

# Watchdog\_1 for KIT\_AURIX\_TC397\_TFT

## Watchdog configuration

AURIX™ TC3xx Microcontroller Training  
V1.0.2



[Please read the Important Notice and Warnings at the end of this document](#)

# Scope of work

---

**The watchdog is serviced in order to prevent a reset.**

This example is showing how to service the WatchDog Timer (WDT) periodically.

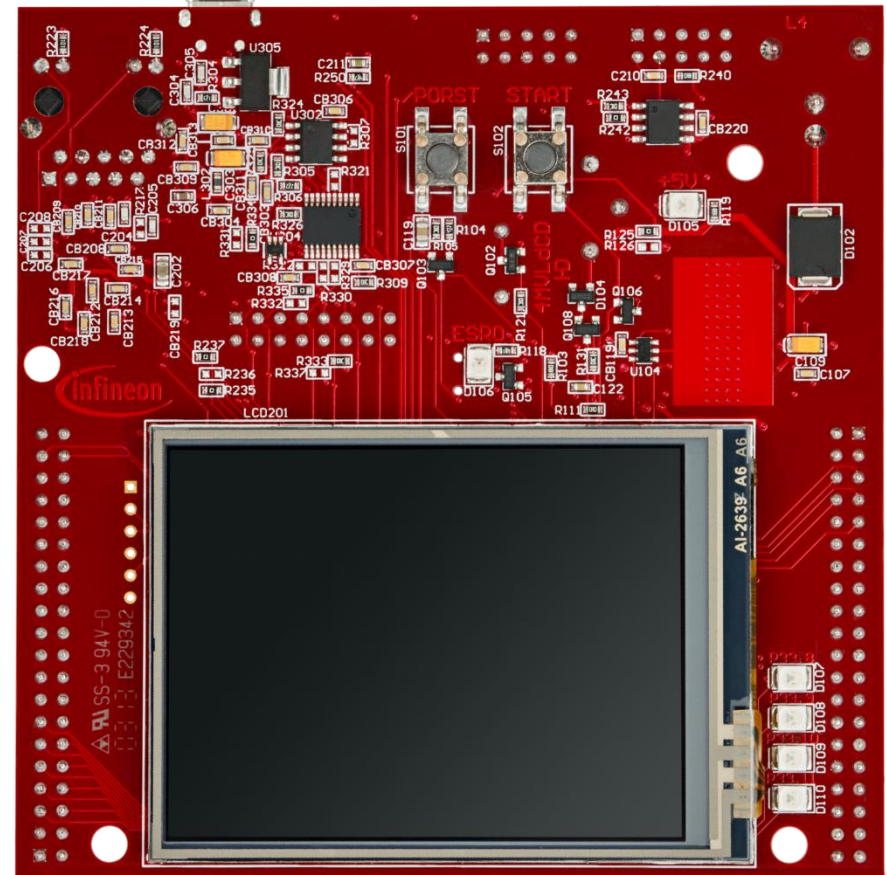
# Introduction

---

- › The watchdog (WD) provides on the one hand a highly reliable and secure way to detect and recover from software or hardware failure and on the other hand a protection against unintended register write accesses
  
- › There are two types of Watchdogs:
  - The Safety WD provides protection against unintended writes to critical system registers and memories. It consists of a timer that causes an SMU alarm request if it is not serviced within a certain time interval
  
  - The individual CPU WDs have similar functionalities as the safety watchdog. They provide protection to CPU registers as well as to particular system registers. The CPU individual timers, if activated, offer the ability to monitor separate CPU execution threads. They have to be serviced similar to the safety watchdog timer within a certain time interval, otherwise a device reset is triggered

# Hardware setup

This code example has been developed for the board  
KIT\_A2G\_TC397\_5V\_TFT.



# Implementation

## Configuring and servicing the CPU 0 watchdog timer

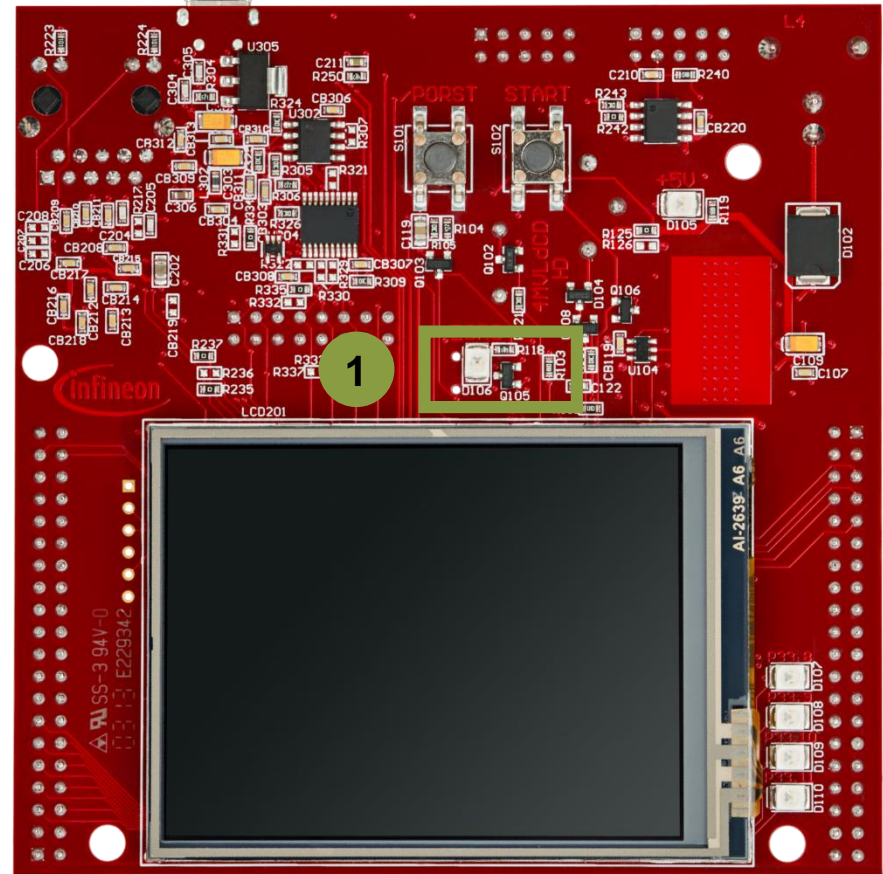
- › Comment out the instruction to disable the CPU watchdog (***IfxScuWdt\_disableCpuWatchdog()***). Thus the CPU0 watchdog remains on and needs to be serviced
- › Configure the reload value (REL) that specifies the timeout of the WD timer using the function ***IfxScuWdt\_changeCpuWatchdogReload()***. The default value is 0xFFFC and the timer counts up. The time window for servicing the WD is very short, therefore it is extended by using 0xE000 as REL
- › The timeout can be calculated using this formula:  
$$\text{timeout} \approx (0xFFFF - \text{REL}) / (f_{\text{SPB}} / \text{WDTSCON1.IRx}) = (0xFFFF - 0xE000) / (100 \text{ MHz} / 16384)$$
**Note:** The value 16384 is a constant, defined by IRx of register WDTSCON1.
- › A time variable is initialized to wait a fixed time inside the infinite loop
- › Call the ***IfxScuWdt\_serviceCpuWatchdog()*** function in order to service the watchdog timer
- › The CPU0 watchdog is serviced periodically every second within the infinite while loop

**Note:** The above related **ScuWdt** functions need a password in order to write into the CPU0 related watchdog registers. The password is retrieved using the function ***IfxScuWdt\_getCpuWatchdogPassword()***. The **ScuWdt** functions are part of the library ***IfxScuWdt.h***.

# Run and Test

After code compilation and flashing the device, perform the following steps:

- › Terminate the debug session if it is open and restart the device. Check whether LED D106 (1) is off - No reset is signaled. Thus the CPU0 watchdog is well serviced periodically
- › Comment out the instruction to service the CPU WD:  
***lfxScuWdt\_serviceCpuWatchdog()***
- › Compile, flash and run the code again
- › Terminate the debug session if it is open and restart the device. LED D106 (1) turns on after some time signaling a reset. Thus the WDT is not serviced anymore



**Note:** in order to use the debugger, OCDS is enabled. Whenever the OCDS is enabled, all watchdogs are disabled by default.

# References



- › AURIX™ Development Studio is available online:
- › <https://www.infineon.com/aurixdevelopmentstudio>
- › Use the „Import...“ function to get access to more code examples.



- › More code examples can be found on the GIT repository:
- › [https://github.com/Infineon/AURIX\\_code\\_examples](https://github.com/Infineon/AURIX_code_examples)



- › For additional trainings, visit our webpage:
- › <https://www.infineon.com/aurix-expert-training>



- › For questions and support, use the AURIX™ Forum:
- › <https://www.infineonforums.com/forums/13-Aurix-Forum>

# Revision history

---

Revision	Description of change
V1.0.2	Updated timeout formula with more details
V1.0.1	Updated Implementation slide according to code implementation
V1.0.0	Initial version



## Trademarks

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

**Edition 2021-03**

**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**

**© 2021 Infineon Technologies AG.**

**All Rights Reserved.**

**Do you have a question about this document?**

**Email: [erratum@infineon.com](mailto:erratum@infineon.com)**

**Document reference**

**Watchdog\_1\_KIT\_TC397\_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](http://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.