GTM_TIM_Capture_1 for KIT_AURIX_TC375_LK

PWM input capturing via TIM

AURIX™ TC3xx Microcontroller Training
V1.0.0

Scope of work

The Timer Input Module (TIM) of the GTM is used to capture an external PWM signal and calculate the PWM signal frequency and duty cycle.

The TIM is used in capture mode. The data from the captured PWM signal is used to calculate the PWM signal frequency and duty cycle in software. The frequency and the duty cycle are then stored in variables.

Introduction

GTM TIM | GTM TOM

- The Generic Timer Module (GTM) is a modular timer unit designed to accommodate many timer applications
- The Timer Input Module (TIM) is responsible for filtering and capturing input signals of the GTM
- The Clock Management Unit (CMU) is responsible for clock generation of the GTM. The Configurable Clock Generation Subunit (CFGU) provides eight clock sources for the GTM submodules: TIM, TBU, MON and ATOM

Introduction

GTM TIM | GTM TOM

- The GTM has an in-built Timer Output Module (TOM) that can offer 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

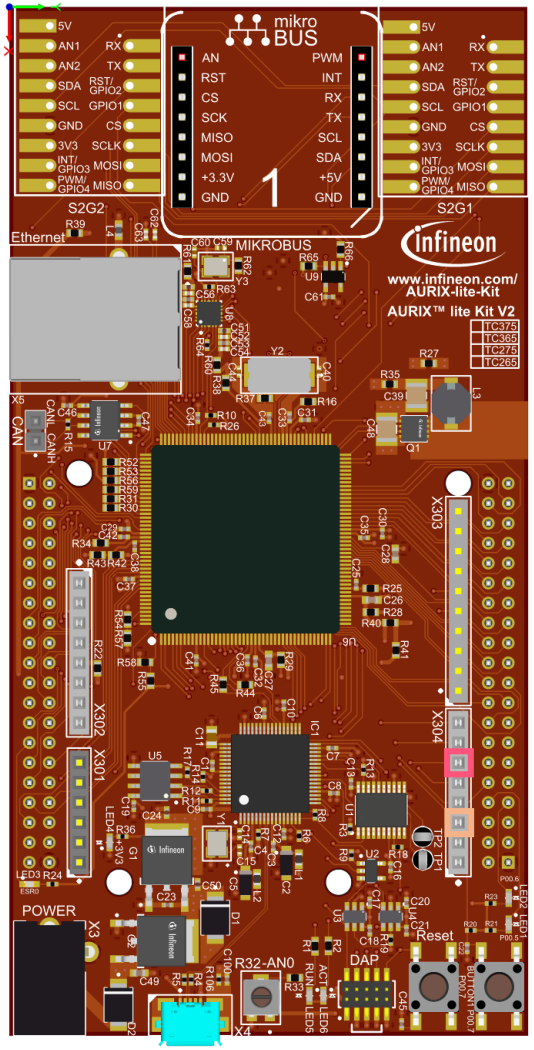
Hardware setup

This code example has been developed for the board KIT_A2G_TC375_LITE.

Connect the TIM input port pin P02.0 with the TOM output port pin P02.3.

X304

8	P02.4
7	P02.5
6	P02.3
5	P10.4
4	P02.1
3	P02.0
2	P15.2 - UART_TX
1	P15.3 - UART_RX



Implementation

Configuring the TIM

The configuration of the TIM is done by calling the initialization function ***init_TIM()***, containing the following steps:

- Enable the GTM by calling the function ***IfxGtm_enable()***
- Enable the CMU clock 0 by calling the function ***IfxGtm_Cmu_enableClocks()***
- Initialize an instance of the ***IfxGtm_Tim_In_Config*** structure with the default parameters by calling the iLLD function ***IfxGtm_Tim_In_initConfig()***
- The ***IfxGtm_Tim_In_Config*** structure allows the setting of multiple parameters in order to initialize the module, including the following:
  - ***filter.inputPin*** – Selection of the input port pin of the TIM (port pin P02.0 in this example)
  - ***filter.inputPinMode*** – Selection of the input mode for the input port pin of the TIM (***IfxPort_InputMode_pulldown*** in this example)
- After setting the configuration parameters, the function ***IfxGtm_Tim_In_init()*** applies the user configuration to the module

All the functions used for the configuration of the TIM are provided by the iLLD header ***IfxGtm_Tim_In.h***.

Implementation

Configuring the TOM

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

- Enable the FXU clocks by calling the function ***IfxGtm_Cmu_enableClocks()***
- Initialize an instance of the structure ***IfxGtm_Tom_Pwm_Config*** with its default values by calling the function ***IfxGtm_Tom_Pwm_initConfig()***
- The ***IfxGtm_Tom_Pwm_Config*** structure allows to set the following parameters to initialize the module:
  - ***tom*** – Selection of the TOM which is counting
  - ***tomChannel*** – Selection of the channel which is driving the output port pin
  - ***period*** – Setting of the period for the PWM signal to the desired value
  - ***dutyCycle*** – Setting of the duty cycle of the PWM signal to the desired value
  - ***pin.outputPin*** – Selection of the output port pin
  - ***synchronousUpdateEnable*** – Enabling of Synchronous Update of the timer
  - ***clock*** – Selection of the clock used for the generation of the PWM

Implementation

Configuring the TOM

- After setting the configuration parameters, the function ***IfxGtm_Tom_Pwm_init()*** applies the user configuration to the module
- Start the PWM with the function ***IfxGtm_Tom_Pwm_start()***

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

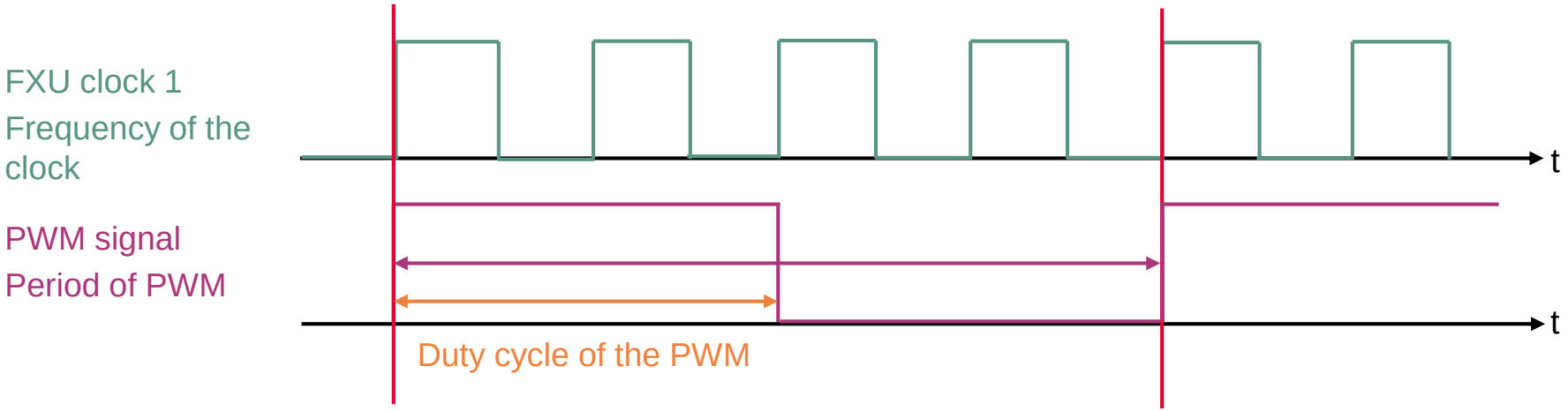
Implementation

PWM generation example

The FXU clock 1 frequency (f_{fxclk1}) is 6.250 MHz in this example. The period value to have the desired PWM frequency (f_{PWM}) is calculated with the following formula:

$$Period = \frac{f_{fxclk1}}{f_{PWM}}$$

In this example: $Period = \frac{6.250\ MHz}{125\ Hz} = 50000\ ticks \Rightarrow 8\ ms$



Implementation

Measuring PWM frequency and duty cycle

The measurement of the PWM frequency and duty cycle is done by calling the function ***measure_PWM()*** containing the following steps:

- Update the measured PWM data by calling the iLLD function ***IfxGtm_Tim_In_update()***
- Get the PWM period in seconds with the iLLD function ***IfxGtm_Tim_In_getPeriodSecond()***
- Calculate the PWM frequency with the formula: frequency(Hz) = 1 / period(s)
- Get the PWM duty cycle by calling the iLLD function ***IfxGtm_Tim_In_getDutyPercent()***

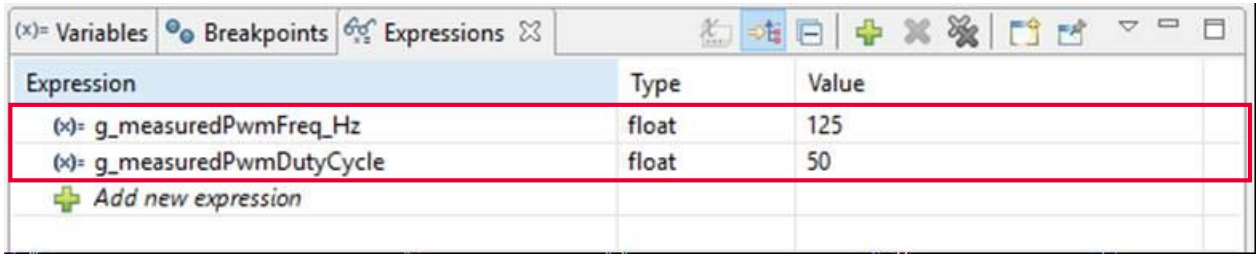
All the iLLD functions above are provided by the iLLD header ***IfxGtm_Tim_In.h***.

Run and Test

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

- Add the variables ***g_measuredPwmFreq_Hz*** and ***g_measuredPwmDutyCycle*** to the **Watch Expressions** on the debugger

- Suspend the program to check the value of the variables



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 2021-03
Published by
Infineon Technologies AG
81726 Munich, Germany

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

Document reference
GTM_TIM_Capture_1_KIT_TC375_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.