TLE9893_2QKW62S_BL_AND_APP

About this document

Scope and purpose

The aim of this guide is to present the scope, the implementation, the algorithm and a demonstration of the **TLE9893_2QKW62S_BL_AND_APP** examples code for the TLE989x Infineon Embedded Power ICs based on Arm® Cortex® M3. This example code can be found in the Keil µVision Pack Installer.

The full functionalities and characteristics of the embedded power devices are described in the datasheets and user's manual. Please refer to these documents for more detailed information. Furthermore, a low level (line-by-line) description of the code is not the aim of this document, although occasionally some code blocks might be reported if necessary to the comprehension.

Note: *The following information is given as a hint for the implementation of the system only and shall not be regarded as a description or warranty of a certain functionality, condition or quality of the referred devices or presented software example.*

Intended audience

Design engineers, system engineers, embedded power designers

Table of contents

1 Introduction

This AppNote describes the concept of the two examples

- BL_AND_APP_BOOTLOADER_EXAMPLE
- BL_AND_APP_APPLICATION_EXAMPLE

This set of examples demonstrates the concept of bootloader (BL) and application (App) and how the two Flash blocks of the device can be used to realize this concept.

In this set of examples, two separate projects are setup, one for the bootloader and one for the application. Both executables for both bootloader and application are loaded in different flash memories. The bootloader is in FLASH0, the application is in FLASH1 (cached).

On power on reset, the Bootloader is executed, and within an endless main loop, a tick count is incremented. If the tick reaches 100.000, the pin P0.1 is toggled and the LED is switched on or off. Then the tick count is reinitialized to 0.

If MON1 is pressed while the bootloader is being executed, the bootloader configures the required settings and then starts the application.

The application also has an endless loop with a tick counter. The counter is incremented, and if the tick reaches 100.000, pin P0.2 is toggled and the LED is switched on or off. Then the tick count is reinitialized to 0.

If MON2 is pressed while application is being executed, the device will reset and it starts execution of bootloader.

Note: the LED at P0.2 blinks faster when the application is executed, compared to the blinking of LED when bootloader is executed. This is because the application is executed from FLASH1 with the Cache enabled. The faster blinking of the LED shows the performance increase due to the cache.

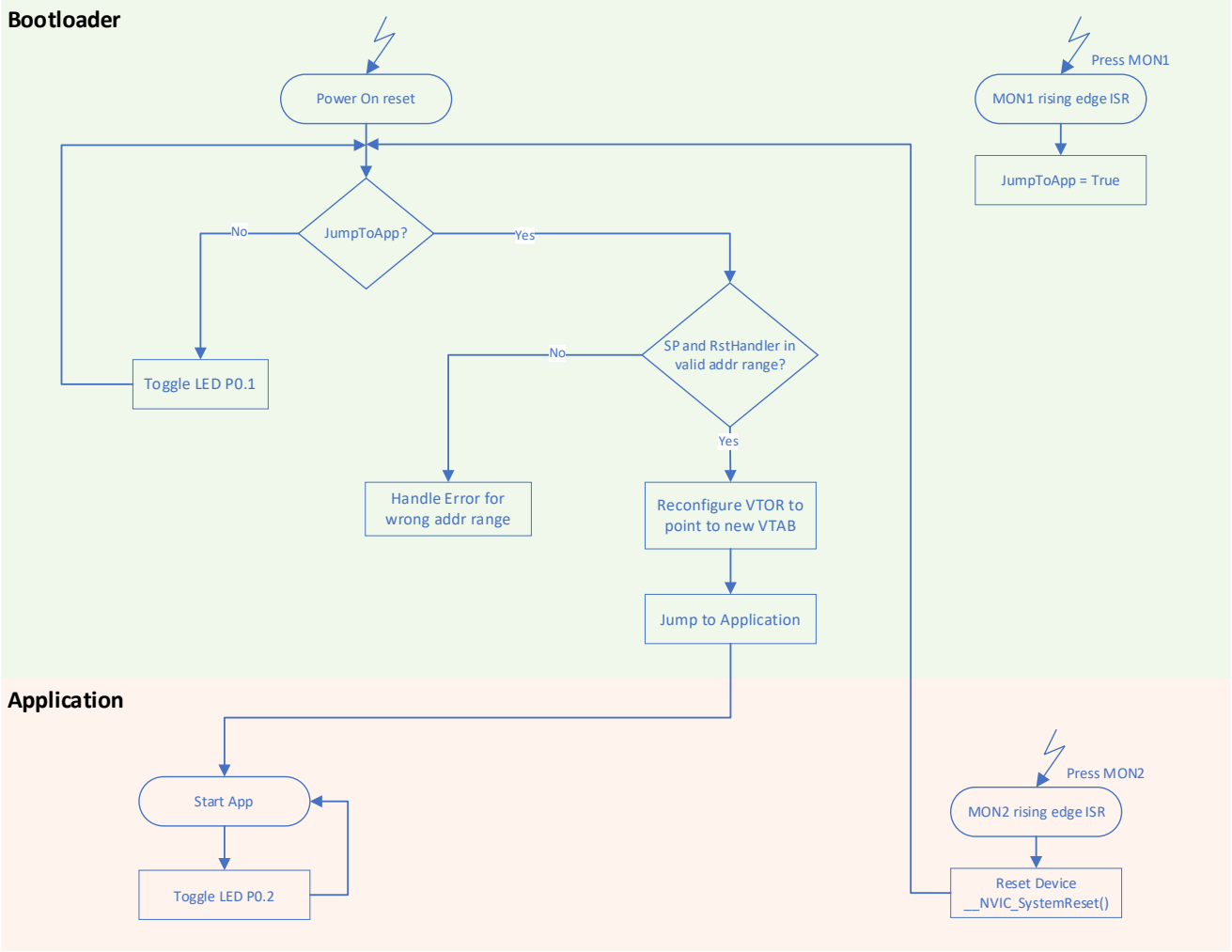

Figure 1 TLE9893_2QKW62S_BL_AND_APP example concept

2 Hardware

This chapter shows how to run the TLE9893_2QKW62S_BL_AND_APP examples with the TLE988x/TLE989x evaluation board. For this the project must be opened and compiled.

Figure 2 shows the TLE988x/TLE989x evaluation board. The application code must be loaded via a debugger (e.g. ULINK or J-Link) to the board. The board must be powered with 12V (red and black connections).

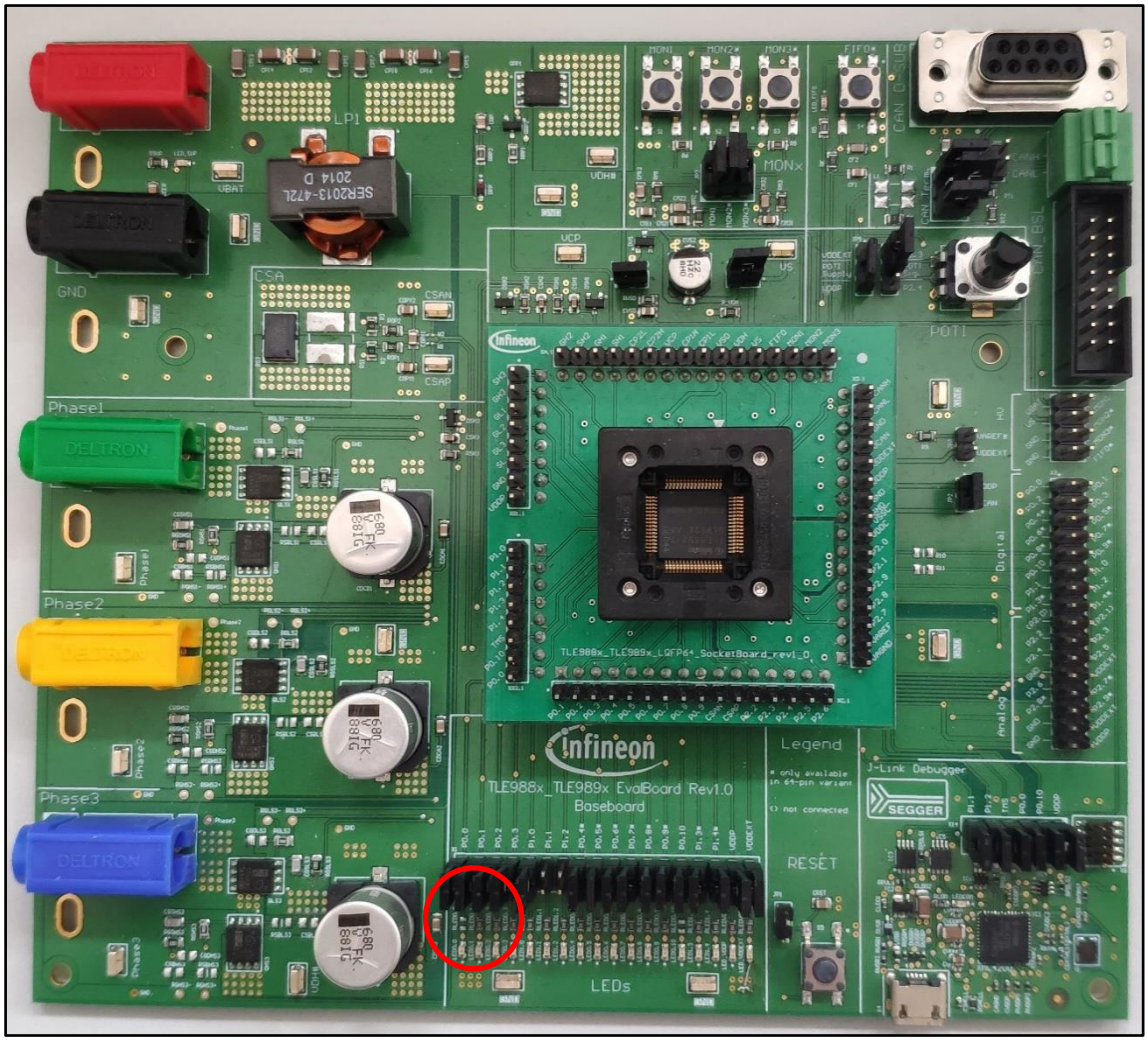

Figure 2 TLE988x/TLE989X evaluation board

3 Implementation

This chapter shows the process to follow to get a working bootloader and application example set.

3.1 Get the example via the Pack Installer for Keil

Open the Pack Installer within the Keil IDE. See Figure 3 below.

Choose the appropriate device (here TLE9893_2QKW62S) on the left-hand side. On the right-hand side, select the tab Examples, where you can access the TLE9893_2QKW62S_AND_AND_APP_BOOTLOADER and TLE9893_2QKW62S_BL_AND_APP_APPLICATION examples.

Clicking on “Copy” will copy the examples on your computer and open them.

Both examples need to be flashed to the target.

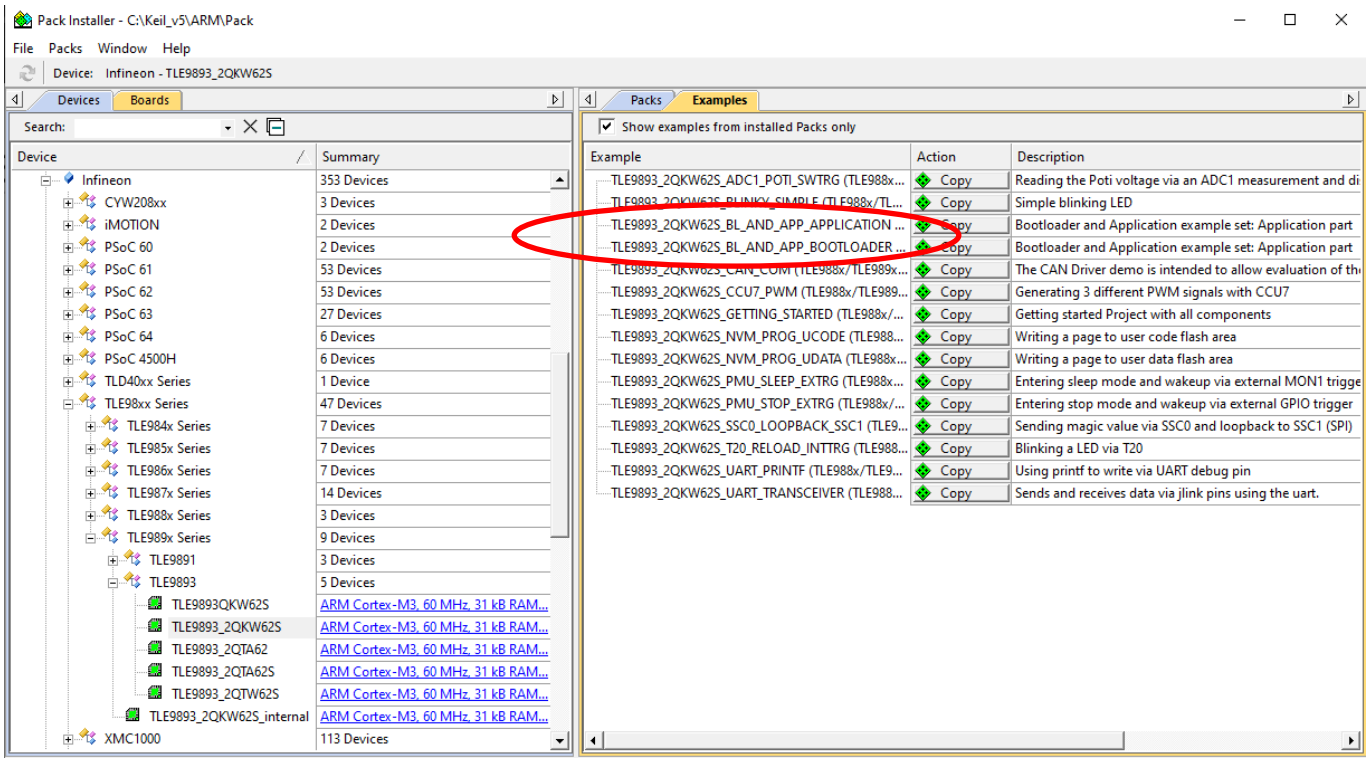

Figure 3 Keil Pack Installer

3.2 Configuration

In order to see the configured settings, start the tool Config Wizard. It is available within the Keil IDE through a shortcut in the Tools menu.

The Config Wizard opens and shows an overall status of the current pin configuration.

Settings in the bootloader project:

- P0.1 is used and configured as output
- MON1 Rising edge interrupt is configured

Settings in the application project:

- P0.2 is used and configured as output

- MON1 Rising edge interrupt is configured

3.3 Settings for loading both examples to the device

The executables for both projects will be flashed in different flash memories. TLE9893_2QKW62S_BL_AND_APP_BOOTLOADER will be loaded to FLASH0, while TLE9893_2QKW62S_BL_AND_APP_APPLICATION will be programmed to FLASH1 memory.

Following settings are required to flash the both executable to correct memory segment (Keil uVision).

Settings in the bootloader project:

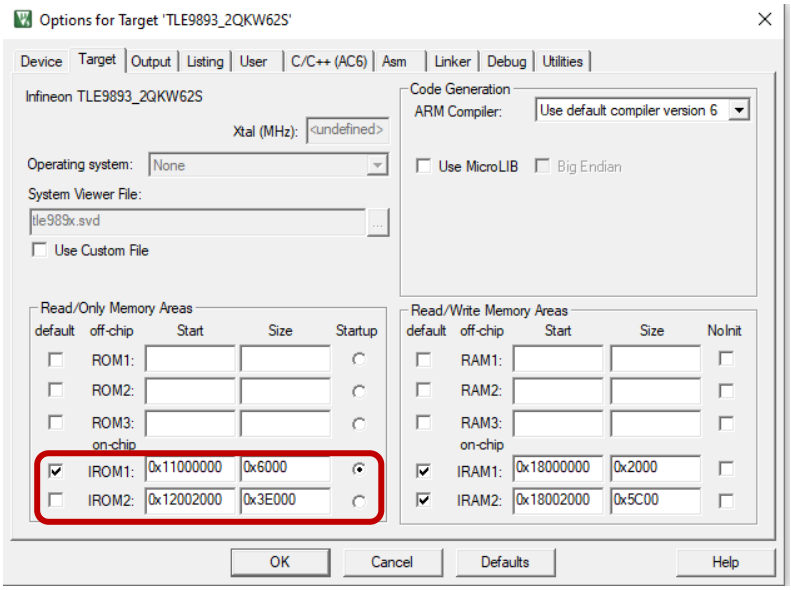

Figure 4 settings for TLE9893_2QKW62S_BL_AND_APP_BOOTLOADER in Keil uVision

Settings in the application project:

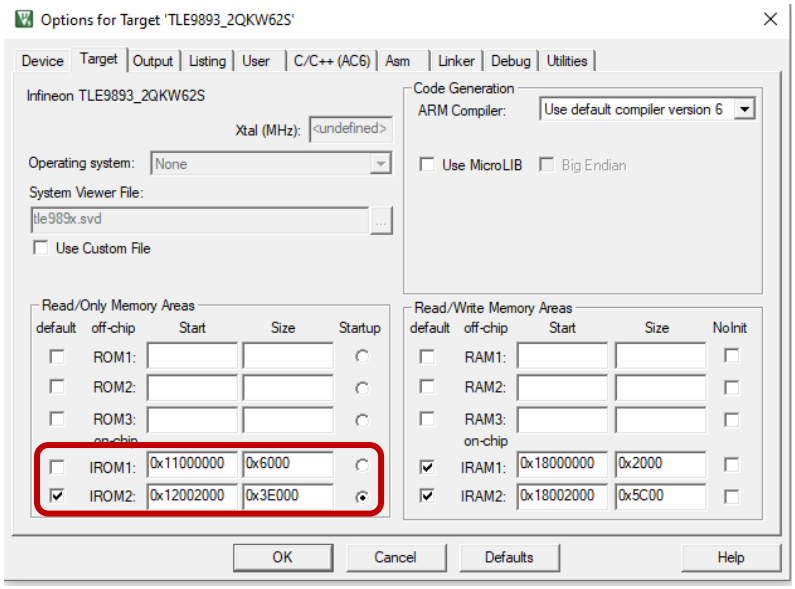

Figure 5 settings for TLE9893_2QKW62S_BL_AND_APP_APPLICATION in Keil uVision

In IAR, a custom linker file has to be added which only uses FLASH0 (bootloader) or only FLASH1 (application). This is already done in the examples; the linker files can be seen in “Options for node” --> “Linker”. An example of the bootloader can be seen in the figure below.

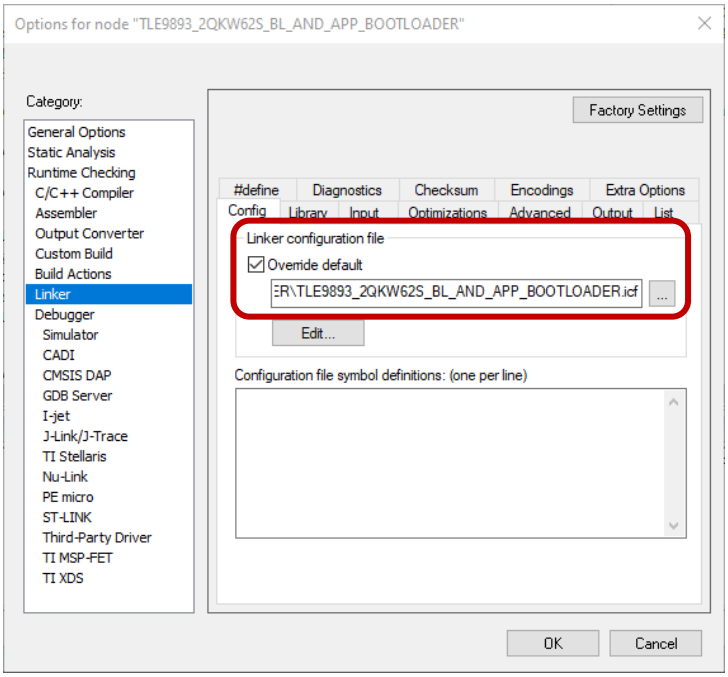

Figure 6 settings for TLE9893_2QKW62S_BL_AND_APP_BOOTLOADER in IAR

In this example have to be flashed one after other, order does not matter. It important that while flashing the second example “Erase Full Chip” must not be selected to not erase the other example.

Following figure shows this setting in Keil uVision:

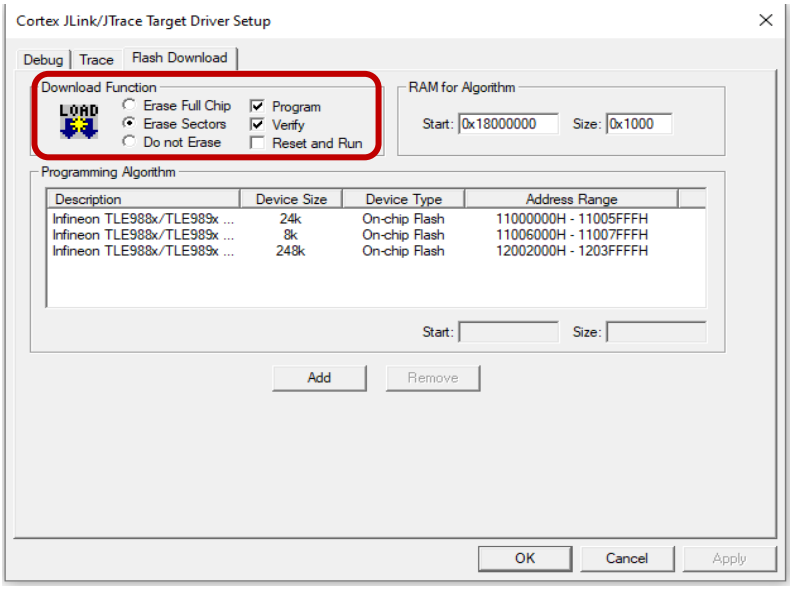

Figure 7 Settings for erasing the device for download in Keil uVision

3.4 Sample code for bootloader

Figure 8 shows the application code for LED blinking operation and jumping to the application program when MON1 is pressed. The watchdog service is executed in order to avoid any stall during the execution.

During each loop, the `u32_tick` value is incremented. If the `u32_tick` value reaches 100.000 here the `GPIO_setP01State` call is executed. This is an available API call which toggles the pin P0.1 through the enum argument `GPIO_STATE_TOOGLE`. Toggle means that the pin is set to low if it was high and vice versa. Once the call is done the `u32_tick` value is reinitialized to 0.

If MON1 is pressed, then b_jumpToApplication is set to 1, in case that some checks are done to make sure that stack pointer and reset handler for application are within valid range of RAMs and FLASH1 respectively.

After that VTOR is re configured to point to the vector table new application. And Finally, program execution is jumped to the application programmed in FLASH1.

```
for (;;)
{
  /* Main watchdog service */
  (void) PMU_serviceFailSafeWatchdog();

  u32_tick++;
  if (u32_tick > 100000)
  {
    GPIO_setP02State(GPIO_STATE_TOGGLE);
    u32_tick = 0;
  }

  if(b_jumpToApplication == true)
  {
    /* Check if stack pointer is in one of the 2 RAMs and reset handler is within Flash11 range */
    if(Address[0] >= RAM_START_ADRESS &&
       Address[0] < RAM_END_ADDRESS &&
       Address[1] >= FLASH1_START_ADDRESS &&
       Address[1] < FLASH1_END_ADDRESS)
    {
      /* Service Watchdog */
      PMU_serviceFailSafeWatchdogSOW();

      /* Disable all interrupts */
      __disable_irq();

      /* point VTOR to new vector table */
      CPU->VTOR.reg = USER_APPLICATION_VTAB_ADDRESS;

      /*Jump to new application */
      BootJumpASM( Address[ 0 ], Address[ 1 ] ) ;
    }
    else
    {
      /* Error hadling in case of wrong stack pointer or reset handler address */
    }
```

Figure 8 TLE9893_2QKW62S_BL_AND_APP sample code

References

See the code examples at **www.infineon.com**

Revision history

Document version	Date of release	Description of changes
1.0	2021-01-30	Initial version
1.1	2022-10-13	Editorial changes

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-10-13

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.