SCU_Reset_Detection_1 for KIT_AURIX_TC375_LK

Detection of reset type

AURIX™ TC3xx Microcontroller Training
V1.0.0

Scope of work

This example shows how to detect the source of the last reset (power-on reset, watchdog reset, etc.)

The AURIX™ TC3xx devices can be reset by various reset sources. The application software is able to determine the source of the last reset based on a routine that evaluates the related reset special function registers. According to the type of reset, one or two LEDs are switched on.

Introduction

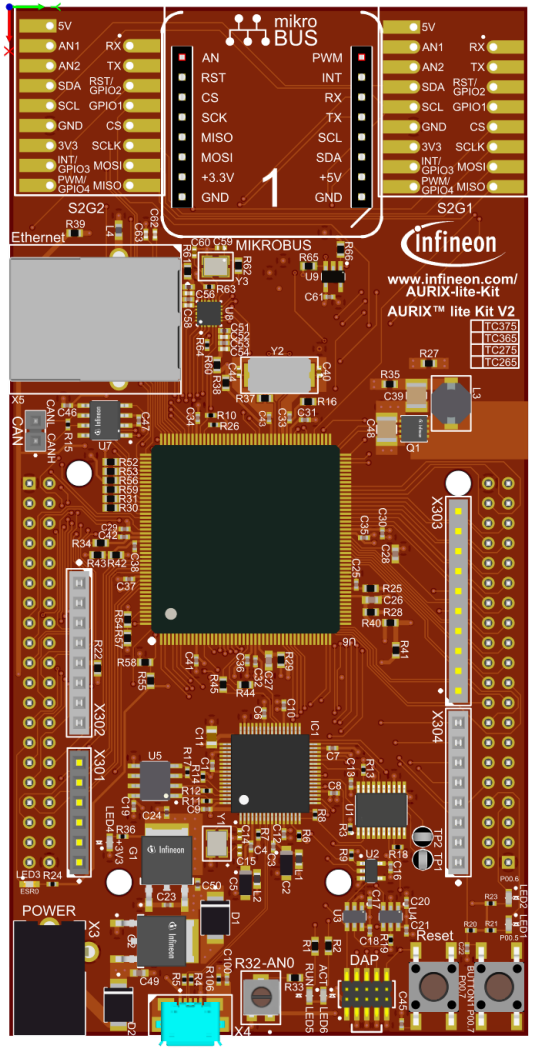
- Resets can be configured and determined in the Reset Control Unit (RCU), belonging to the System Control Unit (SCU)

- Various reset triggers are available, such as:
  - Supply Monitor
  - Embedded Voltage Regulators (EVRs)
  - Power-On Reset (PORST)
  - External Service Request (ESRx)
  - Debug interface

- Consequently, different reset types can be derived, such as:
  - Cold-/Warm-Power-On Reset
  - System Reset
  - Application Reset
  - Debug Reset
  - Module Reset

Hardware setup

This code example has been developed for the board KIT_A2G_TC375_LITE.



Implementation

Reset Detection

To get information about the last occurred reset, the function ***IfxScuRcu_evaluateReset()*** is called inside ***detectResetSource()***. The returned value is a data structure comprising elements such as ***resetType*** and ***resetTrigger***.

- The ***resetType*** specifies the type of the last reset (e.g. Cold Power-On Reset, System Reset, Application Reset or Warm Power-On Reset)

- The ***resetTrigger*** specifies the source of the last reset. For instance, the source can be a Power-On Reset (pressing the PORST-Button), a SW triggered reset or a reset triggered by the debugger or any voltage supervision monitor

The function ***IfxScuRcu_evaluateReset()*** evaluates both the ***RSTSTAT*** and ***RSTCON*** registers

- The ***RSTSTAT*** register is evaluated with regard to which reset bits are set, respectively, cleared. Firstly, the warm reset status bits comprising ***ESRx***, ***SMU***, ***SW***, ***STMx*** and ***CBx*** are evaluated. Secondly, the cold reset status bits comprising ***EVRC***, ***EVR33***, ***SWD*** and ***STBYR*** are evaluated if none of the warm reset status bits are set. Finally, the ***PORST*** bit is evaluated

- The ***RSTCON*** register is evaluated to determine the type of reset based on the trigger configuration

Implementation

Reset Detection (cont.)

Based on the ***resetType*** of the ***lastReset***, LED1, LED2 or both are switched on.

Furthermore, the function ***detectResetSource()*** clears the Cold Power-On sticky bits using the function ***IfxScuRcu_clearColdResetStatus()***. Those bits are not cleared automatically and must be explicitly cleared by the application.

The functions ***IfxScuRcu_evaluateReset()*** and ***IfxScuRcu_clearColdResetStatus()*** can be found in the iLLD header ***IfxScuRcu.h***.

Implementation

Reset Trigger

The function ***triggerSwReset()*** triggers either a software Application Reset or a software System Reset, depending on the macro ***RESET_SRC*** given as parameter.

To trigger a software reset, the request trigger in the Reset Configuration Register must be configured first. This is done through the function ***IfxScuRcu_configureResetRequestTrigger()***.

Then, the CPU EndInit protection is cleared with the function ***IfxScuWdt_clearCpuEndinit()*** and the software reset is triggered calling ***IfxCpu_triggerSwReset()***.

Finally, the CPU EndInit protection should be set again, but this instruction cannot be reached since a software reset is triggered right before.

The function ***IfxScuRcu_configureResetRequestTrigger()*** can be found in the iLLD header ***IfxScuRcu.h***.
The function ***IfxScuWdt_clearCpuEndinit()*** can be found in the iLLD header ***IfxScuWdt.h***.
The function ***IfxCpu_triggerSwReset()*** can be found in the iLLD header ***IfxCpu.h***.

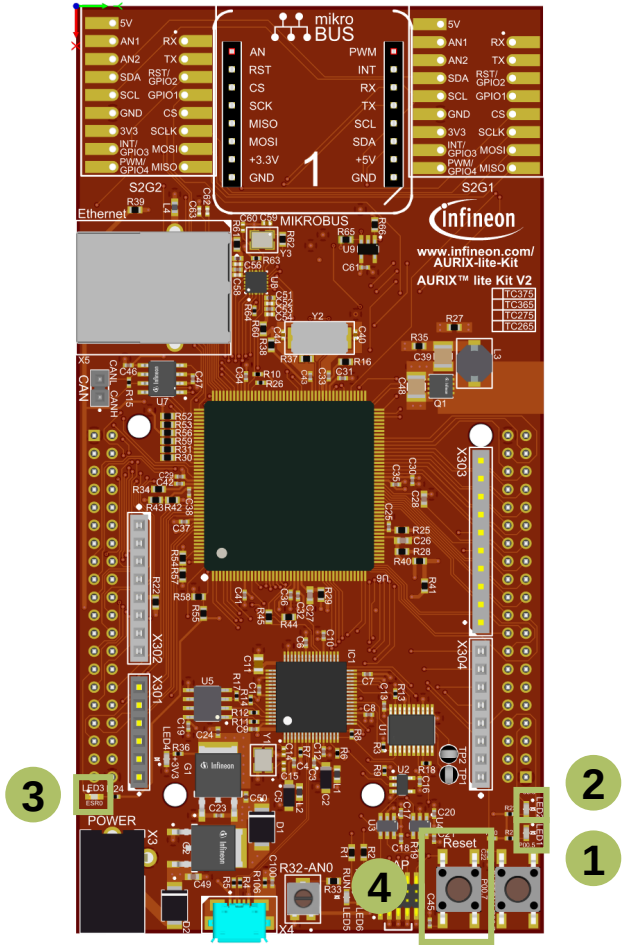
Run and Test

After code compilation and flashing the device, press the PORST button (4) and observe the following behavior:

- Both LED1 (1) and LED2 (2) are turned on for 500 ms because a Warm Power-On reset is detected
- Then, the board is reset by software, therefore the LED3 (3) is blinked once
- Finally, depending on the last occurred reset (given by the ***RESET_SRC*** macro) the LED1 (1) or the LED2 (2) is turned on

The ***RESET_SRC*** macro is firstly set to ***APPLICATION_RESET***. To trigger a system reset, change it to ***SYSTEM_RESET***, re-flash the code, press the PORST button (4) and check that LED2 (2) is switched on after both LED1 (1) and LED2 (2).

Note: To observe the correct behavior of this example, use the Flash button. This ensures that the project is flashed on the board without triggering the debugger.



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

- 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

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

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

Document reference
SCU_Reset_Detection_1_KIT_TC375_LK

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.