AN13026

Enable Touch 2-dimension Gesture Recognition based on KL16Z MCU

Rev. 1 — 9 November 2023 Application note

Document information

Information	Content
Keywords	AN13026, Touch, 2D Touch
Abstract	KL16Z MCU with lower cost but similar self capacitive touch sensing hardware.

1 Introduction

A kind of touch 2-dimension (2D) sensing system using self-capacitive touch sensing channels was already described in *Enable 2D Touch Sensing System Based on the KE15Z MCU* (document AN12933). Based on the existing touch 2D sensing algorithm, in this application note document, we do further development to implement a simple 2D gesture recognition algorithm, on the KL16Z MCU with lower cost but similar self-capacitive touch sensing hardware. In the attached demo project, some gestures on the 2D touchpad can be recognized, including:

- Move up/down/left/right
- Touch 1st/2nd/3rd click
- All cover

A compilable source code project for the demo is also provided as an attachment.

2 Touch 2D sensing algorithm overview

First of all, take an overview about the architecture of the touch 2D sensing method based on the self-capacitive touch sensing channels.

Enable 2D Touch Sensing System Based on the KE15Z MCU (document AN12933) describes the method using a group of specially designed touch pads to build a touch 2D touch sensing hardware platform, as shown in Figure 1.

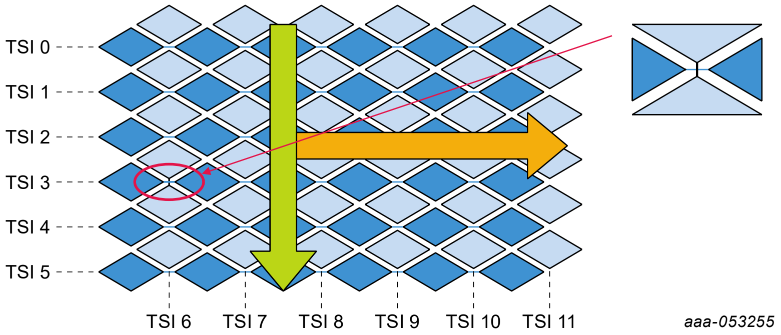

Figure 1. Touch 2D PCB diagram

Actually, they are two groups of touch electrodes organized as two sliders, interleaved with each other, to build a 2D coordinated position system. So, the method just treats the touch 2D sensing system as two touch sliders, and uses the touch slider position calculation algorithm for calculating the touched position separately inside each slider. Then, one slider brings the X value of pixel and the other slider brings the Y value of pixel.

A state machine for processing sensing values in the buffer is implemented to process the samples from the beginning of the application to the position detection cycles, as shown in Figure 2. When the application starts while the sample buffer is empty, the algorithm collects the samples into buffer (Warm up), waits until the samples are stable (Calibration), runs the process to check if the on touch or no touch event is detected and calculates the touch position.

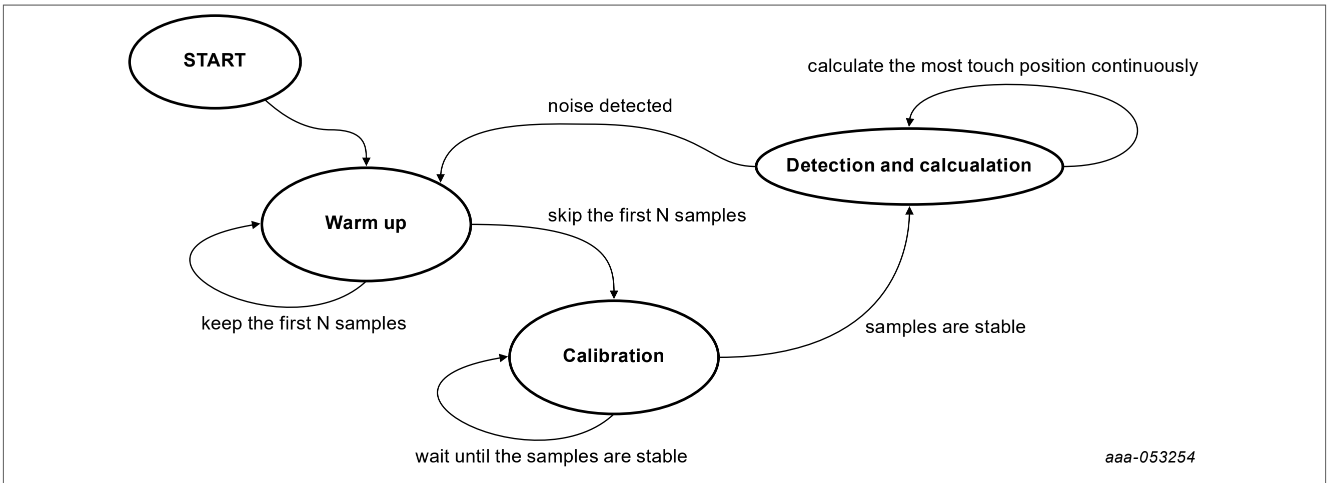

Figure 2. Touch 2D detection state machine

There are also an auto calibration and differences of two levels difference during the position calculation. Once the noise is big enough, the algorithm resets the state machine and runs the calibration again. This makes the algorithm can be self-adapting without manual setting threshold for each sensing channel (even it still needs only one manual setting threshold to set the sensitivity).

When scheduling the tasks of sampling and processing, there is a "ping-pang buffer" mechanism: the sampling service based on the hardware TSI interrupt would capture the current sensing values when the process is dealing with the previous sensing values, as the previous ones were copied to a buffer before starting the new sampling task. As the processing time is always shorter than the sampling time, this makes the sample rate keeps stable and not stalls during the running the processing task. The sampling task and processing task were running parallel, as shown in Figure 3.

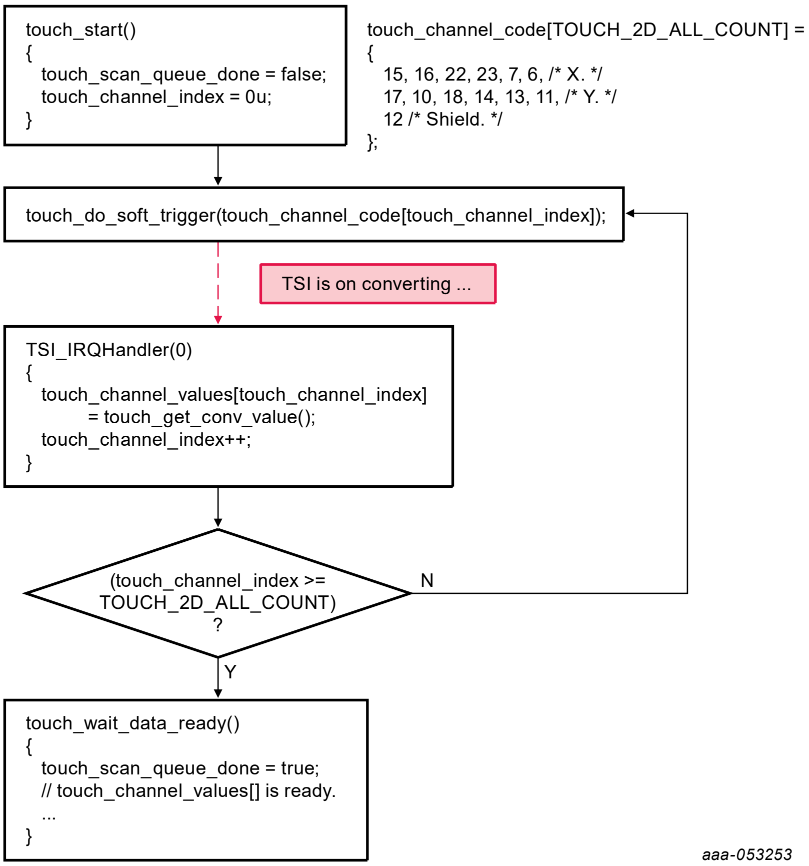
touch_start()
{
touch_scan_queue_done = false;
touch_channel_index = 0u;
}
touch_channel_code[TOUCH_2D_ALL_COUNT] =
{
15, 16, 22, 23, 7, 6, /* X. */
17, 10, 18, 14, 13, 11, /* Y. */
12 /* Shield. */
};
touch_do_soft_trigger(touch_channel_code[touch_channel_index]);
TSI is on converting ...
TSI_IRQHandler(0)
{
touch_channel_values[touch_channel_index]
= touch_get_conv_value();
touch_channel_index++;
}
(touch_channel_index >=
TOUCH_2D_ALL_COUNT)
?
N
Y
touch_wait_data_ready()
{
touch_scan_queue_done = true;
// touch_channel_values[] is ready.
...
}
aaa-053253

Figure 3. Parallel running tasks of sampling and processing

In the end of the processing, the position values of the touched point would be output through the function:

```
app_touch_err = touch_2d_calc_position(app_touch_channel_values,
                       &app_touch_2d_x_pos_raw,
                       &app_touch_2d_y_pos_raw);
```

In this application note, the further development would be based on the position values in the previous algorithm, keeping the continuous calculated positions in a queue, and recognizing the gestures according to their movement of position. In the demo code, we would like to implement a function like the following:

```
app_touch_2d_gesture_event_raw =
 touch_2d_gesture_process( app_touch_2d_x_pos_raw,
                           app_touch_2d_y_pos_raw,
                          (app_touch_err == 0) );
```

The `touch_2d_gesture_process()` function would get the X and Y value of the touched position, and whether the touch is available, then output the judgment of the event in the following list:

```
typedef enum
{
  eTouch_2d_gesture_event_no_touch_keep = 0, /* no touch. */
  //eTouch_2d_gesture_event_no_touch_just = 1,
  eTouch_2d_gesture_event_on_touch_just = 2, /* 1st click. */
  eTouch_2d_gesture_event_on_touch_move_0 = 3, /* no move, on touch. */
  eTouch_2d_gesture_event_on_touch_move_1 = 4, /* move up. */
  eTouch_2d_gesture_event_on_touch_move_2 = 5, /* move down. */
```

```
   eTouch_2d_gesture_event_on_touch_move_3 = 6, /* move left. */
   eTouch_2d_gesture_event_on_touch_move_4 = 7, /* move right. */

   /* to support continuous clicks. */
   eTouch_2d_gesture_event_no_touch_interval, /* short leave interval. */
   eTouch_2d_gesture_event_on_touch_just_2nd_click, /* 2nd click. */
   eTouch_2d_gesture_event_on_touch_just_3rd_click, /* 3rd click */

   eTouch_2d_gesture_event_on_touch_no_process, /* other. */
 } touch_2d_gesture_event_t;
```

3 Hardware overview

The Touch 2D KL16Z board is using 5 + 5 channels to build the Touch 2D sensing system. There are also 11 LEDs on the board to show the running status of the application. The MCU is MKL16Z64VFT4, with Arm Cortex-M0+ core @ 48 MHz core clock, 64 KB FLASH, 8 KB RAM, QFN48 package, and TSI module, which is used to support touch sensing application. The board is as shown in Figure 4.

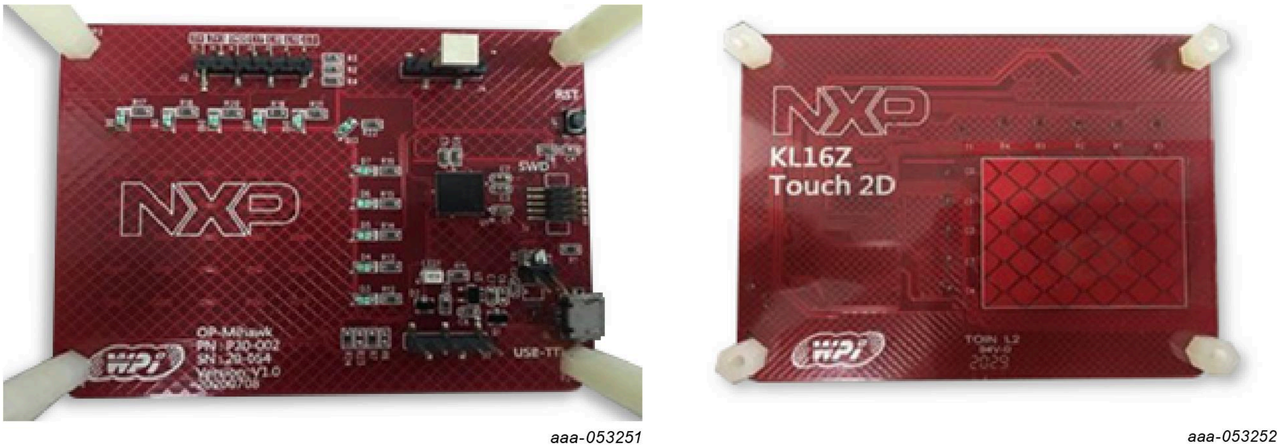

Figure 4. Touch 2D KL16Z board

Table 1. Pin connections

Touch 2D KL16Z board	MKL16Z pin	Pin mux
R0	PTA1	ALT0, TSI0_7
R1	PTA2	ALT0, TSI0_6
R2	PTB0	ALT0, TSI0_0
R3	PTB1	ALT0, TSI0_3
R4	PTB2	ALT0, TSI0_2
C0	PTB3	ALT0, TSI0_8
C1	PTB16	ALT0, TSI0_9
C2	PTB17	ALT0, TSI0_10
C3	PTC0	ALT0, TSI0_13
C4	PTC1	ALT0, TSI0_14
Shield	PTC2	ALT0, TSI0_15
UART0_RX	PTD6	ALT3, UART0_RX

Table 1. Pin connections*...continued*

Touch 2D KL16Z board	MKL16Z pin	Pin mux
UART0_TX	PTD7	ALT3, UART0_TX
I2C0_SDA	PTE18	ALT4, I2C0_SDA
I2C0_SCL	PTE19	ALT4, I2C0_SCL

4 2D gesture recognition algorithm

The 2D gesture recognition algorithm starts from detecting the movement of touch position first. To detect the movement, a FIFO queue of the calculated position is used to keep the most recent two positions. Once a new position is pushed into the queue, the oldest one is kicked out, so there is a previous position and a current position. We can make a difference from the current position to the previous one to check the movement. It was easy to do the judgment when the movement is in 1-dimension, as there are only two directions. Here for the 2D situation, we must open a little mind and make the extension.

When we do the difference between positions in a 2D position system, the X value minus X value, Y value minus Y value. Then comparison between the delta X and delta Y to find the one with the bigger absolute value. The direction of the bigger one represents the movement direction of the position in the available range of up/down/left/right. If the bigger one is still less than a given threshold, the behavior would be marked as staying the same position, with no movement.

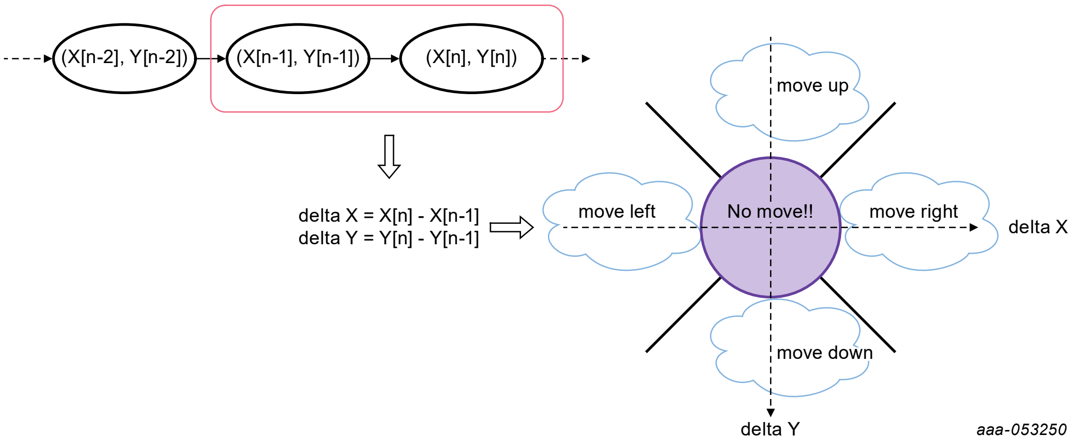

Figure 5. Algorithm to judge position movement in a 2D coordination system

Until now, we have three features for a sample of position:

- On touch or no touch
- Movement direction if on touch
- Timing relationship of successive samples

These features are used to judge the events on the touch panel, including:

- Move up/down/left/right, or stay the same place (no move).
- Touch 1st/2nd/3rd click, the interval between clicks is in a given short period.
- No touch keep and no touch interval are the state for no touch conditions.

Each event is triggered from the previous one under the indicated conditions. The event transmit machine between are as shown in Figure 6.

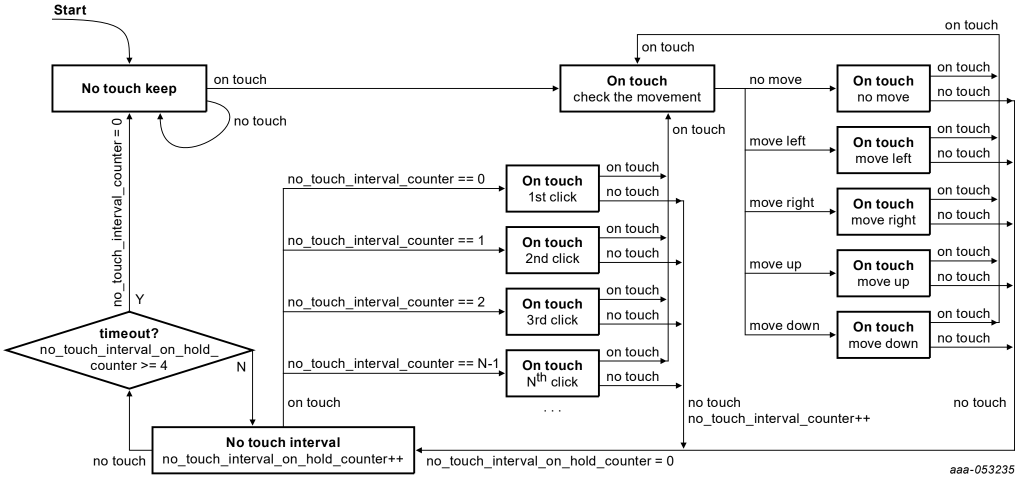

Figure 6. Event transmit machine for a 2D touch gesture recognition

In the application project, the sample rate for a group of all channels is 25 Hz, which is controlled by an LTPMR0 module. After each sensing period, the `touch_2d_calc_position()` and `touch_2d_gesture_process()` functions are called periodically for each group of new sensing values. The `touch_2d_calc_position()` function gets the information whether the panel is on touch and the position. The `touch_2d_gesture_process()` function fills these information into the event transmit machine and outputs the judgment of event for gestures.

5 Customizing

Some characters can be manually customized on the experience and requirement.

5.1 Timeout period of continuous click interval

The current time period of a continuous click interval is about 200 ms. This value is controlled by the setting for the interrupt period of LPTMR0, 40 ms, and the `no_touch_interval_on_holder_counter`. Every 40 ms, the `touch_2d_gesture_process()` function is called again, and the `no_touch_interval_on_holder_counter` variable is increased once when the current state is for the event of No Touch Interval. It is used to compare with the number 4 (0 is the first one, 4 means the fifth period, for 5 × 40 ms = 200 ms), as shown in Figure 6, to check whether the interval is timeout. If the coming on touch condition caused by the click arrives before timeout, the new click is considered in a successive/continuous click sequence, and marked as an Nth click event. If no on touch condition comes before the timeout (no touch or later touch), the No Touch Interval state (causing the No Touch Interval event) is transmitted to the No Touch Keep state (causing the No Touch Keep event), which means the current state is the No Touch stably, and no continuous click detection is active for the following samples.

5.2 Enable the detection of event for more continuous clicks

In the current example project, only the first three clicks in a continuous click sequence are activated. However, more clicks can be recognized within the algorithm as well. It is controlled by the available values for the `no_touch_interval_counter` variable. This variable records the count of intervals in a continuous click sequence, and increases when a new click coming. Once the continuous click event happens, the `touch_2d_gesture_process()` function checks the value of this variable and reports the related continuous click events, including for the first three clicks. The detection for more clicks is still available, as the no touch `interval_counter` keeps increasing for the new click and does not reset until the continuous click sequence is broken (the interval is timeout).

For example, to pick out the eighth click, only the following addition code is needed:

```
touch_2d_gesture_event_t touch_2d_gesture_process(int32_t x_pos, int32_t y_pos,
 bool on_touch)
{
    if (on_touch)
    {
        if ( (touch_2d_gesture_event_pre ==
 eTouch_2d_gesture_event_no_touch_keep)
                || (touch_2d_gesture_event_pre ==
 eTouch_2d_gesture_event_no_touch_interval)
            )
    {
        if (touch_2d_gesture_no_touch_interval_counter == 0u)
        {
            touch_2d_gesture_event = eTouch_2d_gesture_event_on_touch_just;
        }
        else if (touch_2d_gesture_no_touch_interval_counter == 1u)
        {
            touch_2d_gesture_event =
 eTouch_2d_gesture_event_on_touch_just_2nd_click;
        }
        else if (touch_2d_gesture_no_touch_interval_counter == 2u)
        {
            touch_2d_gesture_event =
 eTouch_2d_gesture_event_on_touch_just_3rd_click;
        }
        else if (touch_2d_gesture_no_touch_interval_counter == 7u)
        {
            touch_2d_gesture_event =
 eTouch_2d_gesture_event_on_touch_just_8th_click;
        }
        else
        {
            touch_2d_gesture_event =
 eTouch_2d_gesture_event_on_touch_no_process;
        }
        ...
```

5.3 Support "all cover" gesture

The current algorithm supports the gestures based on the movement of the one-touch point. The all cover gesture is not just using one position. It is considered as a special case before calculating the position, as its judgment is not based on the current position algorithm.

We use the sum of different offsets for all channels in the touch panel as the critical character to check the all cover event. Once a channel is touched, its sensing value causes an offset against its baseline of no touch. This offset is the difference for the touch event of the channel. The closer the finger is near the touch pad, the bigger the difference value would be. In the self-capacitive touch sensing method, which is used in the touch module of KL16Z now, these differences for each channel are independent. So, when multiple channels are touched together in the same time, they would have a big difference together. Then the sum of these differences can be increased significantly than the one in the no-touch state. So, it can be used to judge how many multiple channels are touched at the same time, as more touch area would cover more channels.

Actually, its implementation is in the `touch_2d_calc_position()` function, and returns the code 3 when the all cover event is detected. For details about the coding, see AN13026SW.

6 Note about the source code in the document

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

7 Revision history

Table 2 summarizes the revisions to this document.

Table 2. Revision history

Revision number	Release date	Description
1	09 November 2023	Text update and images updated to svg format
0	October 2020	Initial public release

Legal information

Definitions

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

Disclaimers

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Trademarks

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

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

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Contents

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

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

Date of release: 9 November 2023
Document identifier: AN13026