STM_Interrupt_1
for KIT_AURIX_TC275_SB

Interrupt generation via STM trigger

AURIX™ TC2xx Microcontroller Training
V1.0.3

The STM is configured to trigger an interrupt every 500 ms. The interrupt toggles an LED.

The System Timer (STM) module counts up to a configured compare value, when it reaches the specific value, it triggers an interrupt and resets its counter value. The value is set to trigger the interrupt every 500 ms; at each interrupt an LED is toggled.

Introduction

› The System Timer (STM) is a free running 64-bit counter that can be used for timing applications requiring both high precision and long period

› Among other features, the STM has the capability to **generate interrupts** when its count reaches a predefined **compare value**

› The comparison is flexible in terms of bit sets.
Any of the 64 bits of the STM can be selected for comparison

Hardware setup

This code example has been developed for the board KIT_AURIX_TC275_ARD_SB.
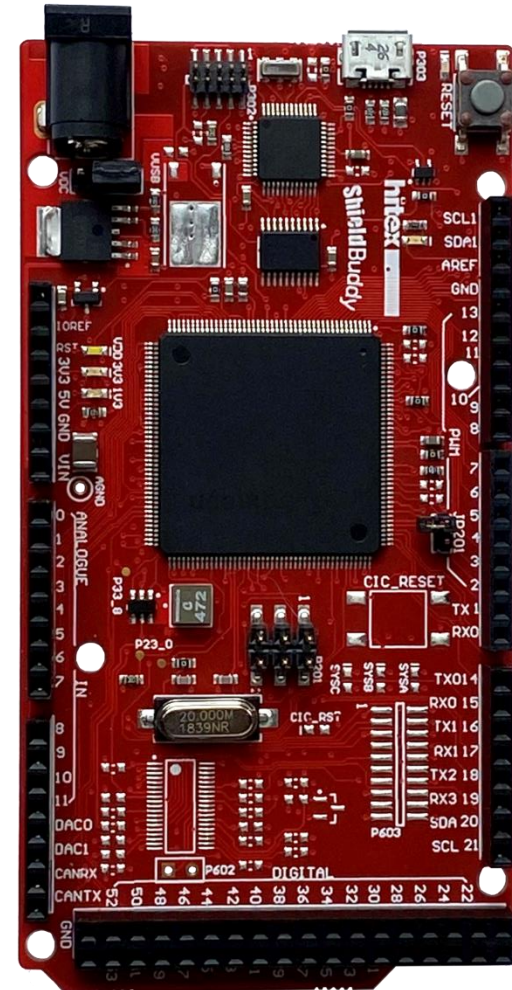
Implementation

Configure and control the LED

The LED is toggled by controlling the port pin to which it is connected using functions from the iLLD header ***IfxPort.h***.

Inside the function ***initPeripherals()***, the port pin is configured to push-pull output mode using the function ***IfxPort_setPinMode()***.

During program execution, inside the interrupt service routine ***isrSTM()***, the LED is switched on and off using the function ***IfxPort_setPinState()***.

Implementation

Configure the STM

Configuration of the STM is done inside the function *initSTM()* by initializing an instance of the *IfxStm_CompareConfig* structure with default values through the function *IfxStm_initCompareConfig()*.
Then, the following parameters are modified:

› *ticks* – the total amount of ticks to count before the interrupt generation
› *triggerPriority* – priority of the interrupt generated by the STM on compare match. It can be a value from 0 to 255, with 0 meaning interrupt is disabled and 255 is the highest priority
› *typeOfService* – to define which service provider is responsible for handling the interrupt. This can be any of the available CPUs or the DMA

The configuration is then applied to the STM via the function *IfxStm_initCompare()*.

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

Configure the Interrupt Service Routine

The method implementing the ISR (*isrSTM()*) needs to be assigned a **priority** via the macro *IFX_INTERRUPT(isr, vectabNum, priority)*.

When triggered, the ISR:
› Instructs the STM to raise the next interrupt after a certain amount of time by **updating** its **compare register** with the function *IfxStm_increaseCompare()*
› Toggles the LED by changing the state of the port pin using the function *IfxPort_setPinState()*

Run and Test

After code compilation and flashing the device, observe the **LED** (1), which should be blinking.
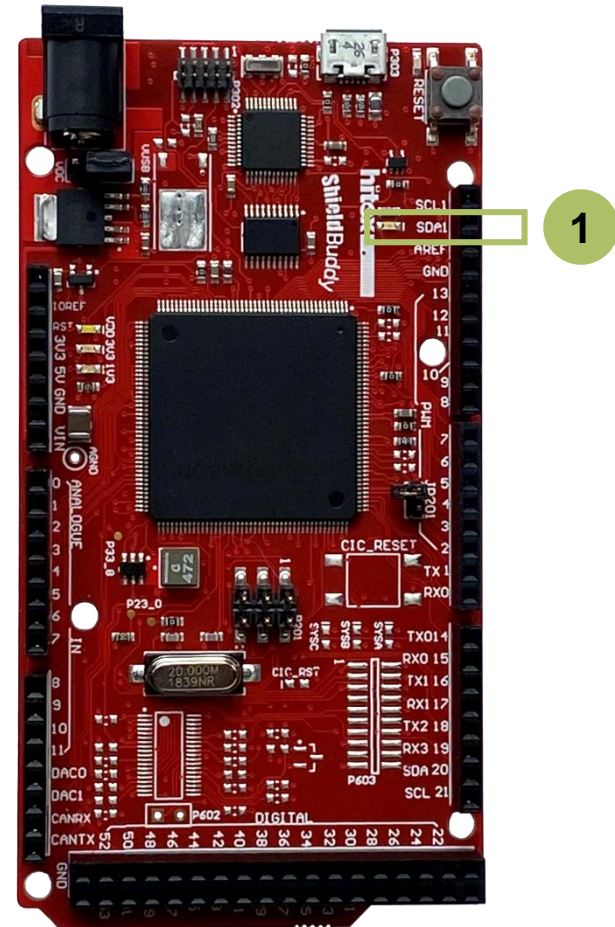
References

› 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

Revision history

Revision	Description of change
V1.0.3	Fixed description of ISR priority assignment
V1.0.2	Update of version to be in line with the code example's version
V1.0.1	Improved description of the used iLLD function (page 6)
V1.0.0	Initial version

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.