

CCU6_Interrupt_1 for KIT_AURIX_TC334_LK

CCU6 interrupt generation

AURIX™ TC3xx Microcontroller Training
V1.0.0



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

Scope of work

The CCU6 unit generates an interrupt every 500 ms and increments a counter in its Interrupt Service Routine (ISR).

Timer T12 of the CCU6 unit generates an interrupt on a "period match" event every 500 ms. Inside the ISR, a counter variable is incremented. The LEDs toggle based on the value of the counter:

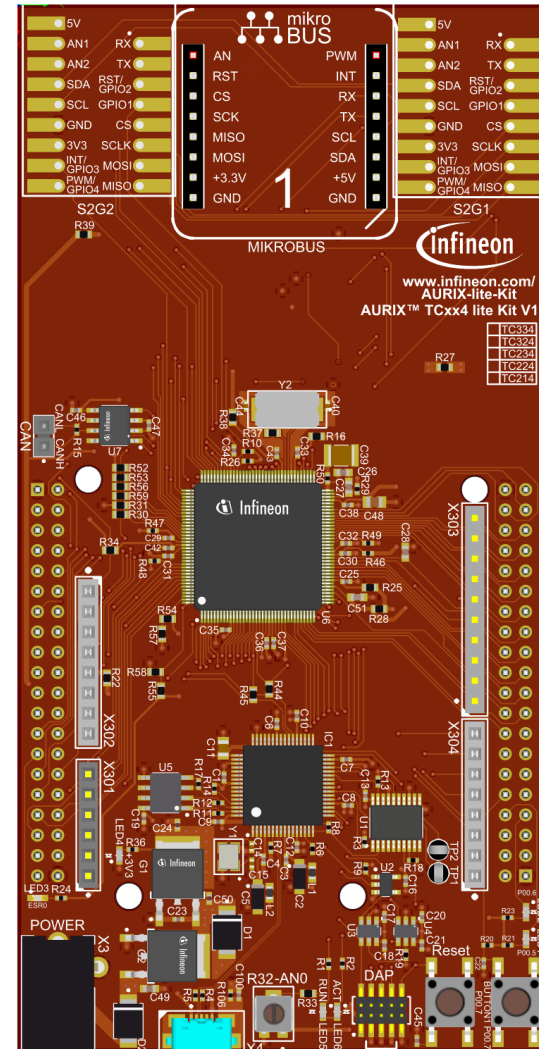
- › LED1 on and LED2 off: counter % 2 == 0
- › LED1 off and LED2 on: counter % 2 == 1

Introduction

- › The Capture/Compare Unit 6 (CCU6) is a high-resolution 16-bit capture and compare unit with application specific modes, mainly used for AC drive control
- › The CCU6 unit is made up of a Timer T12 Block with three capture/compare channels and a Timer T13 Block with one compare channel. In this example, Timer T12 Block is used
- › Among other features, the CCU6 has the capability to **generate interrupts** when its count reaches a predefined **period value** (16-bit)

Hardware setup

This code example has been developed for the board `KIT_A2G_TC334_LITE`.



Implementation

Configure and control the LEDs

The LEDs can be switched on/off by controlling the port pins to which they are connected using methods from the iLLD headers ***IfxPort.h***.

In the setup phase, the port pins connected to the LEDs are configured as push-pull output mode using the function ***IfxPort_setPinMode()***.

During program execution, the LEDs are switched on and off using the function ***IfxPort_setPinState()***.

Implementation

Configure the CCU6 unit

Configuration of the CCU6 Timer is done by initializing an instance of the ***IfxCcu6_Timer_Config*** structure, which contains the following fields:

- › ***base*** – a structure that allows to set:
 - ***frequency*** – input clock frequency in Hz of the timer module
 - ***period*** – 16-bit register that determines the maximum count value for the timer
- › ***timer*** – a parameter that allows to choose which of the two timers to configure
- › ***interrupt1*** – a structure that allows to set:
 - ***priority*** – priority of the generated interrupt by CCU6. It can be a value from 0 to 255, with 0 meaning interrupt is disabled, and 255 is the highest priority
 - ***type of service*** – defines the service provider responsible for handling the interrupt, which can be any of the available CPUs, or the DMA
- › ***trigger*** – a structure for configuring the trigger, that allows to start Timer T13 in sync with Timer T12

The function ***IfxCcu6_Timer_initModuleConfig()*** fills the configuration structure with default values and ***IfxCcu6_Timer_initModule()*** function initializes the timer module with the user configuration.

Both functions can be found in the iLLD header ***IfxCcu6_Timer.h***.

Implementation

Timer module frequency

In Timer Mode, the input clock of the timer module is derived from the internal module clock f_{CC6} .

By default, the internal module clock f_{CC6} is initialized by the iLLDs to run at 100 MHz.

The initialization is done by the function ***lfxScuCcu_init()*** that can be found in the iLLD header ***lfxScuCcu.h***.

Starting from this value, it is possible to set the timer module frequency to the following exact values:

100000000 Hz	= 100 MHz	(f_{CC6})
50000000 Hz	= 50 MHz	($f_{CC6} / 2$)
25000000 Hz	= 25 MHz	($f_{CC6} / 4$)
12500000 Hz	= 12.5 MHz	($f_{CC6} / 8$)
6250000 Hz	= 6.25 MHz	($f_{CC6} / 16$)
3125000 Hz	~ 3 MHz	($f_{CC6} / 32$)
1562500 Hz	~ 1.5 MHz	($f_{CC6} / 64$)
781250 Hz	~ 780 KHz	($f_{CC6} / 128$)
390625 Hz	~ 390 KHz	($f_{CC6} / 256$)
195312.5 Hz	~ 200 KHz	($f_{CC6} / 512$)
97656.25 Hz	~ 100 KHz	($f_{CC6} / 1024$)
48828.12 Hz	~ 50 KHz	($f_{CC6} / 2048$)
24414.06 Hz	~ 25 KHz	($f_{CC6} / 4096$)
12207.03 Hz	~ 12.5 KHz	($f_{CC6} / 8192$)
6103.51 Hz	~ 6 KHz	($f_{CC6} / 16384$)
3051.75 Hz	~ 3 KHz	($f_{CC6} / 32768$)

Note: Any value can be set as ***frequency*** parameter, but the software will round up the chosen value to the nearest higher frequency listed in the above table (e.g. setting ***timerConfig.base.t12Frequency = 400000*** the timer will run at 781250 Hz).

Implementation

ISR Frequency calculation

To calculate the actual ISR frequency f_{ISR} (rate at which the ISR is triggered by the CCU6), two parameters have to be considered: **CCU6 clock frequency** and **period**.

- › The **CCU6 clock frequency** can be freely selected as discussed in the previous slide
- › The **period** value is stored in a 16-bit register, which limits its maximum value to 65535

The ISR frequency can then be calculated as:

$$f_{\text{ISR}} = (\text{CCU6 clock frequency}) / (\text{period} + 1)$$

Implementation

Period calculation example

Target: set a period frequency of $f_{\text{ISR}} = 10 \text{ Hz}$

Considering the maximum value of the period (65535) and the final ISR frequency that needs to be achieved (10 Hz), the ISR frequency formula from the previous slide leads to the following:

$$10 \text{ Hz} = (\text{CCU6 clock frequency}) / (\text{period} + 1) = (\text{CCU6 clock frequency}) / (65535 + 1)$$

which leads to: **CCU6 clock frequency** = $(65535 + 1) * 10 \text{ Hz} = 655360 \text{ Hz}$

Since the maximum value of 65535 for the period cannot be exceeded, it is needed to search into the previous table for the nearest lower exact frequency available, which is 390625 Hz.

Taking this value, the period can be easily re-calculated as:

$$\text{period} = (\text{CCU6 clock frequency} / f_{\text{ISR}}) - 1 = (390625 / 10) - 1 \approx 39061$$

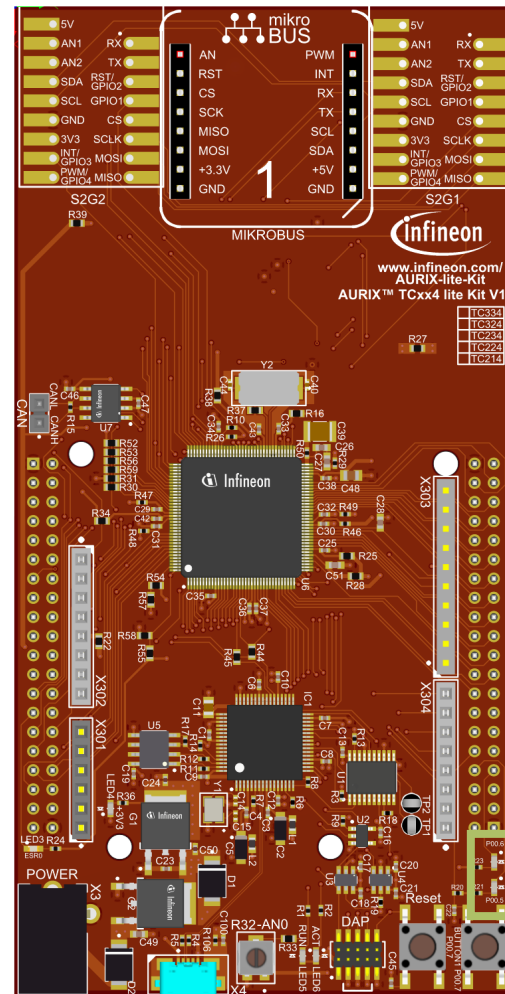
Finally, selecting these values as **period** and **CCU6 clock frequency**, f_{ISR} can be calculated:

$$f_{\text{ISR}} = (\text{CCU6 clock frequency}) / (\text{period} + 1) = 390625 / 39062 \approx 10 \text{ Hz}$$

Run and Test

After code compilation and flashing the device, observe the behavior of the LEDs.

- › Check that LED1 and LED2 are blinking in sequence every 500 ms



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 2021-12

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

CCU6_Interrupt_1_KIT_TC334_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.