AN14131

How to Reach ADC Maximum Conversion Speed on LPC86x

Rev. 1 — 24 November 2023 Application note

Document information

Information	Content
Keywords	AN14131, LPC86x, ADC, LPCXpresso860-MAX
Abstract	This application note describes how to test maximum ADC conversion speed on LPC86x and provides example code to implement tests.

1 Introduction

1.1 Introduction

This application note describes how to test maximum Analog-to-Digital Converter (ADC) conversion speed on LPC86x and provides example code to implement tests. The 12-bit ADC is a successive-approximation ADC designed for operation within an integrated microcontroller system-on-chip. It is available on all LPC84x and LPC86x devices.

The main features of ADC module are:

- 12-bit successive approximation ADC.
- Input multiplexing among 12 pins.
- Two configurable conversion sequences with independent triggers.
- Optional automatic high/low threshold comparison and **zero-crossing** detection.
- Power-down mode and low-power operating mode.
- Measurement range VREFN to VREFP (typically 3 V; not to exceed VDDA voltage level).
- 12-bit conversion rate of up to 1.9 Msamples/s.
- Burst conversion mode for single or multiple inputs.
- DMA support.
- Hardware calibration mode.

1.2 LPC86x ADC Usage Note

LPC86x ADC is enhanced based on LPC84x. Before development, pay attention to the following items:

- Calibrate ADC at 30 MHz core frequency and raise core frequency to the maximum frequency after ADC calibration is successful.
- A programmable divider is included to scale the system clock to the maximum ADC clock rate of 48 MHz.
- A fully accurate conversion requires 25 of these ADC clocks. The ADC maximum sample rate is 1.9Msps = 48MHz/25 cycle.
- To reach maximum ADC conversion speed, the VDD and VDDA must be in 2.4V to 3.6V.

1.3 Test result

As described in the data sheet:

Table 1. ADC maximum conversion speed in data sheet

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{IA}	analog input voltage		0	—	VDDA	V
V_{ref}	reference voltage	on pin VREFP	2.4	—	VDDA	V
C_{ia}	analog input capacitance		—	—	26	pF
$f_{clk\ (ADC)}$	ADC clock frequency		—	—	48	MHz
f_s	sampling frequency		—	—	1.9	Msamples/s

Table 1. ADC maximum conversion speed in data sheet*...continued*

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
E_D	differential linearity error		—	±3.0	—	LSB
$E_{L(adj)}$	integral nonlinearity		—	±2.5	—	LSB
E_O	offset error		—	±2.5	—	LSB
$V_{err(fs)}$	full-scale error voltage		—	0.1	—	%
Z_i	input impedance	f_s = 1.9 Msamples/s	0.06	—	—	MΩ

This application note provides example code to verify the conversion speed.

The results are as below:

- In the 12-bit mode, the maximum conversion speed can reach 1.9 Msps.

As shown in Table 1, the test results are aligned with the data sheet.

2 Implementation

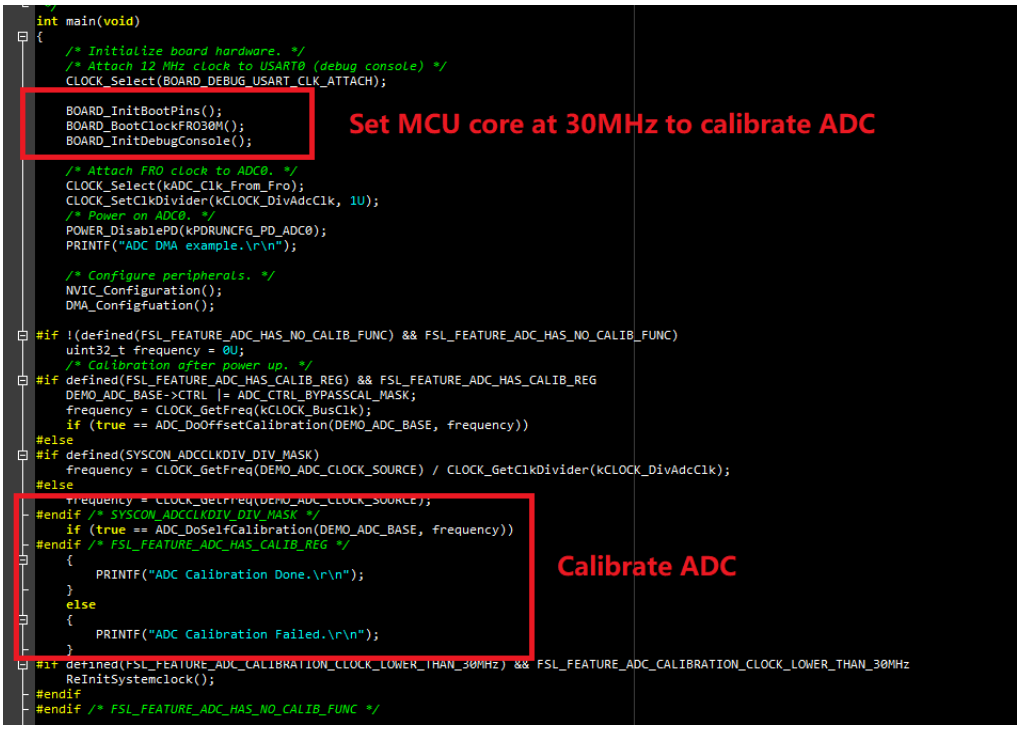
To reach the maximum conversion speed, use DMA to receive and transfer ADC conversion result. So, the example code is based on the SDK example:

\SDK_2_13_0_LPCXpresso860MAX\boards\lpcxpresso860max\driver_examples\adc\lpc_adc_dma

Make sure that you have already been familiar with that example above and related hardware.

To reach the maximum conversion speed, configure ADC in the fastest mode, which includes:

1. Set MCU start at 30 MHz and finish ADC calibration at the beginning.




```
int main(void)
{
    /* Initialize board hardware. */
    /* Attach 12 MHz clock to USART0 (debug console) */
    CLOCK_Select(BOARD_DEBUG_USART_CLK_ATTACH);

    BOARD_InitBootPins();
    BOARD_BootClockFRO30M();
    BOARD_InitDebugConsole();

    /* Attach FRO clock to ADC0. */
    CLOCK_Select(kADC_Clk_From_Fro);
    CLOCK_SetClkDivider(kCLOCK_DivAdcClk, 1U);
    /* Power on ADC0. */
    POWER_DisablePD(kPDRUNCFG_PD_ADC0);
    PRINTF("ADC DMA example.\r\n");

    /* Configure peripherals. */
    NVIC_Configuration();
    DMA_Configfuation();

#if !(defined(FSL_FEATURE_ADC_HAS_NO_CALIB_FUNC) && FSL_FEATURE_ADC_HAS_NO_CALIB_FUNC)
    uint32_t frequency = 0U;
    /* Calibration after power up. */
#if defined(FSL_FEATURE_ADC_HAS_CALIB_REG) && FSL_FEATURE_ADC_HAS_CALIB_REG
    DEMO_ADC_BASE->CTRL |= ADC_CTRL_BYPASSCAL_MASK;
    frequency = CLOCK_GetFreq(kCLOCK_BusClk);
    if (true == ADC_DoOffsetCalibration(DEMO_ADC_BASE, frequency))
#else
#if defined(SYSCON_ADCCLKDIV_DIV_MASK)
    frequency = CLOCK_GetFreq(DEMO_ADC_CLOCK_SOURCE) / CLOCK_GetClkDivider(kCLOCK_DivAdcClk);
#else
    frequency = CLOCK_GetFreq(DEMO_ADC_CLOCK_SOURCE);
#endif /* SYSCON_ADCCLKDIV_DIV_MASK */
    if (true == ADC_DoSelfCalibration(DEMO_ADC_BASE, frequency))
#endif /* FSL_FEATURE_ADC_HAS_CALIB_REG */
    {
        PRINTF("ADC Calibration Done.\r\n");
    }
    else
    {
        PRINTF("ADC Calibration Failed.\r\n");
    }
#if defined(FSL_FEATURE_ADC_CALIBRATION_CLOCK_LOWER_THAN_30MHz) && FSL_FEATURE_ADC_CALIBRATION_CLOCK_LOWER_THAN_30MHz
    ReInitSystemclock();
#endif
#endif /* FSL_FEATURE_ADC_HAS_NO_CALIB_FUNC */
```

Figure 1. ADC calibrated

2. Re-initialize the system clock if the system requires a system frequency faster than 30 MHz.

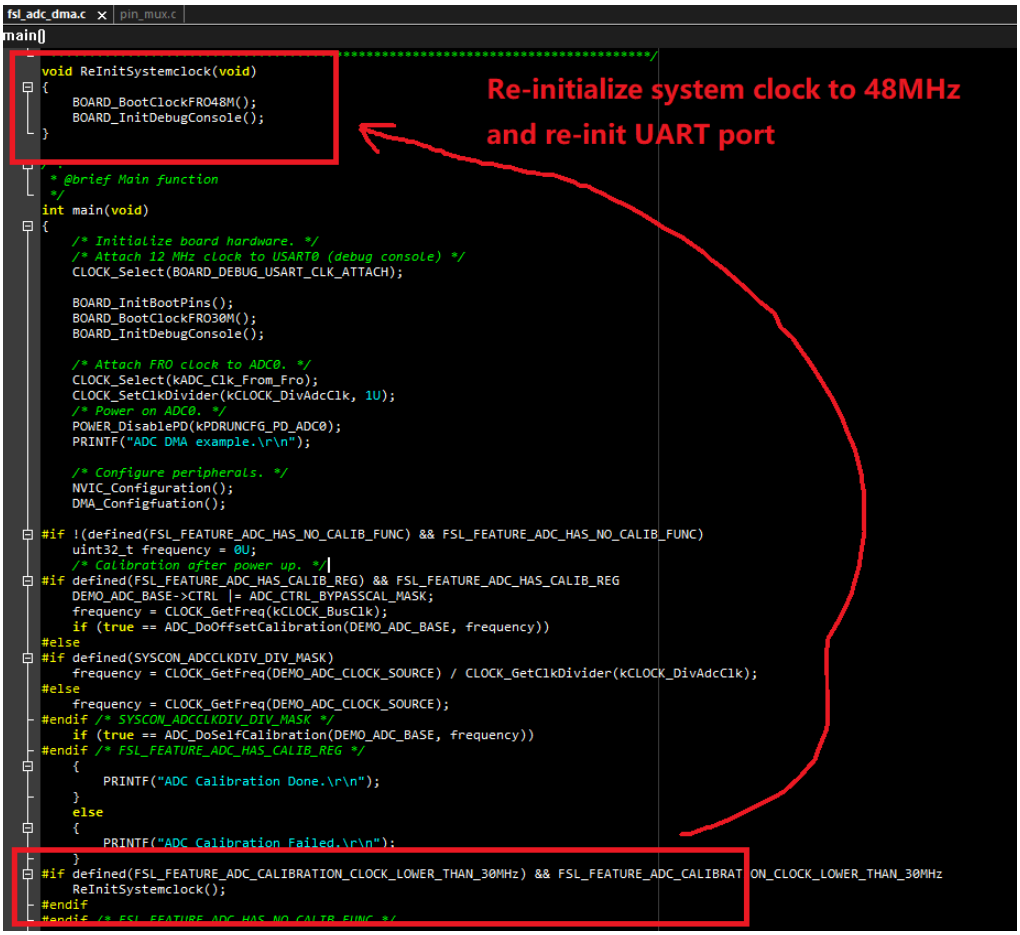


Figure 2. Re-initialize system frequency to 48 MHz

3. Change the ADC input clock to **FRO48MHz** if needed (ADCCLK = 48 MHz, maximum allowable ADC clock on LPC86x).

```
/* Attach FRO clock to ADC0. */
CLOCK_Select(kADC_Clk_From_Fro);
CLOCK_SetClkDivider(kCLOCK_DivAdcClk, 1U);
```

4. ADC sets to continue conversion mode, which means that after the conversion command execution completes, the next conversion command automatically starts.

```
static void DMA_Configfuation(void)
{
    dma_channel_config_t dmaChannelConfigStruct;
    dma_channel_trigger_t dmaChannelTriggerStruct;

    g_XferConfig = DMA_CHANNEL_XFER(true,                       /* Reload link descriptor after current exhaust, */
                                    true,                       /* Clear trigger status. */
                                    true,                       /* Enable interruptA. */
                                    false,                      /* Not enable interruptB. */
                                    sizeof(uint32_t),           /* Dma transfer width. */
                                    kDMA_AddressInterleave0xWidth, /* Dma source address no interleave  */
                                    kDMA_AddressInterleave0xWidth, /* Dma destination address no interleave  */
                                    sizeof(uint32_t)            /* Dma transfer byte. */
    );
    /* Init DMA. This must be set before INPUTMUX_Init() for DMA peripheral reset will clear the mux setting. */
    DMA_Init(DEMO_DMA_BASE);

    /* Configure DMAMUX. */
    INPUTMUX_Init(INPUTMUX);
    INPUTMUX_AttachSignal(INPUTMUX, DEMO_DMA_ADC_CHANNEL, kINPUTMUX_Adc0SeqaIrqToDma);

    /* Configure DMA. */
    DMA_EnableChannel(DEMO_DMA_BASE, DEMO_DMA_ADC_CHANNEL);
    DMA_CreateHandle(&g_DmaHandleStruct, DEMO_DMA_BASE, DEMO_DMA_ADC_CHANNEL);
    DMA_SetCallback(&g_DmaHandleStruct, DEMO_DMA_Callback, NULL);
    /*
     * Configure the DMA trigger:
     * The DATAVALID of ADC will trigger the interrupt. This signal is also for thie DMA triger, which is changed 0 ->
     * 1.
     */
    dmaChannelTriggerStruct.burst = kDMA_EdgeBurstTransfer1;
    dmaChannelTriggerStruct.type  = kDMA_RisingEdgeTrigger;
    dmaChannelTriggerStruct.wrap  = kDMA_NoWrap;

    /* Prepare and submit the transfer. */
    DMA_PrepareChannelTransfer(&dmaChannelConfigStruct,             /* DMA channel transfer configurationstructure. */
                               (void *)DEMO_ADC_DATA_REG_ADDR,      /* DMA transfer source address. */
                               (void *)g_AdcConvResult,             /* DMA transfer destination address. */
                               g_XferConfig,                        /* Xfer configuration */
                               kDMA_MemoryToMemory,                 /* DMA transfer type. */
                               &dmaChannelTriggerStruct,            /* DMA channel trigger configurations. */
                               (dma_descriptor_t *)&(s_dma_table[0]) /* Address of next descriptor. */
    );
    DMA_SubmitChannelTransfer(&g_DmaHandleStruct, &dmaChannelConfigStruct);

    /* Set two DMA descripters to use ping-pong mode.  */
    DMA_SetupDescriptor(&(s_dma_table[0]), g_XferConfig, (void *)DEMO_ADC_DATA_REG_ADDR, (void *)g_AdcConvResult,
                        (dma_descriptor_t *)&(s_dma_table[1]));
    DMA_SetupDescriptor(&(s_dma_table[1]), g_XferConfig, (void *)DEMO_ADC_DATA_REG_ADDR, (void *)g_AdcConvResult,
                        (dma_descriptor_t *)&(s_dma_table[0]));
}
```

Figure 3. Next DMA transfer starts automatically

2.1 Running the demo

1. Compile AN14131SW based on the previous description. Compile and download the code. Open the UART terminal with 9600-N-8-N-1. Press the **RESET** button to run the code. Then the welcome log displays as below:

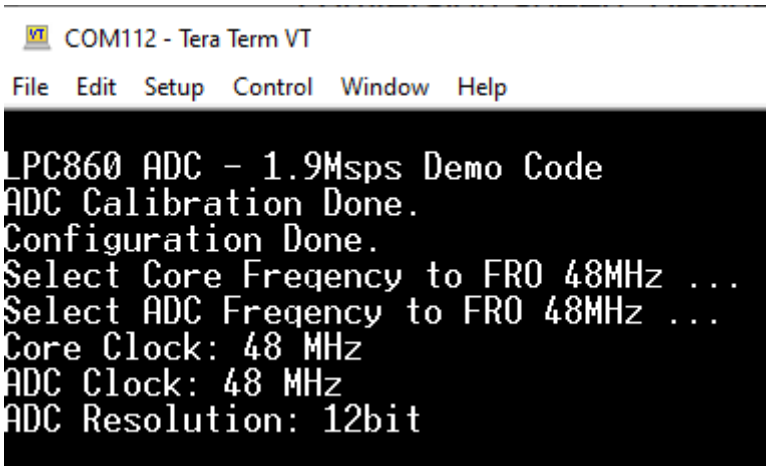


Figure 4. Welcome log of ADC sample rate example

2. Connect to function generator or any other ADC input source (Figure 5 shows the `PIO0_7/ADC_0` position).

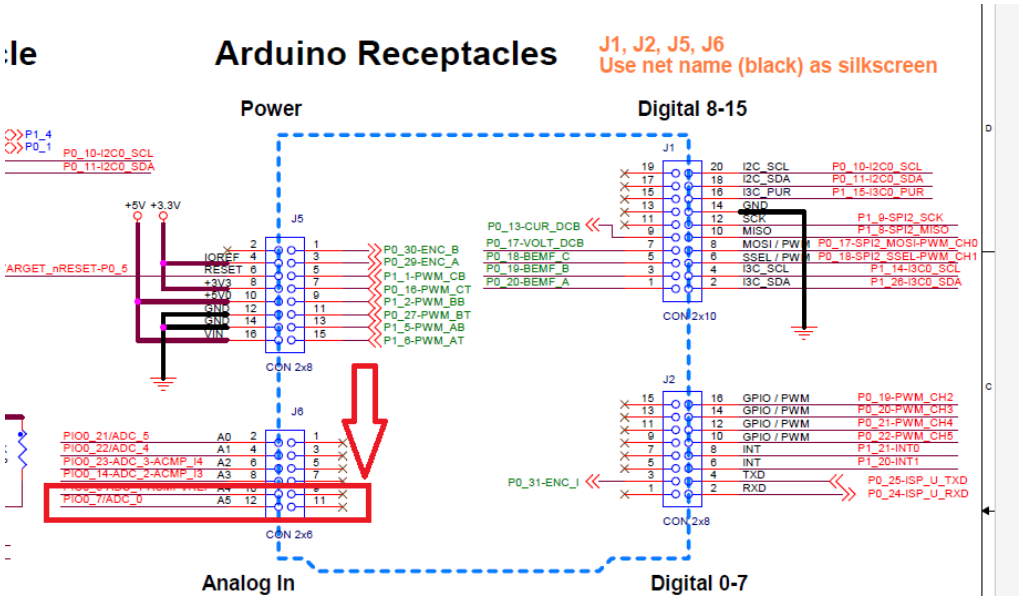


Figure 5. P0_7 (ADC_CHN0, J6-12) used in this example in LPCXpresso860-MAX board

3. Press any key to start the ADC conversion test. The program automatically calculates the conversion time and speed. The result displays on the UART terminal:

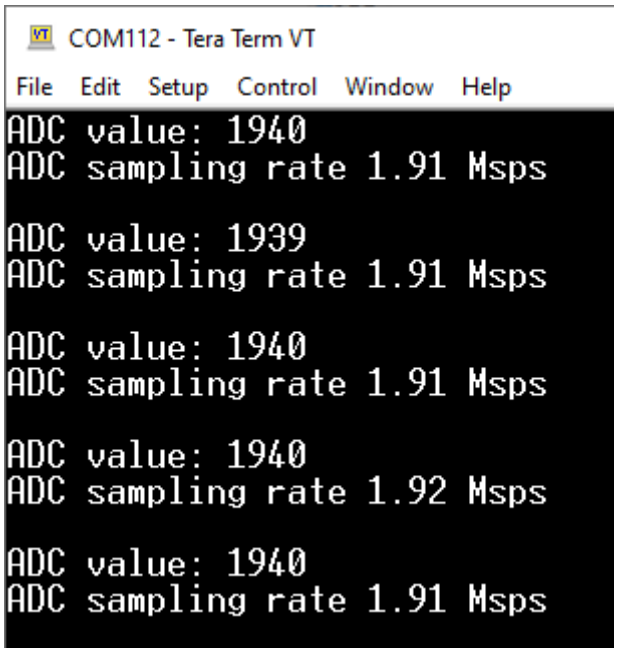


Figure 6. ADC sample rate result in UART terminal

3 Conclusion

This application note provides example code to evaluate ADC maximum conversion speed. The result shows that in 12-bit mode, the ADC maximum conversion speed can reach the specifications in the *LPC86x Data Sheet* (document LPC86x).

4 Reference

- *LPCXpresso860-MAX Board User Manual* (document LPCXpresso860-MAXUM)
- *LPC86x Data Sheet* (document LPC86x)

5 Note about the source code in the document

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

Copyright 2023 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 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.

6 Revision history

Table 2 summarizes the revisions to this document.

Table 2. Revision history

Revision number	Release date	Description
1	24 November 2023	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

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

Date of release: 24 November 2023
Document identifier: AN14131