GPT12_Timer_Concatenation_1 for KIT_AURIX_TC275_LK

Timer concatenation of GPT12

AURIX™ TC2xx Microcontroller Training
V1.0.0

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

Scope of work

A 32-bit timer is formed by the concatenation of two timers of the GPT12 module. On timer overflow an interrupt is generated, which toggles an LED.

A 32-bit timer is formed by the concatenation of two 16-bit timers of the GPT12 module. When the 32-bit timer overflows, an interrupt is generated toggling an LED.

Copyright © Infineon Technologies AG 2021. All rights reserved.

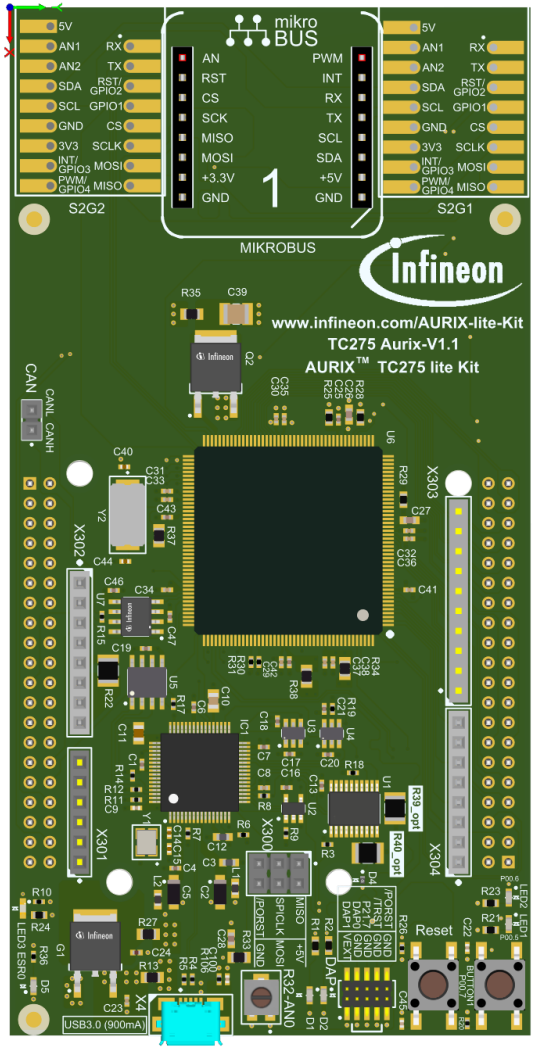
Introduction

- The General Purpose Timer Unit GPT12 is divided into two GPT blocks (GPT1 and GPT2)
- The GPT12 unit incorporates five 16-bit timers which may operate independently in several different modes
- Block GPT2 contains two timers/counter: the core timer T6 and one auxiliary timer T5
- Both timers of block GPT2 can run in Timer Mode, Gated Timer Mode or Counter Mode
- An additional Capture/Reload register (CAPREL) supports also capture and reload operation
- The timer T5 in Counter Mode and the timer T6 in Timer Mode are used together in this example

Copyright © Infineon Technologies AG 2021. All rights reserved.

Hardware setup

This code example has been developed for the board KIT_AURIX_TC275_LITE.



Copyright © Infineon Technologies AG 2021. All rights reserved.

Implementation

Configuration of the GPT12 module

The function ***init_GPT12_module()*** is used to configure the two timers T5 and T6 of GPT2 for concatenated 32-bit operation. It contains the following steps:

- The GPT12 module is enabled by calling the function ***IfxGpt12_enableModule()***
- GPT2 prescaler is set to ***IfxGpt12_Gpt2BlockPrescaler_2*** to get the highest possible frequency by calling ***IfxGpt12_setGpt2BlockPrescaler()***
- The functions ***IfxGpt12_T6_setMode()*** and ***IfxGpt12_T5_setMode()*** are used to configure the modes of both timers. The core timer T6 is set to ***IfxGpt12_Mode_timer*** and the auxiliary timer T5 is set to ***IfxGpt12_Mode_counter***
- Both the rising and falling edge of the Output Toggle Latch of timer T6 are selected as active transition with the function ***IfxGpt12_T5_setCounterInputMode()***
- The function ***IfxGpt12_T5_setRemoteControl()*** configures the timer T5 to be activated by the run bit of core timer T6. ***IfxGpt12_T6_run()*** function starts timer T6 and implicitly timer T5, because the overflow of T6 is used as clock for T5

The functions above are provided by the iLLD header ***IfxGpt12.h***.

Copyright © Infineon Technologies AG 2021. All rights reserved.

Implementation

Configuring the interrupt service routing for block GPT2

- The function ***IfxGpt12_T5_getSrc()*** returns the pointer to the service request control (SRC) register of timer T5
- The function ***IfxGpt12_T6_getSrc()*** returns the pointer to the SRC register of timer T6
- The above mentioned pointers to the SRC registers are used to initialize and enable the ISRs by setting the service provider and the priority number. This is done via the functions ***IfxSrc_init()*** and ***IfxSrc_enable()*** provided by the iLLD header ***IfxSrc.h***
- The method implementing the ISR needs to be assigned a **priority** via the macro ***IFX_INTERRUPT(isr, vectabNum, priority)***

Copyright © Infineon Technologies AG 2021. All rights reserved.

Implementation

The Interrupt Service Routines (ISRs)

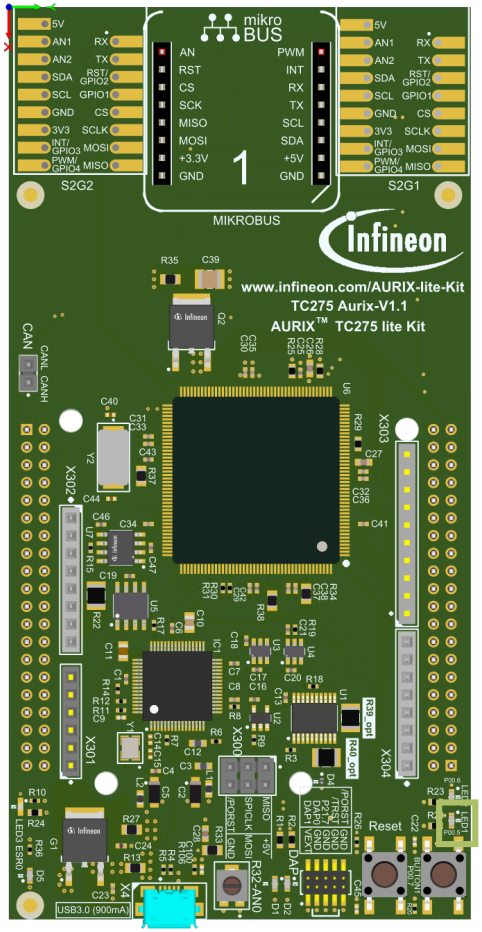
- The incrementation of the counter values is different for both timers:
  - The counter of timer T6 (Timer Mode) is incremented with the frequency of timer T6
  - The counter of timer T5 (Counter Mode) is incremented by each overflow of timer T6
- The overflow of each timer triggers an interrupt
- Each timer has associated its own Interrupt Service Routine
- The ISR of timer T6 is triggered after the overflow of the 16-bit timer value of timer T6
- Due to the timer concatenation, the ISR of timer T5 is triggered with the overflow of the 32-bit timer value. This ISR toggles an LED
- The 32-bit timer value is calculated in the ISR of T6 and is represented by ***timerValueHIGH*** (T5) and ***timerValueLOW*** (T6)

Copyright © Infineon Technologies AG 2021. All rights reserved.

Run and Test

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

- Check if LED1 (1) is toggling its state every 86 seconds:
  - Maximum value of 32-bit timer is 4294967295
  - Frequency of GPT12 module is 100 MHz
  - Prescaler of GPT2 block is set to 2
  - Timer T6 frequency is 100 MHz / 2 = **50 MHz**, which means the value of the 32-bit timer will be incremented each **20 ns** (1 / 50 MHz)
  - Overflow of 32-bit timer after **≈ 86 s** (4294967295 * 20 ns)
- Check in the debugger if parameter ***g_timerValue32Bit*** is incrementing



Copyright © Infineon Technologies AG 2021. All rights reserved.

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

Copyright © Infineon Technologies AG 2021. All rights reserved.

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

Edition 2021-06
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

Document reference
GPT12_Timer_Concatenation_1_KIT_TC275_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.