# SMU_Fault_Signaling_1 for KIT_AURIX_TC297_TFT

## Fault signaling via a port pin

AURIX™ TC2xx Microcontroller Training
V1.0.0

# Scope of work

**The Fault Signaling Protocol (FSP) pin is configured by the SMU to signal a fault by changing the state of the port pin P33.8.**

During initialization the FSP is set to fault free state, indicated by a high level of the port pin P33.8. The software triggers an alarm that leads to the FSP fault state, thus the port pin P33.8 switches to low level state.
After one second from triggering the alarm, the FSP release command is sent by software. The port pin P33.8 state switches to high level again one second after the command is sent, since the minimum fault state time is configured to two seconds.

# Introduction

- The Safety Management Unit (SMU) is a central and modular component of the safety architecture providing a generic interface to manage the behavior of the microcontroller under the presence of faults.
- The SMU centralizes all the alarm signals related to the different hardware and software-based safety mechanisms.
- The SMU implements a Fault Signaling Protocol (FSP) reporting internal faults to the external environment.
- Each individual alarm can be configured to activate the fault signaling protocol.
- Once configured, the FSP protocol drives the P33.8 port pin to signal the internal state.

# Introduction

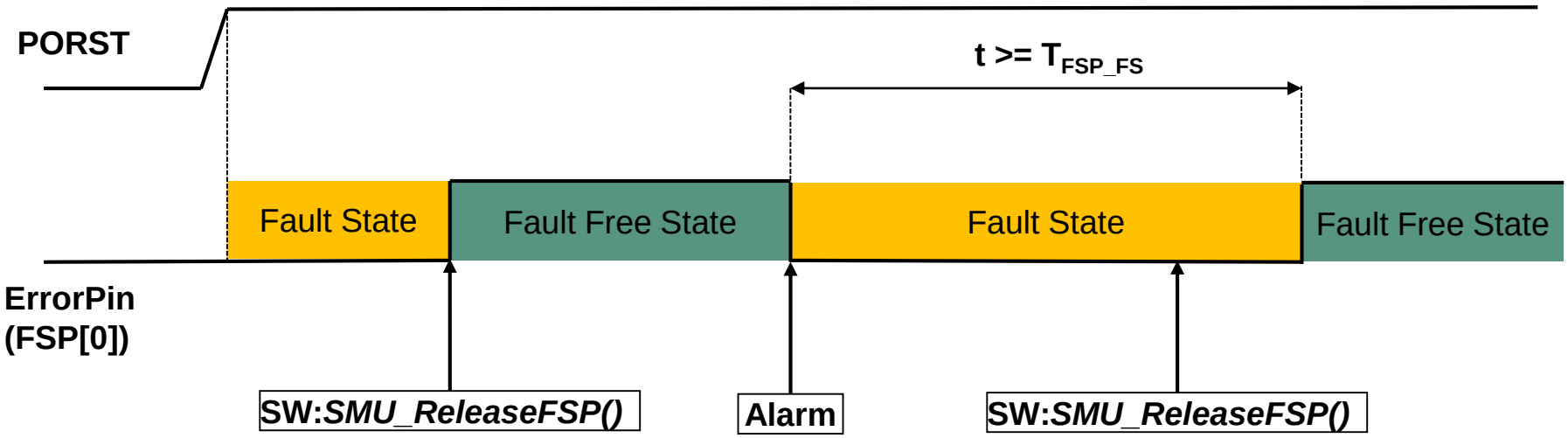
- The FSP has three states:
  - The **Power-on Reset** state
    After Power-on Reset the SMU is disconnected from the ports and the SMU FSP output shall be the Fault State.
  - The **Fault-free State**
  - The **Fault State**

- The FSP can be configured using the following modes:
  - **Bi-stable fault signaling protocol**: single pin output (push-pull active low configuration using FSP[0]), also called Error Pin.
  - **Time switching protocol**: single-bit timed protocol using FSP[0]

**Note**: FSP[0] is the bit 0 of the SMU_STS.FSP bit field, it reflects the state of the ErrorPin (port pin P33.8)  driven to the external environment.

# Introduction

- **Bi-stable fault signaling protocol**:

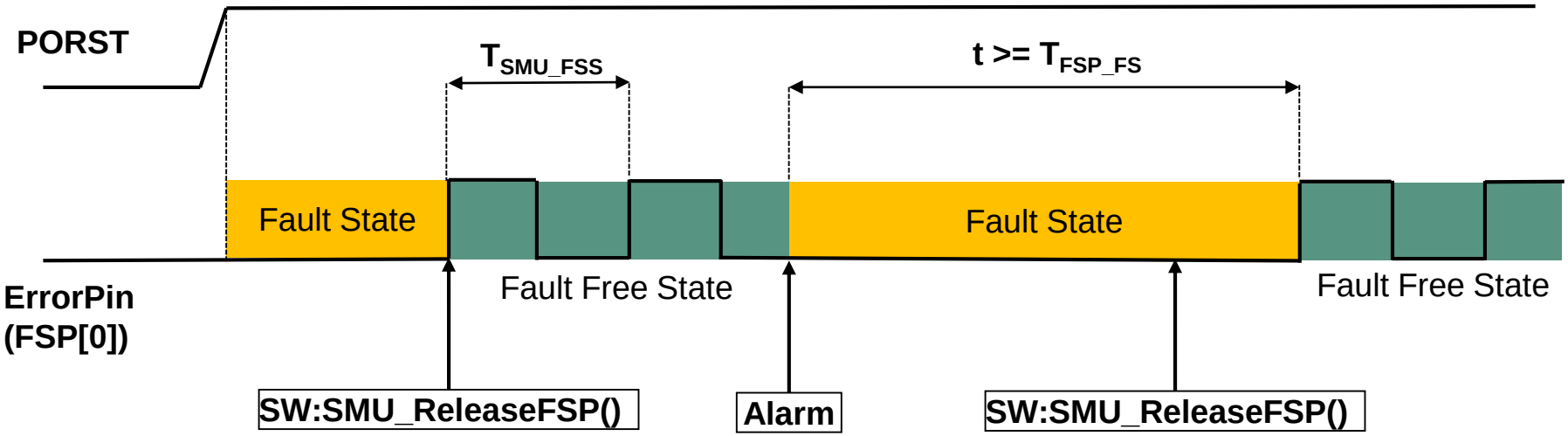



- After Power-on Reset FSP[0] = 0, FSP is in the fault state
- After configuration, the SW must set the fault free state FSP[0] = 1
- If an alarm, which is configured to trigger the FSP, is detected, the FSP immediately enters the fault state, FSP[0] = 0, until SW calls ***SMU_ReleaseFSP()*** and sets the fault free state

**NOTES:**

- After an Alarm, if an ***SMU_ReleaseFSP()*** command is received within t < $\mathbf{T_{FSP\_FS}}$, the command is logged and automatically executed when $\mathbf{T_{FSP\_FS}}$ is reached.
- The $\mathbf{T_{FSP\_FS}}$ is the minimum Fault state period (> 250 us), configured via the SMU_FSP register.

## Time switching protocol:

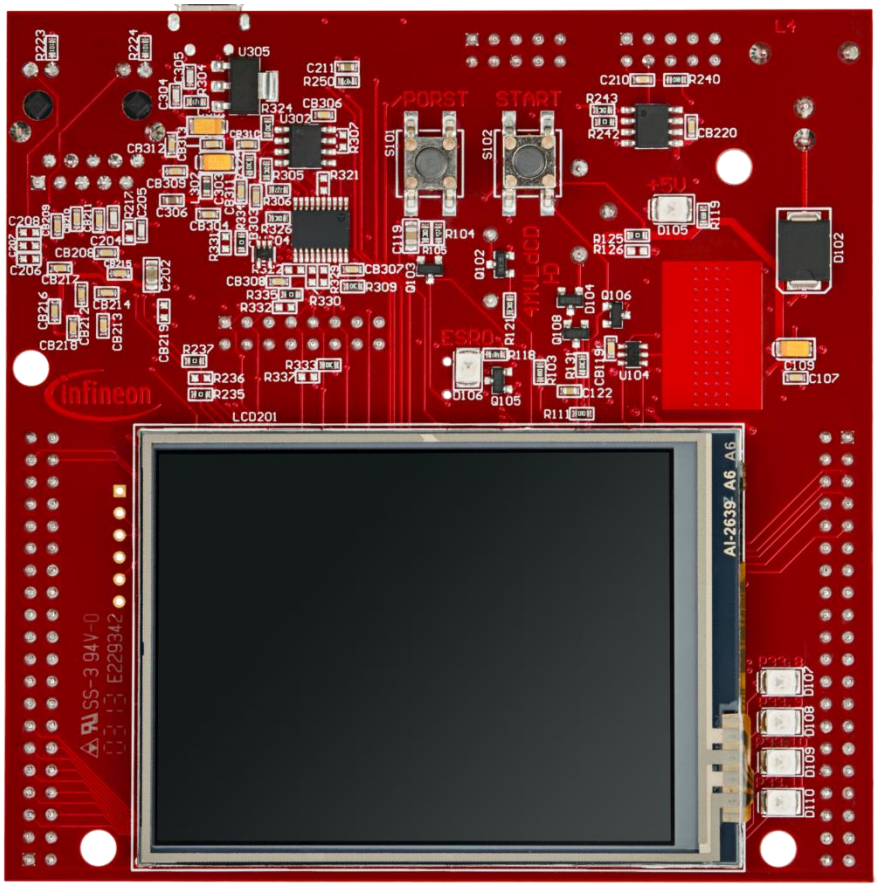



Same functionalities as the **Bi-stable FSP** (refer to previous slide), with only one difference:

- In the fault free state, FSP[0] oscillates between logic level 0 and logic level 1 with the period $T_{SMU\_FSS}$ (duty cycle 50%) configured via the SMU_FSP register.

# Hardware setup

This code example has been developed for the board KIT_AURIX_TC297_TFT_BC-Step.



# Implementation

**FSP initialization:**

- The initialization of the FSP protocol is done by defining an instance of the ***IfxSmu_configFsp*** structure.
- The structure is filled with default values by the function ***IfxSmu_initFspConfig()***.
- Afterwards, the following parameters are modified to enable fault to run state transition and to set a minimum fault state period:
  - ***enableFaultToRunStateTransition*** is set to ***TRUE***
  - ***faultStateTime*** is set to 2 seconds
- The function ***IfxSmu_initFsp()*** is used to initialize the FSP protocol.
- The function ***IfxSmu_releaseFsp()*** switch the FSP state to fault free state.

**Note:** After Power-on-Reset (PORST) the FSP is in fault state.

The above functions can be found in the iLLD header ***IfxSmu.h***.

# Implementation

**Alarm Configuration:**

- The SMU configuration is protected against corruption. That is why, it is mandatory to unlock it before the configuration process using the ***IfxSmu_unlock()*** function.
- The Software Alarm 0 is configured to trigger the FSP entering the fault state. The ***IfxSmu_enableAlarmFsp()*** function ensures this functionality.
- Then the SMU configuration is locked again using the ***IfxSmu_lock()*** function.

**Start the SMU state machine:**

- The SMU state machine is launched by calling the ***IfxSmu_start()*** function.

The above functions can be found in the iLLD header ***IfxSmu.h***

# Implementation

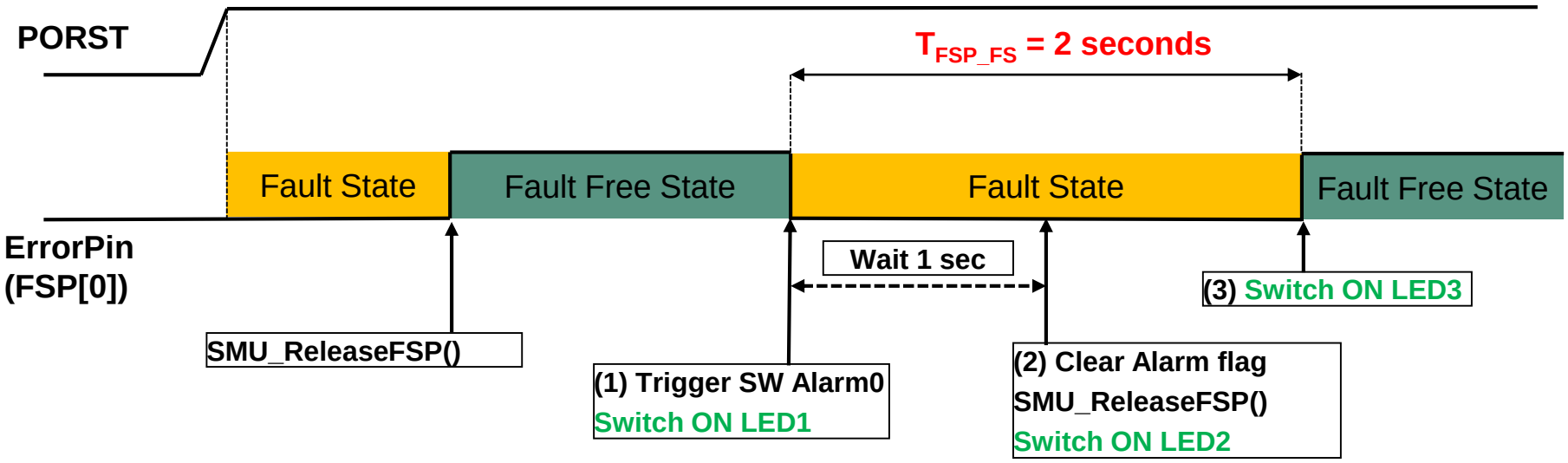
**Training scenario, after FSP and SMU initialization:**

1. The Software Alarm 0 is triggered, using ***IfxSmu_triggerAlarm()*** and the FSP enters the fault state.
2. After one second:
   - The alarm flag is cleared using ***IfxSmu_enableClearAlarmStatus()*** and ***IfxSmu_clearAlarm()***
   - The FSP release command is sent using ***IfxSmu_releaseFsp()***
3. The FSP enters the **fault free state** after exactly two seconds from triggering the Alarm, because $T_{FSP\_FS}$ is set to 2 seconds.

**Notes:**

› FSP mode is the **Bi-stable** mode (default FSP configuration)

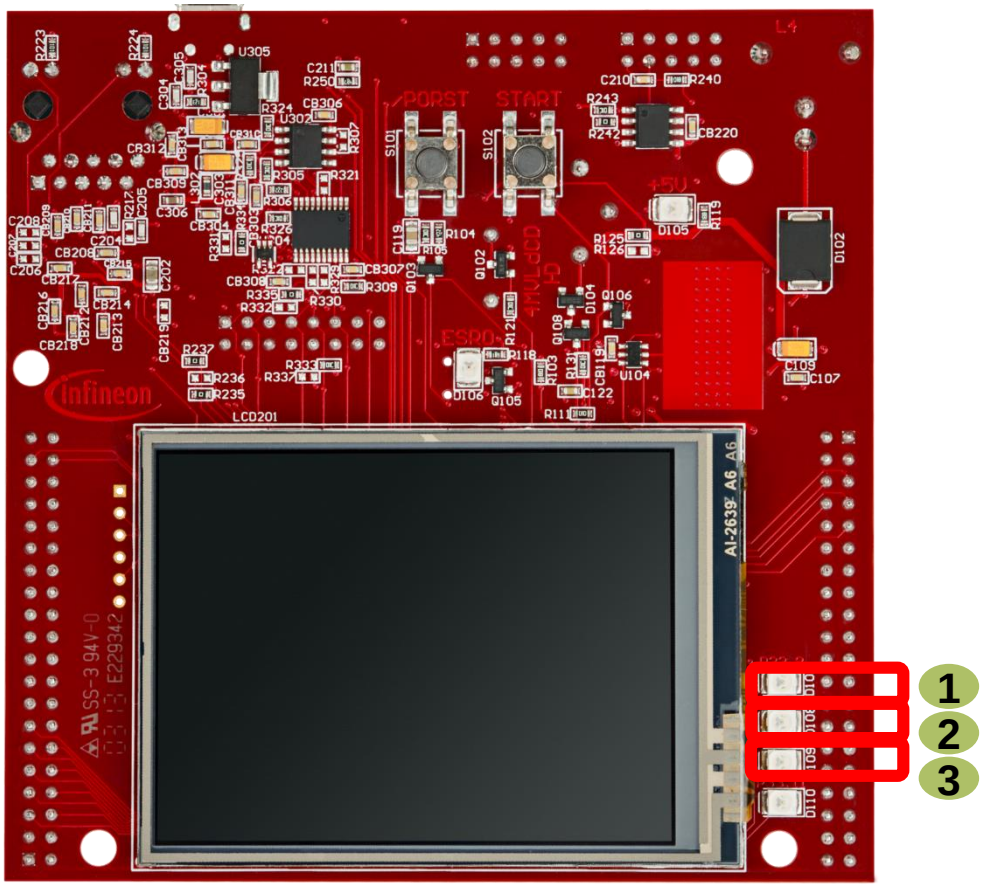
› The **Minimum** fault state period: Two seconds (refer to FSP initialization)



# Run and Test

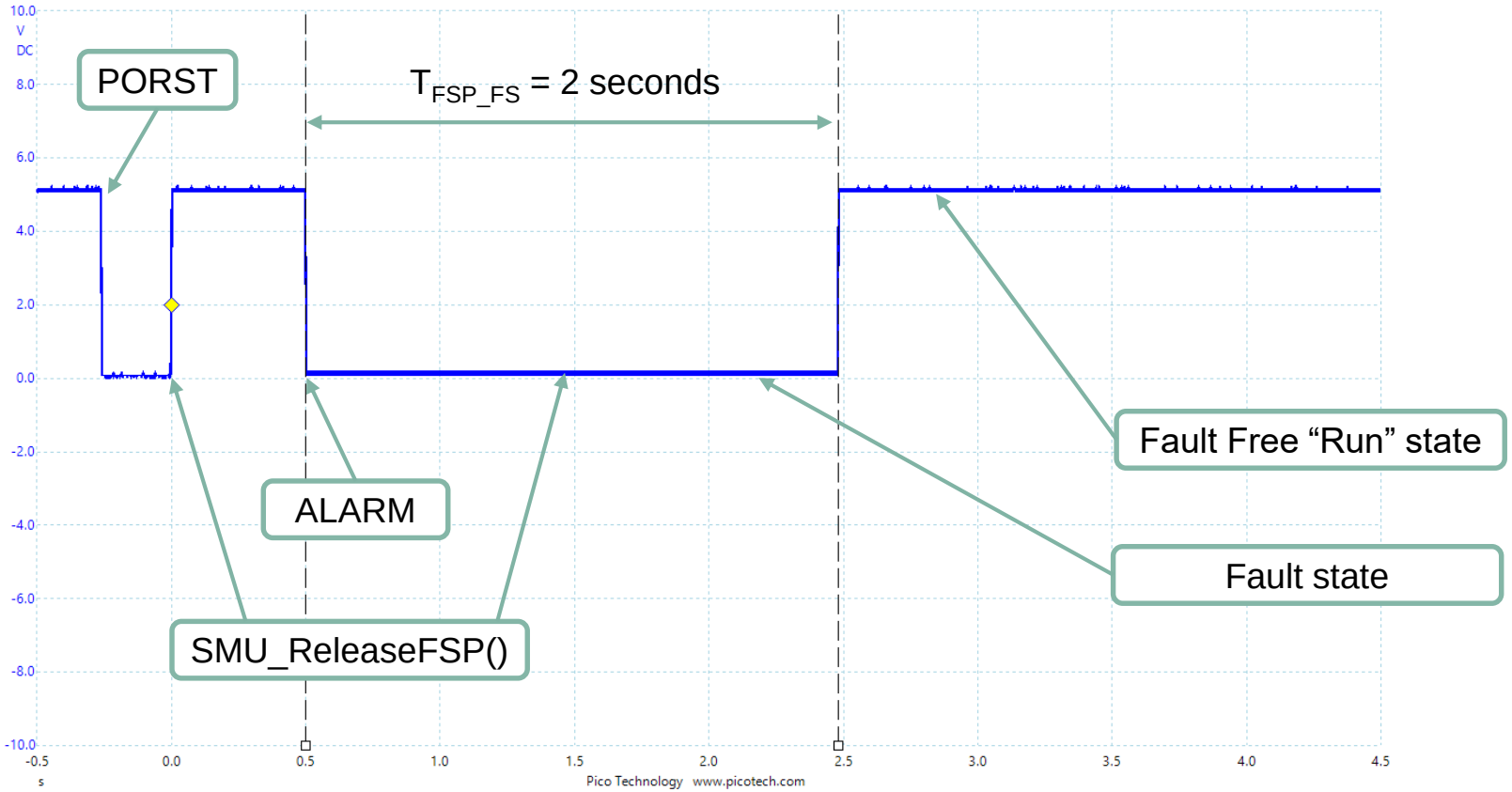
After code compilation and flashing the device, check the behavior of ***LED1, LED2*** and ***LED3***:

- **LED1** switches ON when the Alarm is triggered
- **LED2** switches ON after one second, when the FSP release command is sent
- **LED3** switches ON when the FSP enters the fault free state, two seconds after the alarm is triggered



# Run and Test

Using an oscilloscope, it is possible to observe the ErrorPin (P33.8) status, which is driven by the FSP protocol:



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

**Edition 2020-02**
**Published by**
**Infineon Technologies AG**
**81726 Munich, Germany**



**Do you have a question about this document?**
**Email: erratum@infineon.com**

**Document reference**
**SMU_Fault_Signaling_1_KIT_TC297_TFT**

**IMPORTANT NOTICE**
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie") .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, 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.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

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.