GTM_TOM_ADC_1 for KIT_AURIX_TC275_LK

GTM TOM ADC conversion triggering

AURIX™ TC2xx Microcontroller Training
V1.0.0

Scope of work

GTM TOM is used to generate a PWM signal and is used to trigger the Versatile Analog-to-Digital Converter (VADC). Based on the read value, the intensity of a LED is driven.

The LED is driven by pin 6 of the port 00. The state of the pin is controlled by the PWM signal generated by the TOM timer of GTM. The PWM duty cycle is defined by the result value of VADC.

Introduction

- The Generic Timer Module (GTM) is a modular timer unit designed to accommodate many timer applications
- It has an in-built Timer Output Module (TOM) that can offer up to 16 independent channels to generate output signals
- The Clock Management Unit (CMU) is responsible for clock generation of the GTM. The Fixed Clock Generation (FXU) is one of its subunits and it provides five predefined non-configurable clocks for GTM modules, including the TOM

Introduction

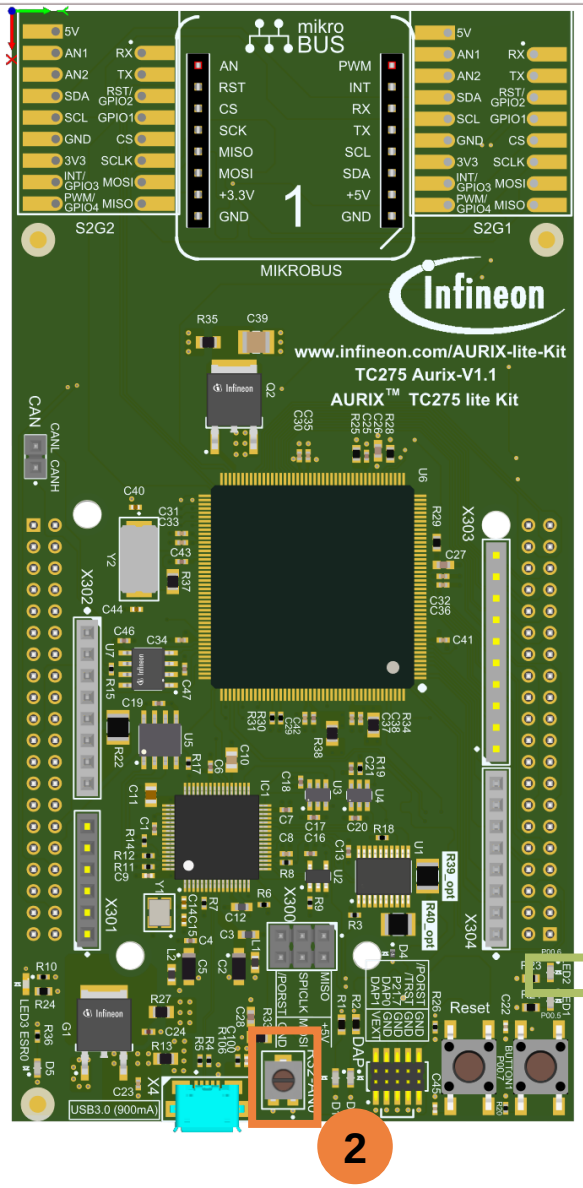
- The Versatile Analog-to-Digital Converter module (VADC) of the AURIX™ TC27x comprises 8 independent analog to digital converters, each converting with a resolution up to 12-bit
- Several request sources can request an Analog/Digital conversion following different configurations. A conversion can be requested to be done once or repeatedly
- Interrupts can be generated once conversions are finished
- External peripherals like GTM, etc. can trigger a sampling request

Hardware setup

This code example has been developed for the board KIT_AURIX_TC275_LITE.

LED2 (1) is used for this example.

The pin AN0 (2), connected to the board's potentiometer, is used to control the intensity of LED2.



Note: The reference voltage (VAREF) of the VADC on the board KIT_AURIX_TC275_LITE is 3.3 V.

Implementation

Configuring the TOM

The configuration of the TOM is done by calling the initialization function ***initGtmTomPwm()*** containing the following steps:

- Enable the GTM by calling the function ***IfxGtm_enable()***
- Enable the FXU clocks by calling the function ***IfxGtm_Cmu_enableClocks()***

The function ***IfxGtm_Tom_Pwm_initConfig()*** initializes an instance of the structure ***IfxGtm_Tom_Pwm_Config*** with its default values.

The ***IfxGtm_Tom_Pwm_Config*** structure can be modified to set the following parameters to initialize the module:

- ***tom*** – Selection of the TOM which is counting (TOM 0 in this example)
- ***tomChannel*** – Selection of the channel which is driving the LED (Channel 13 in this example)
- ***period*** – Setting of the period for the PWM signal to the desired value
- ***pin.outputPin*** – Selection of the LED as output pin
- ***synchronousUpdateEnable*** – Enabling of synchronous update of the timer

Implementation

Configuring the TOM

After configuration, the function ***IfxGtm_Tom_Pwm_init()*** initializes and activates the TOM with the user configuration.
Start the PWM with the function ***IfxGtm_Tom_Pwm_start()***.

Setting the duty cycle

The setting of the duty cycle is done by calling the function ***setDutyCycle()***, which contains the following steps:

- Set the ***dutyCycle*** parameters of the configuration structure to set the duty cycle of the PWM signal to the desired value
- Call the function ***IfxGtm_Tom_Pwm_init()*** to reconfigure the TOM with the new value of the duty cycle

All the functions used for the configuration of the TOM are provided by the iLLD header ***IfxGtm_Tom_Pwm.h***.

Implementation

Configuration of the VADC

The configuration of the VADC is done in the ***initVadc()*** function in three different steps:

- Configuration of the **VADC module**
- Configuration of the **VADC group**
- Configuration of the **VADC channel**

Configuration of the VADC module

The functions used for configuring the VADC module are:

- ***IfxVadc_Adc_initModuleConfig()*** – initializes the VADC module configuration structure with the default values
- ***IfxVadc_Adc_initModule()*** – initializes the VADC module with the user configuration, which in this case is the default configuration

Implementation

Configuration of the VADC group

Configuration of the VADC group is done by initializing an instance of the ***IfxVadc_Adc_GroupConfig*** structure, which contains the following fields:

- ***groupId*** – a parameter that allows to choose which of the groups to configure
- ***master*** – indicates which group is the master. In this example only one channel of one group is used, therefore the chosen group is also the master
- ***arbiter*** – a structure that represents the enabled request sources (which can be Scan, Queue and/or Background sources; in this example the Scan source is used)
- ***scanRequest*** – a structure to set the Scan Request Source configuration
  - ***autoscanEnabled*** – specifies if Autoscan mode is active
- ***queueRequest*** – a structure to set the queued mode configuration
  - ***triggerConfig*** – a structure that specifies the trigger configuration, which includes
    - ***gatingSource*** – a parameter to specify the gate input for the group
    - ***triggerSource*** – a parameter to specify the source of the trigger

The functions used for configuring the VADC group are:

- ***IfxVadc_Adc_initGroupConfig()*** – fills the group configuration structure with default values
- ***IfxVadc_Adc_initGroup()*** – initializes the VADC group specified in the parameters with the user configuration

Implementation

Configuration of the VADC channel

Configuration of the VADC channel is done by initializing an instance of the ***IfxVadc_Adc_ChannelConfig*** structure, which contains the following fields:

- ***channelId*** – a parameter that allows to choose which of the channels to configure
- ***resultRegister*** – indicates the register where the sample value is stored
- ***resultPriority*** – the priority of the result trigger interrupt
- ***resultServProvider*** – interrupt service provider for the result trigger interrupt. This can be any of the available CPUs or the DMA. In this example, the interrupt is used to set the duty cycle of the PWM generated by the GTM TOM
- ***backgroundChannel –*** boolean that specifies if the channel is scanned in a background task

The functions used for configuring the VADC channel are:

- ***IfxVadc_Adc_initChannelConfig()*** – fills the channel configuration structure with default values
- ***IfxVadc_Adc_initChannel()*** – initializes the channel with the user configuration
- ***IfxVadc_Adc_addToQueue()*** – adds over-sampling for current measurement
- ***IfxVadc_Adc_setScan()*** – adds the channel used for the measurement to the scan sequence

Update the Duty Cycle

The function ***ISRresultADC()*** is used to update the duty cycle of the PWM signal.
The function ***IfxVadc_Adc_getResult()*** gets the result from the VADC result register, that is then mapped into a duty cycle value to update the PWM signal.

All the functions used for configuring the VADC module, group and channel can be found in the iLLD header ***IfxVadc_Adc.h***.

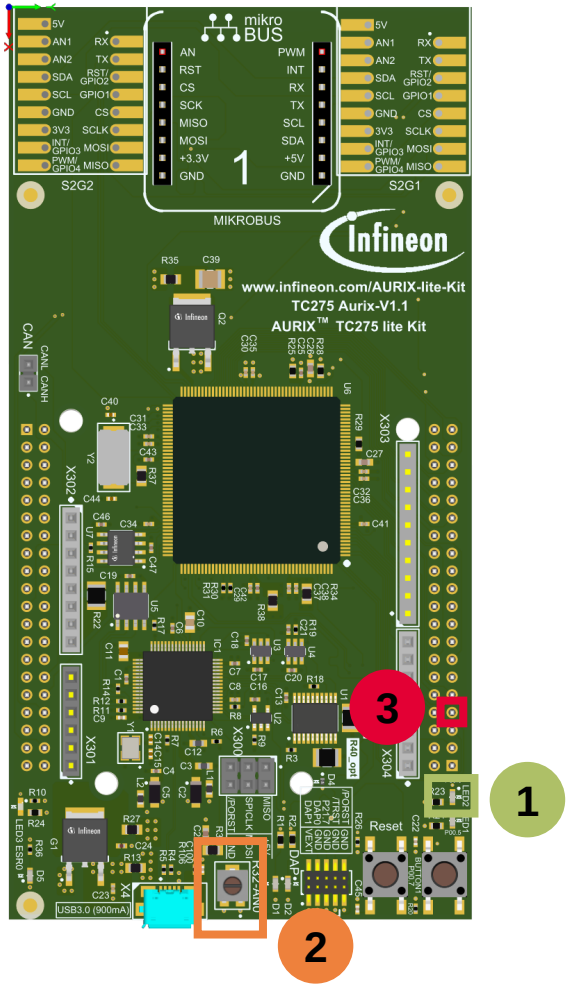
Run and Test

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

- Turn the potentiometer connected to pin AN0 (2) to drive the intensity of the LED2 (1)

Alternatively, an oscilloscope probe can be connected on pin 7 of X2 connector (3) to visualize the PWM signal.

	X2		
GND	40	39	VEXT
P33.10	38	37	P33.9 - TXD1_S2G1
RXD1_S2G1 - P33.8	36	35	P33.7
P33.6	34	33	P33.5
P33.4	32	31	P33.3
P33.2	30	29	P33.1
P33.0	28	27	AN0 - Potentiometer
AN1	26	25	AN2
AN3	24	23	AN4
AN5	22	21	AN6
AN7	20	19	AN44
AN45	18	17	AN46
AN47	16	15	VAREF1
P00.11	14	13	P00.12
P00.9	12	11	P00.10
Button1 - P00.7	10	9	P00.8
LED1 - P00.5	8	7	P00.6 - LED2
P00.3	6	5	P00.2
P00.1	4	3	P00.0 - TXDCAN
VDD_USB	2	1	GND



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 2022-12
Published by
Infineon Technologies AG
81726 Munich, Germany

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

Document reference
GTM_TOM_ADC_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.