SCU_Reset_Detection_1

Detection of reset type

AURIX™ TC2xx Microcontroller Training
V1.0.0

Please read the Important Notice and Warnings at the end of this document

Scope of work

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

The AURIX™ TC2xx 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 register.

Introduction

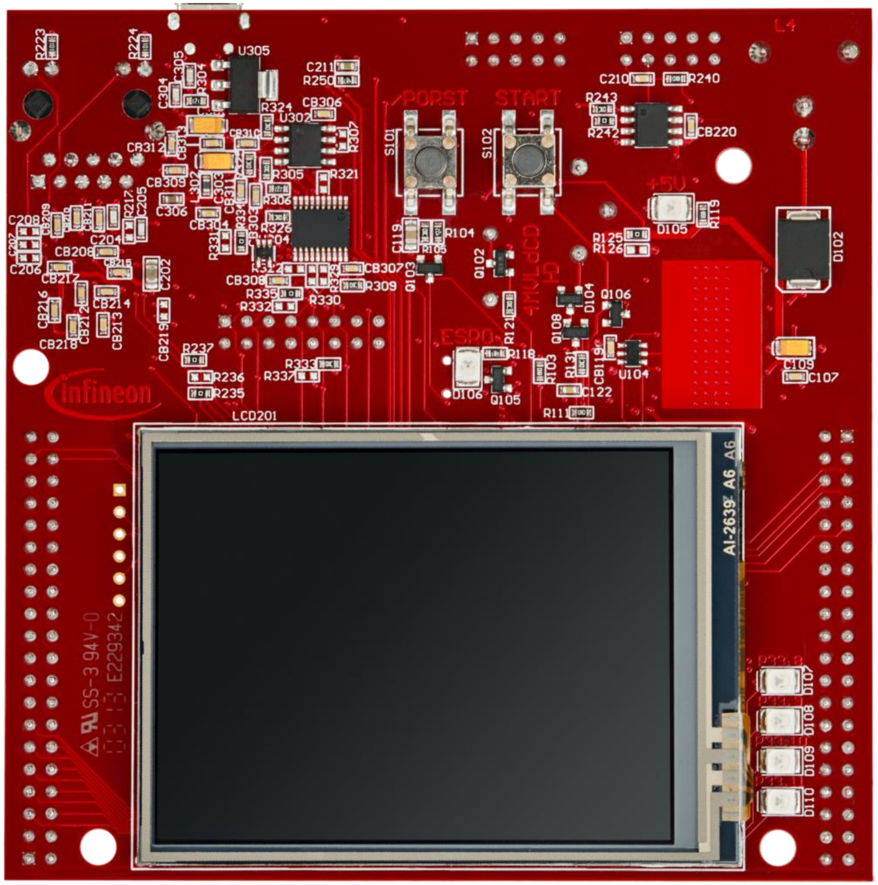
- Resets can be configured and determined in the Reset Control Unit (RCU), belonging to the System Control Unit (SCU).

- There are various reset triggers such as SupplyMonitor, EVRs, PORST, ESRx, JTAG.

- 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_AURIX_TC297_TFT_BC-Step.



Implementation

startScuResetDetection()

- This function executes the ***evaluateReset()*** function, which provides information about the last occurred reset. 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. a 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 PORST-Button), a SW triggered reset or a reset triggered by debugger or any voltage supervision monitor.

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

- The local variable ***swReset*** can be used to specify the type of SW-Reset initiated by the function ***triggerSWReset()***.

Implementation

evaluateReset()

- The function ***evaluateReset()*** evaluates both the ***SCU.RCU.RSTSTAT*** and ***SCU.RCU.RSTCON*** registers.

- The ***SCU.RCU.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 ***EVR13***, ***EVR33***, ***SWD*** and ***STBYR*** are evaluated if none of the warm reset status bits are set. Finally, the ***PORST*** bit is evaluated.

- The ***SCU.RCU.RSTCON*** is evaluated to determine the type of reset, specified for the warm reset status bits, except debugger related reset sources.

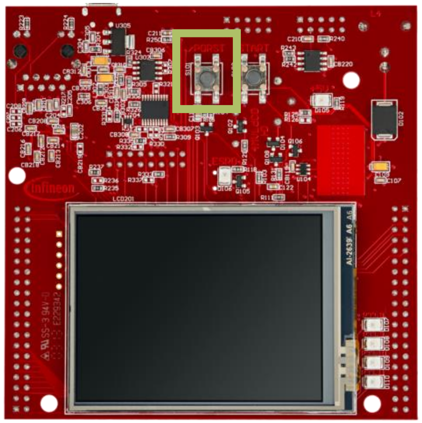
triggerSWReset()

- This function was developed only for testing purposes. Based on the local variable ***swReset***, it triggers either a SW Application Reset or a SW System Reset.

Run and Test

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

1. Run the code
2. Suspend the code execution
3. Watch the local structure variable ***lastReset*** (the elements ***lastReset.resetType*** and ***lastReset.resetTrigger***)
4. Check whether the ***lastReset.resetType*** is set to 'application' reset and whether the ***lastReset.resetTrigger*** is set to 'cb3'.
5. Press button 'PORST'



6. Perform steps 1. through 3.
7. Check whether the ***lastReset.resetType*** is set to 'warmpoweron' reset and whether the ***lastReset.resetTrigger*** is set to 'porst'.

Run and Test

8. Set the local variable ***swReset*** to '1'
9. Perform steps 1. through 3.
10. Check whether the ***lastReset.resetType*** is set to 'application' reset and whether the ***lastReset.resetTrigger*** is set to 'sw'.
11. Set the local variable ***swReset*** to '2'
12. Perform steps 1. through 3.
13. Check whether the ***lastReset.resetType*** is set to 'system' reset and whether the ***lastReset.resetTrigger*** is set to 'sw'.

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 2019-10
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

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.