-4533 to perform the boundary scan. For more information about µTrace for Cortex-M, see Figure 2.

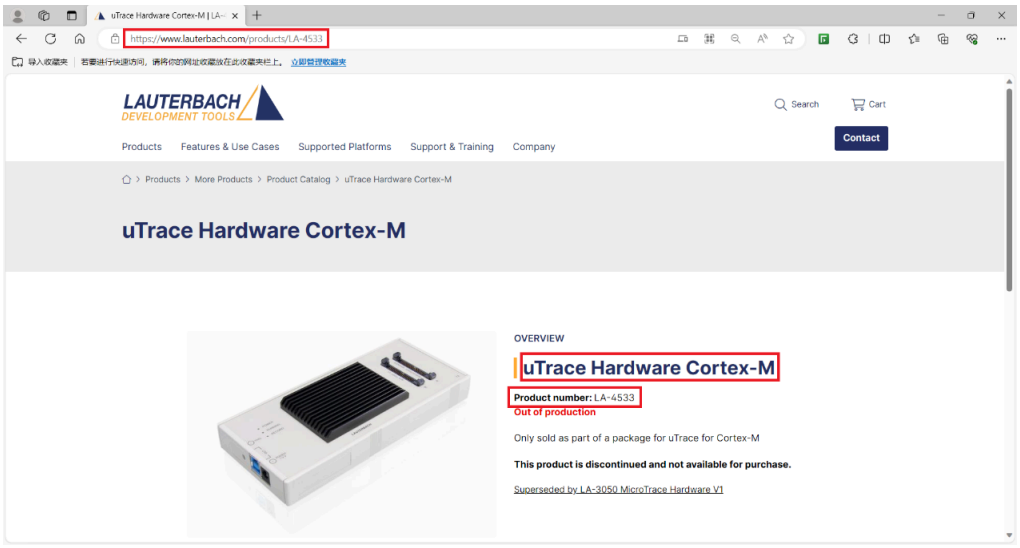

Figure 2. µTrace for Cortex-M debugger

3.1.2 TRACE32

TRACE32 is a simulation test tool developed by Lauterbach, it runs on PC. It is used with µTrace for Cortex-M for on-chip/external Flash programming, debugging, tracing, and JTAG boundary scan. It supports various processor architectures, including standard processors such as ARM, MIPS, PowerPC and DSP, soft cores, and coprocessors.

For the boundary scan, TRACE32 not only provides GUI operations for interactive tests, but also supports scripts for automated tests. If your test must execute various commands, such as system settings, JTAG, BSDL, make a script containing these commands, and complete the test by executing the script. It is efficient and reduces the possibility of errors in the command-line mode.

TRACE32 supports command-line input. The command-line input is at the bottom of the TRACE32 main page, starting with B::. and allows completing a series of operations by inputting commands, such as system reset, system settings, BSDL file loading, boundary scan test.

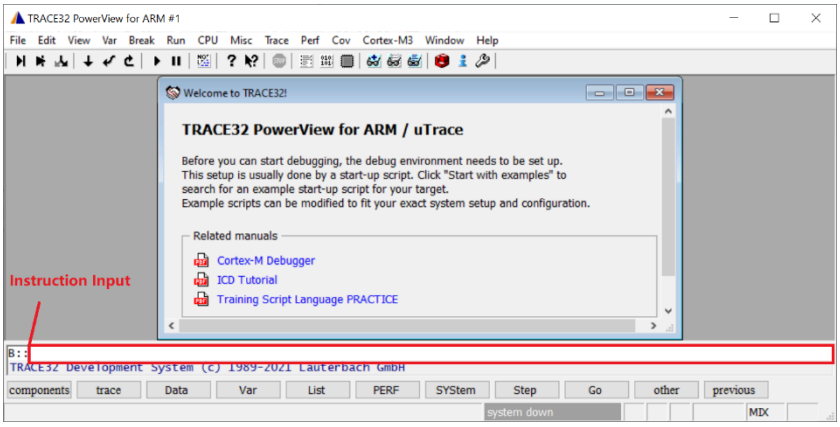

Figure 3. Main page for TRACE32 for ARM

To download TRACE32, see Figure 4.

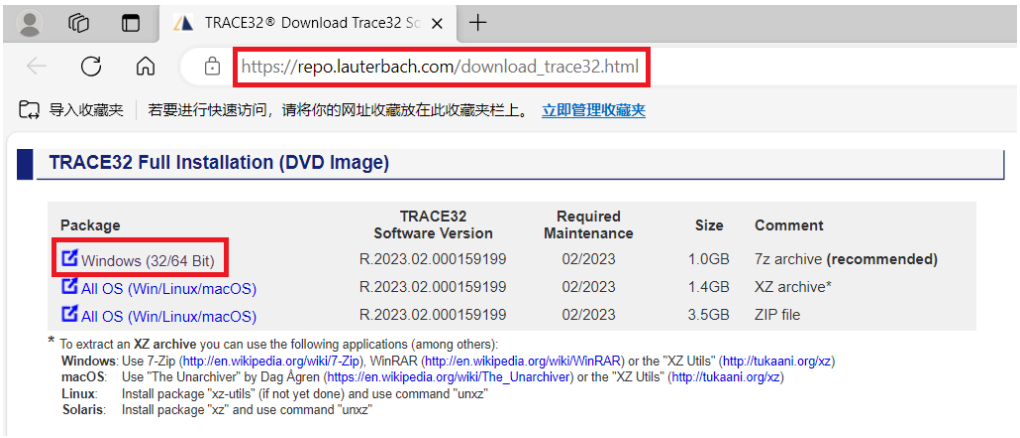

Figure 4. Download TRACE32

3.2 Hardware environment

µTrace for Cortex-M consists of:

- Universal debugger
- Debug cable

Figure 5. µTrace for Cortex-M composition

Figure 6 shows the schematic diagram for hardware connection of the entire boundary scan system.

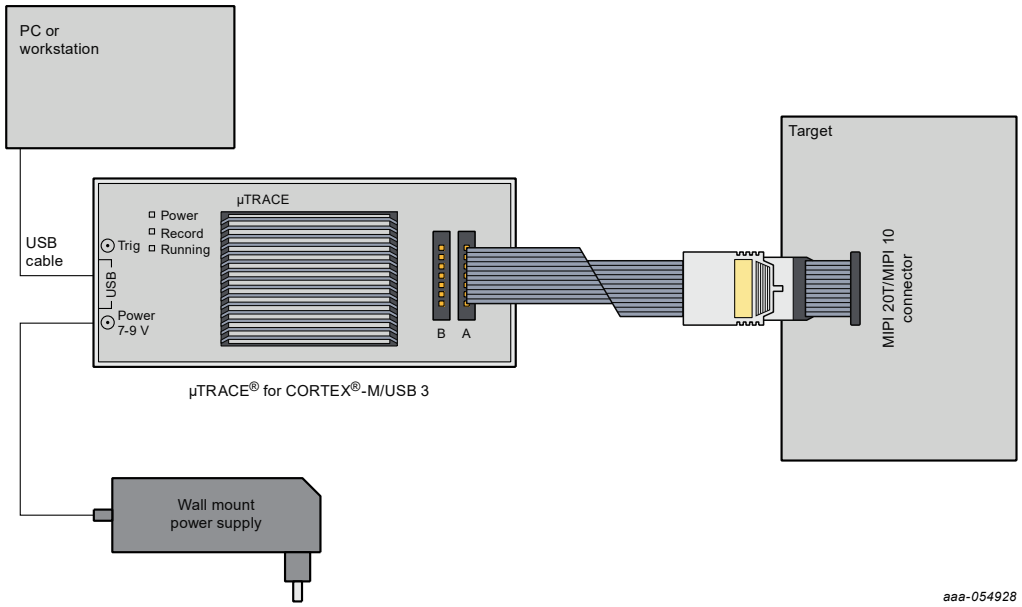

Figure 6. Schematic diagram of hardware connection

Suggestions for users include:

- To prevent the debugger or target board from being damaged, do not plug or unplug the debugger while the target board is powered on. The recommended sequence for powering on or off is as follows:

— Power on: debugger >target board

— Power off: target board>debugger

- The debugger interface has pin 1. Double-check the direction to prevent damage to the debugger or the target board.

The steps for setting up the hardware environment are described as follows:

1. Connect the µTrace for Cortex-M debugger to the FRDM-MCXA153 board through a standard JTAG interface.

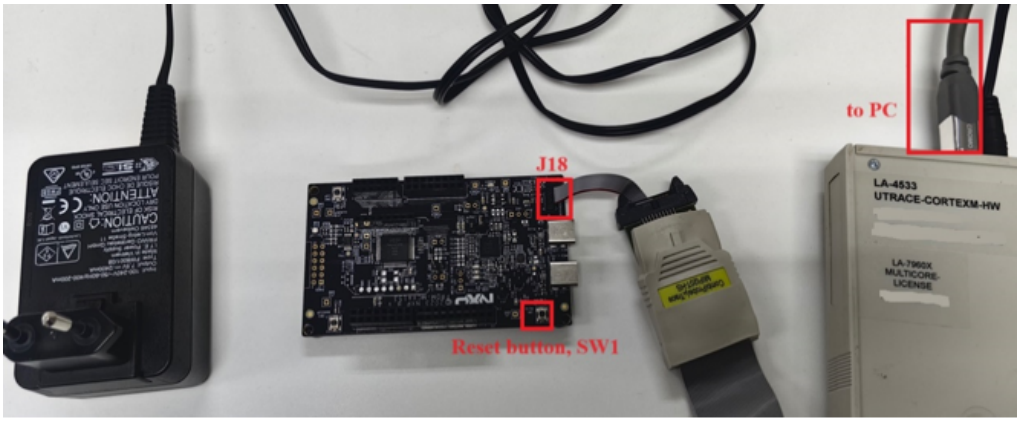

Figure 7. Boundary scan hardware setting

2. Connect the µTrace for Cortex-M debugger to the PC through the USB cable, and then power on the debugger with a 5 V power adapter. Open **Device Manager** on the PC. Lauterbach equipment appears in Trace32 Devices as shown on Figure 8. If not, check the connection.

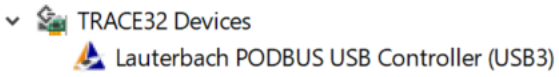

Figure 8. Lauterbach equipment in Device Manager

3. Power up your target board.

3.3 Enter boundary scan mode

To let the MCXA series stay in boundary scan mode, keep the **Reset** button (SW1) in the FRDM-MCXA153 board pressed during the boundary scan test.

4 Interactive boundary scan test

To perform the boundary scan test using µTrace for Cortex-M debugger and TRACE32 software, follow the steps below:

1. Open the TRACE32 software and choose the ARM32 USB.

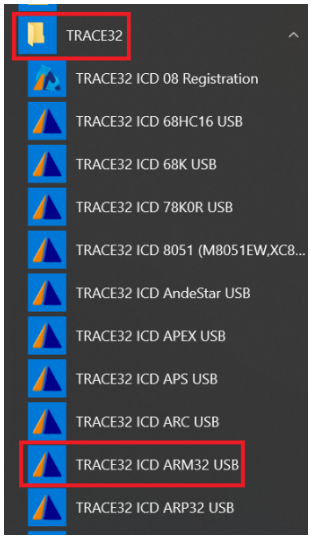

Figure 9. Open TRACE32 for ARM32

2. The main page of TRACE32 for ARM32 is shown in Figure 10. If the status bar at the bottom of the main page shows power down instead of system down, check the power supply of the debugger and the connection with the JTAG interface of the FRDM-MCXA153 board.

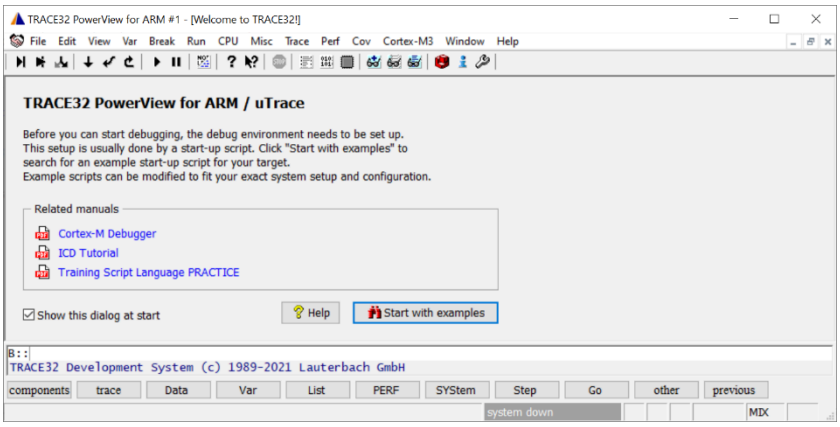

Figure 10. Main page of TRACE32 for ARM32

3. Click **CPU->System Settings…** in the menu bar, a system setting dialog appears. Perform the system settings as shown in Figure 11.

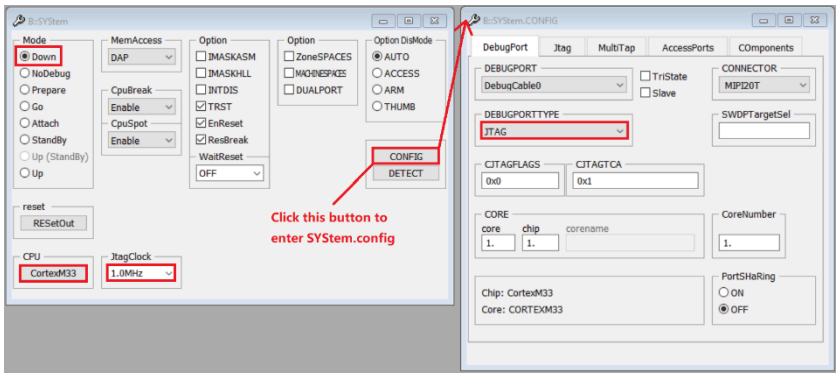

Figure 11. System settings

4. Type the below commands, with each command followed by entering:

```
BSDL.RESet
BSDL.ParkState Select-DR-Scan
BSDL.state
```

5. The BSDL.state window appears. Click the **FILE** button and load the BSDL file you want to validate. The BSDL.state window is shown in Figure 12.

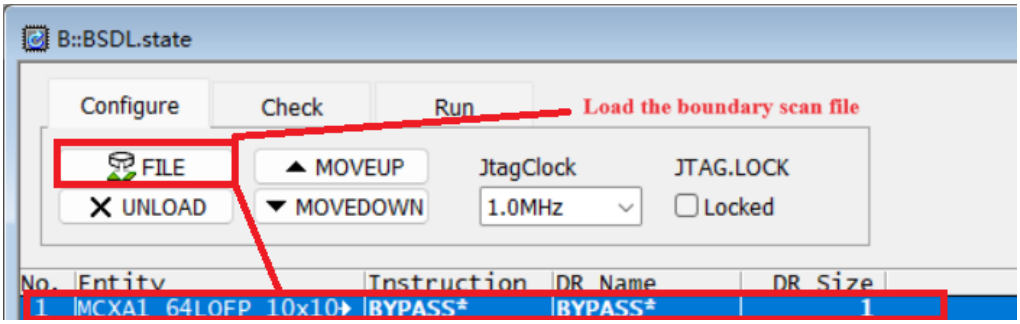

Figure 12. BSDL. state window

6. After loading the file, type in the command below:

```
BSDL.SOFTRESET
```

7. Switch to the **Check** tab of the BSDL.state window. As shown in Figure 13 and Figure 14, click the **BYPASSall** button and the **IDCODEall** button to see if both results can pass. Double-click the entity

name as shown in Figure 14. The IDCODE test result can be seen in the BSDL.SET window as shown in Figure 15.

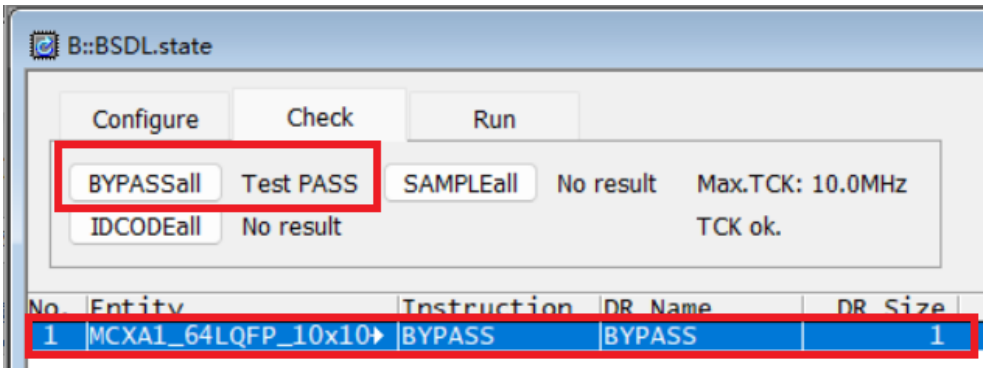

Figure 13. Check BYPASS

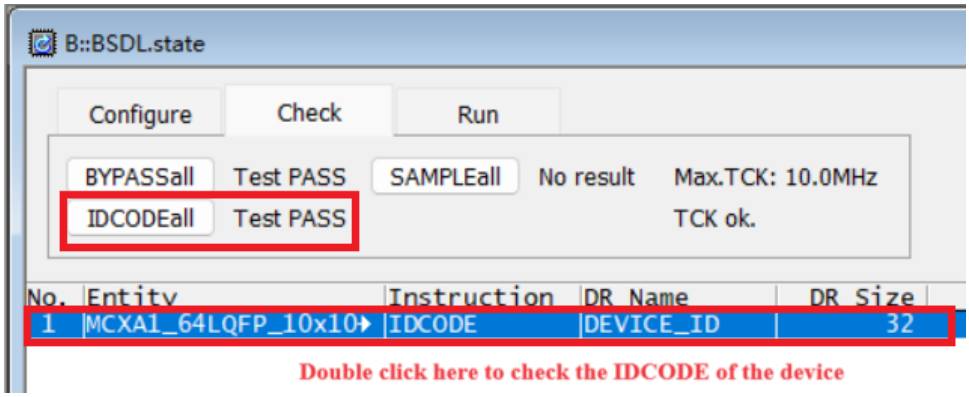

Figure 14. Check IDCODE

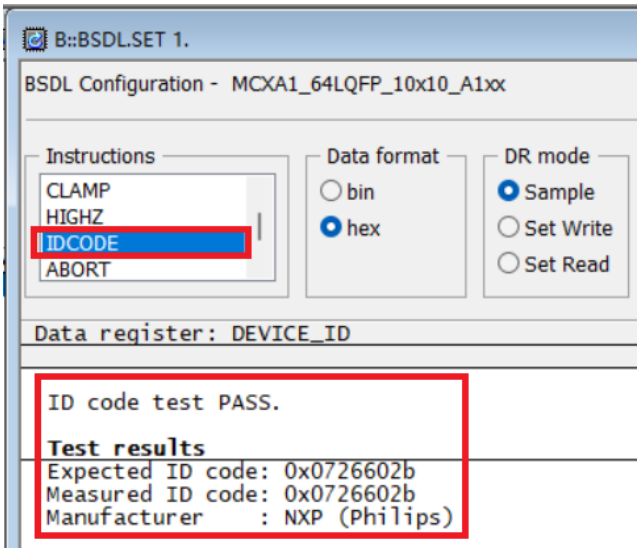

Figure 15. IDCODE test result

8. After clicking the **SAMPLEall** button, **No result** becomes **Test done**. Double-click the entity name as shown in Figure 16, the SAMPLE test result can be seen in the BSDL.SET window as shown in Figure 17.

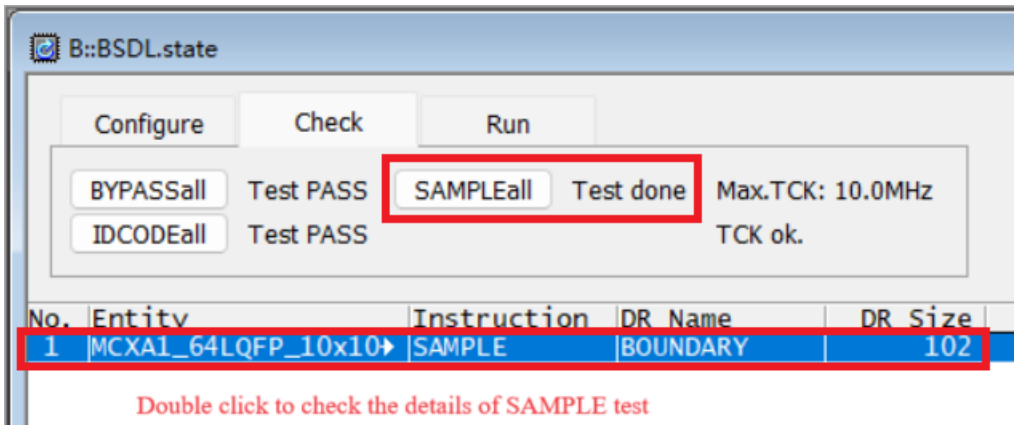

Figure 16. Check SAMPLE

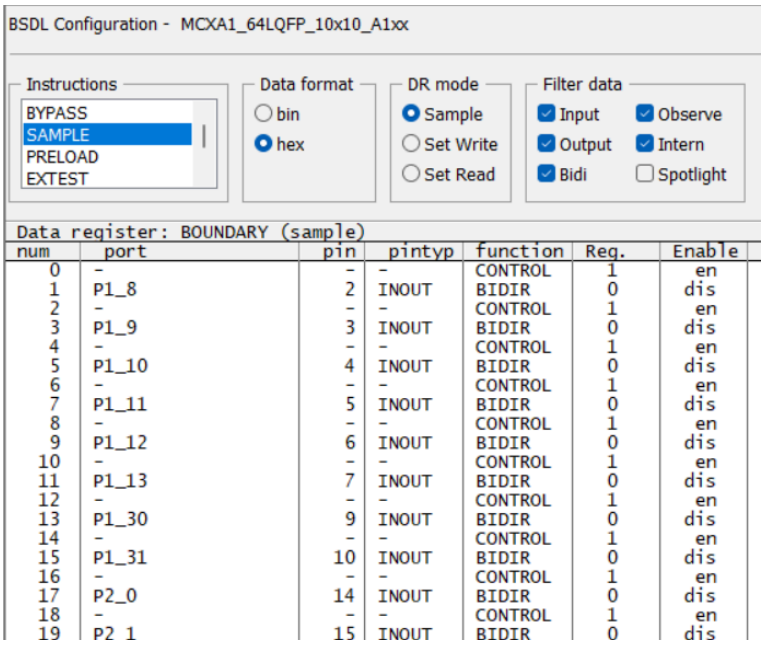

Figure 17. SAMPLE test result

9. Enter the BSDL.SET command on the TRACE32 command line, and the BSDL.SET window appears. In the **Instructions** field, click **EXTEST** and in the **DR mode** field, choose **Set Write** as shown in Figure 18

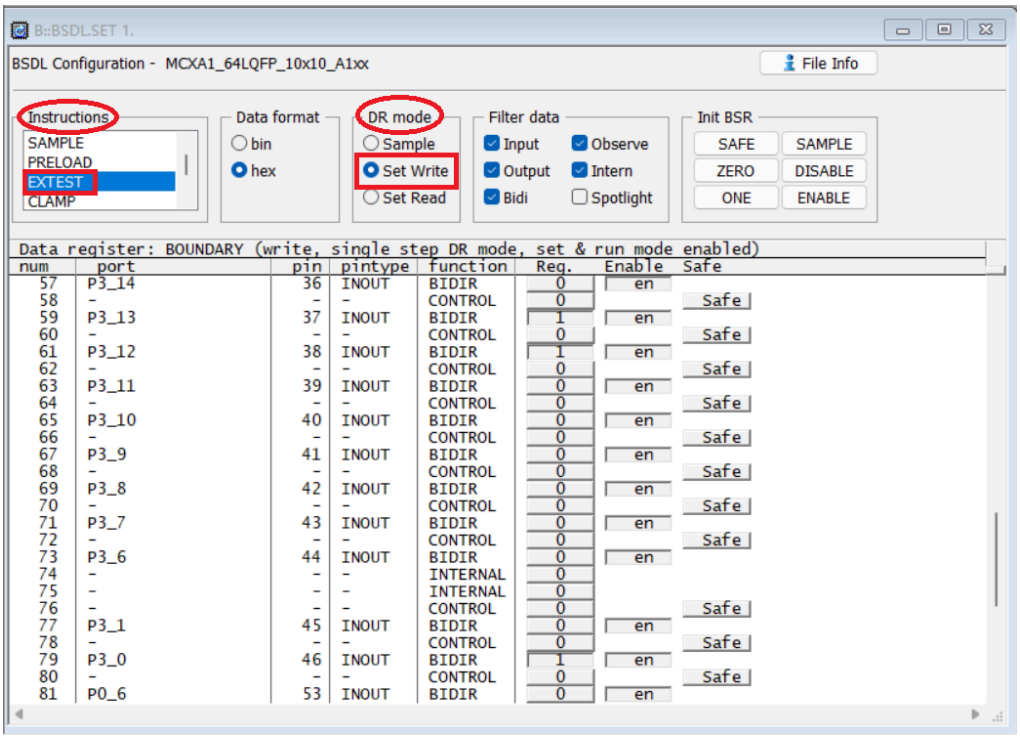

Figure 18. EXTEST settings in BSDL.SET window

Switch to the **Run** tab in the BSDL.state window and click **SetAndRun**, as shown in Figure 18.

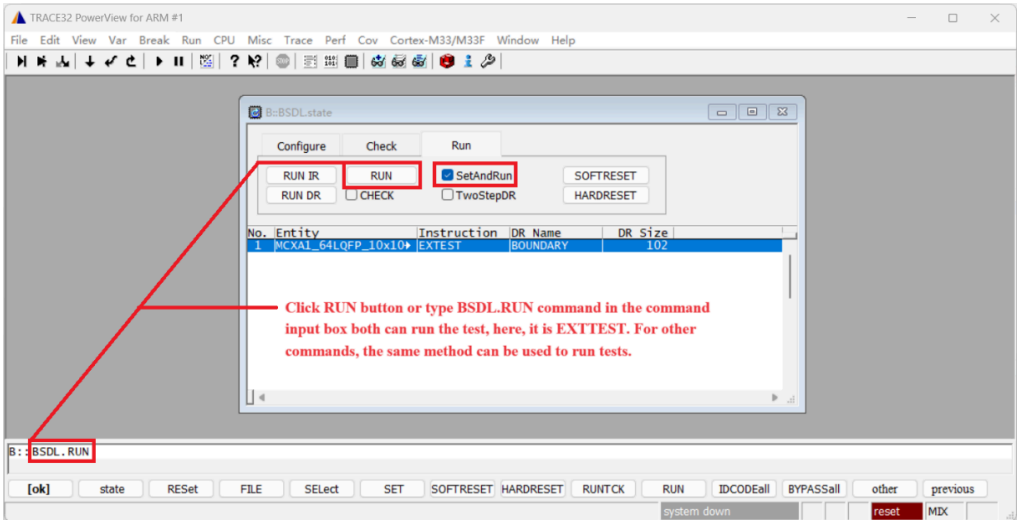

Figure 19. EXTEST settings in BSDL.state window

Switch back to the BSDL.SET window. The FRDM-MCXA153 board has an RGB led. This led has R, G, B components and they are connected to P3_12, P3_13, and P3_0 pins respectively. The R, G, B components are off or on when P3_12, P3_13, or P3_0 output is high or low level. Judge the pin output level by observing the corresponding RGB component status.

Before controlling the output level of a pin, enable it by clicking the **ENABLE** button in the **Init BSR** group or the **en** button in the **Enable** column corresponding to the pin you want to test. Control the pin to output a high or low level by changing the button status in the Reg. column corresponding to the tested pin to 1 or 0.

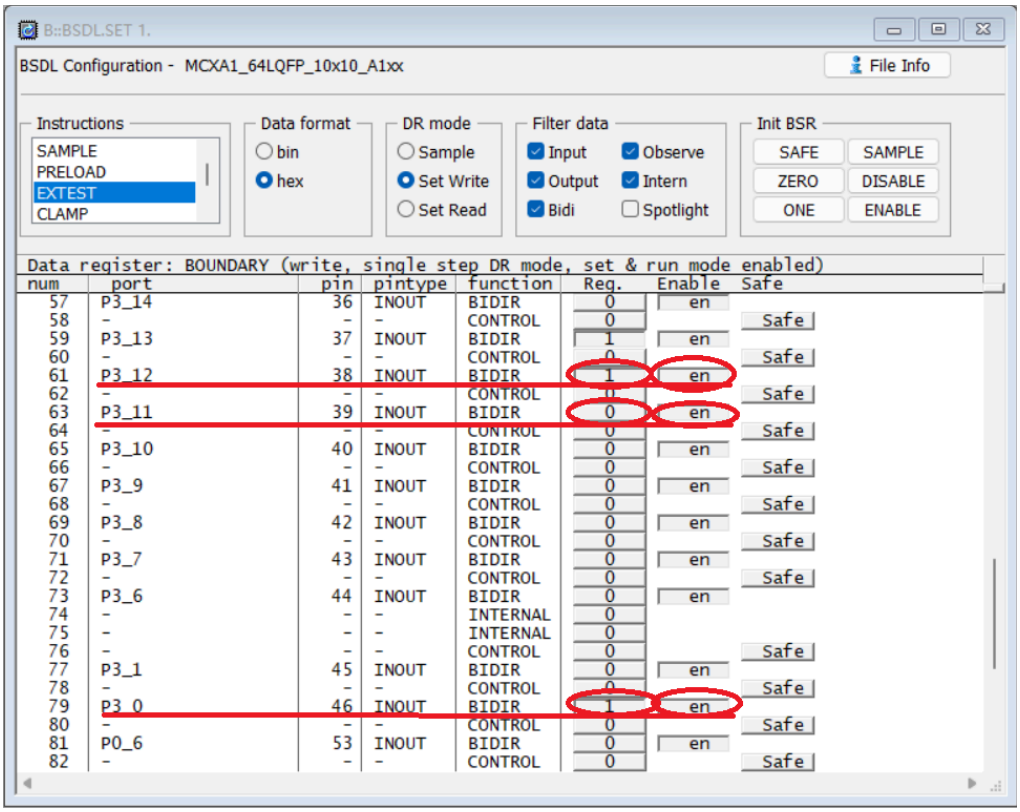

Figure 20. Enable and toggle signal level

According to the above method, traverse all the IO pins defined in the BSDL file. If all the IO pins pass the test, it means that the EXTTEST test of BSDL has passed.

10. The PRELOAD test is used with EXTTEST. As Figure 21 shows, clicking the **ONE** or **ZERO** button can control all GPIOs to output high or low level and clicking the **ENABLE** button in the **Init BSR** group can enable all GPIOs.

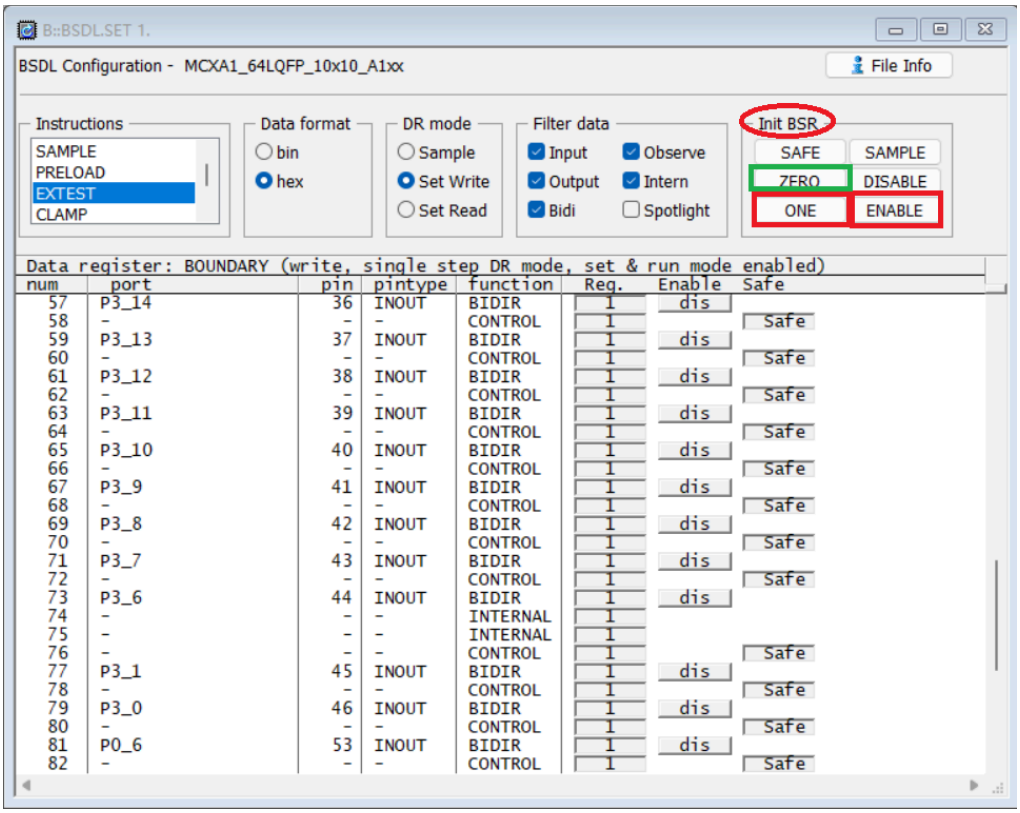

Figure 21. Preload IO output status

Run PRELOAD and then run EXTTEST, use a multimeter to measure if the logic state of all IO pins really matches the preload values.

According to the above method, traverse all the IO pins defined in the BSDL file. If all the IO pins pass the test, it means that the PRELOAD test of BSDL has passed.

11. Choose **HIGHZ** in the **Instructions** group in the BSDL.SET window and then run the HIGHZ test. All IO pins defined in the BSDL file are in a high-impedance state. Taking 3.3 V logic as an example, if an intermediate level, such as 1.65 V, is applied to a pin in a high-impedance state, applying a multimeter on the pin, it must be 1.65 V. This is because the pin in the high-impedance state is regarded as an open circuit due to its high-impedance and will not cause significant voltage drop to the external driving source.

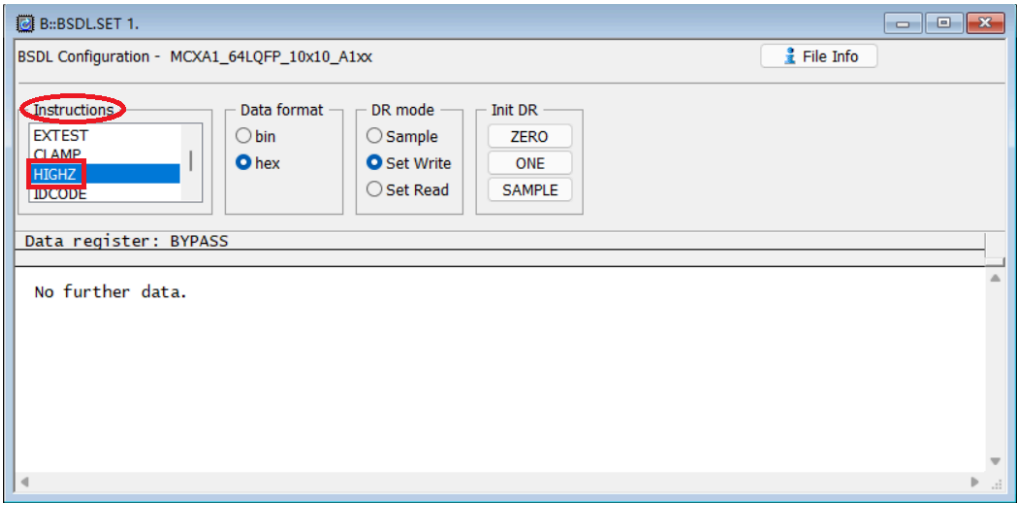

Figure 22. HIGHZ settings

After the HIGHZ test starts, all pins are in a high-impedance state. Select a pin and apply an intermediate level, here 1.64 V is selected. Use a multimeter to measure the pin level. Figure 23 shows the HIGHZ test result of the P1_6 pin.

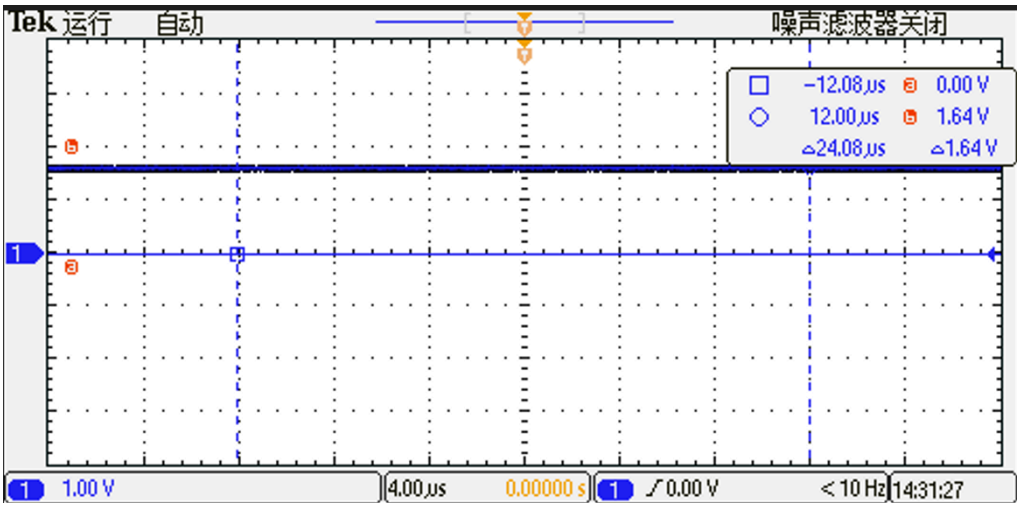

Figure 23. HIGHZ test result

As Figure 23 shows, when the external driving source is applied to the pin P1_6, it causes almost no voltage drop, indicating that P1_6 is indeed in a high-impedance state.

According to the method described above, traverse all the IO pins defined in the BSDL file. If all the IO pins pass the test, it means that the HIGHZ test of BSDL has passed.

5 Automated boundary scan test

The automated boundary scan test is more convenient for quick testing. To improve the test efficiency, TRACE32 supports a practice script. The automated boundary scan test can be performed by writing a script program.

In the **File** menu item on the main page of TRACE32, three submenu items related to the script are provided: **New Script**, **Open Script...**, and **Run Script...**. They are used to create, open, and run script.

A script example used to automate the boundary scan test is presented below.

```
;System setup
SYStem.Mode Down                    ;Disables the debug mode.
SYStem.CPU  CortexM33               ;Tells TRACE32 the exact CPU type
                                    ;used on your target, CPU core of
                                    ;LPC553x is Cortex-M33.
SYStem.CONFIG.DEBUGPORTTYPE JTAG    ;Specifies which probe cable shall
                                    ;be used, here, JTAG is selected
SYStem.JtagClock    1MHz            ;Selects JTAG frequency (TCK)
;BSDL Settings
BSDL.RESet                          ;Initialize the boundary scan engine
BSDL.ParkState Select-DR-Scan       ;Set PartState as Select-DR-Scan
BSDL.state                          ;Open BSDL.state window
;Configure boundary scan chain
BSDL.FILE lpc553x100.bsdl           ;your BSDL file name
;Check boundary scan chain
BSDL.SOFTRESET
IF !BSDL.CHECK.BYPASS()             ;BYPASS Test
(
BSDL.BYPASSall
```

```
PRINT %ERROR "Bypass test failed."
ENDDO
)
IF !BSDL.CHECK.IDCODE()            ;IDCODE Test
(
 BSDL.IDCODEall
 PRINT %ERROR "ID code test failed."
 ENDDO
)
;Perform SAMPLE test
BSDL.SAMPLEall

;Perform EXTTEST
;Pin output settings, you can add other pin output settings
BSDL.SET 1. PORT PIO0_7 0            ;Set PIO0_7 output as 0
BSDL.RUN DR                          ;Only apply data register settings
                                     ;to the boundary scan chain
BSDL.SET 1. IR EXTEST                ;Only apply instruction register
                                     ;settings to the boundary scan chain
BSDL.RUN                             ;BSDL run

;Perform HIGHZ test
BSDL.SET 1. IR HIGHZ                 ;Only apply instruction register
                                     ;settings to the boundary scan chain
BSDL.RUN                             ;BSDL run
```

6 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2024 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7 Revision history

Table 4. Revision history

Document ID	Release date	Description
AN14209 v.1	12 April 2024	Initial version

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

Date of release: 12 April 2024
Document identifier: AN14209