IOM_PWM_Monitor_1
PWM monitor via IOM

AURIX™ TC2xx Microcontroller Training
V1.0.0

A PWM signal's duty cycle is monitored using the IOM module.

The PWM signal is routed through the port to the Input Output Monitor (IOM) module. Its duty cycle is continuously compared with a threshold. If the value of the duty cycle goes below the threshold, an LED is switched on.

Introduction

› In this example, the following modules are used:
  - IOM (Input Output Monitor)
  - SMU (Safety Management Unit)
  - IR (Interrupt Router)
  - GTM (Generic Timer Module)

› All the modules above are presented in this tutorial, although a specific focus is given to the IOM and the SMU modules.

Introduction

IOM	SMU	IR	GTM

› The Input Output Monitor (IOM) serves as a smart I/O comparison unit:
  − Checks for the correct operation of system peripheral outputs that might serve and/or control externally-attached hardware
  − Monitors the correct operation of the hardware itself, including any sensors whose signals might serve as an input to the monitoring function

› The IOM can be used to either monitor a signal for a desired behaviour or compare a monitor signal with a reference signal for safety requirements.

› Up to 16 monitoring points can be configured, with the capability to generate a system event that can be driven from:
  − one individual configuration
  − several individual configurations
  − multiple occurrences of a monitored condition
  − combination of several conditions depending upon configuration

› Features of IOM:
  − 16 Filter & Prescaler Channels
  − 16 Logic Analyzer Modules (LAMs)
  − 1 EXOR combiner, configurable, to select a range up to 8 GTM inputs
  − 1 Event Combiner Module (ECM), taking the 16 local event signals and generating a system event signal from a single, combination or multiple local event(s)
  − System Peripheral Bus (SPB) interface for configuration and status register interaction

Introduction

IOM	SMU	IR	GTM

› The Safety Management Unit (SMU) provides a generic interface to manage the behaviour of the microcontroller under the presence of faults.

› It centralizes all the alarm signals related to the different hardware and software-based safety mechanisms.

› The alarms are classified into alarm groups. This does not define any hierarchy but only a logical mapping to internal configuration registers. The configuration options therefore specify the behaviour of the SMU when an alarm event is detected:
  – The alarm event can trigger an internal action
  – The alarm event can indicate a fault to the external environment via the Error Pin

› Internal action resulting from an alarm event can be interfaced to the Interrupt Router via the following signals:
  – SMU Interrupt Request 0
  – SMU Interrupt Request 1
  – SMU Interrupt Request 2

› The SMU in combination with the embedded safety mechanisms enables the detection of more than 99% of the critical failure modes of the microcontroller.

Introduction

IOM	SMU	IR	GTM

› The interrupt system in the AURIX™ TC2xx devices is implemented in the **Interrupt Router** (IR).

› **Interrupt Requests** (or Service Requests) can be serviced either by the CPUs or by the DMA module (both called Service Providers).

› An interrupt can be triggered by:
  − Each module connected to the IR
  − External peripherals
  − Software via General Purpose Service Requests (GPSR)

› Each Service Provider supports **up to 255** service priority levels:
  − 0 to disable the interrupt
  − 255 for highest priority

# Introduction

IOM	SMU	IR	GTM

› The Generic Timer Module (GTM) is a modular timer unit designed to accommodate many timer applications.

› It has an in-built Advanced Router Unit (ARU) that can be used to exchange specific data between sub-modules without CPU interaction.

› The ARU-connected Timer Output Module (ATOM), which is part of the GTM, is able to generate complex output signals.

› 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.

Hardware setup

This code example has been developed for the board KIT_AURIX_TC297_TFT_BC-Step.
Connect the two pins P00.0 PWM signal and P33.1 IOM monitor input to each other.

X103

VCC_IN	1	2	V_UC(+5V)
GND	3	4	GND
P33.10	5	6	P33.9
P14.8	7	8	P14.7
P14.6	9	10	P10.6
P10.7	11	12	P10.4
P02.0	13	14	P02.1
P02.2	15	16	P02.3
P02.4	17	18	P02.5
P02.6	19	20	P02.7
P02.8	21	22	P00.0
P00.1	23	24	P00.2
P00.3	25	26	P00.4
P00.5	27	28	P00.6
P00.7	29	30	P00.8
P00.9	31	32	P00.10
P00.11	33	34	P00.12
AN45	35	36	AN44
AN17	37	38	AN16
AN25	39	40	AN24
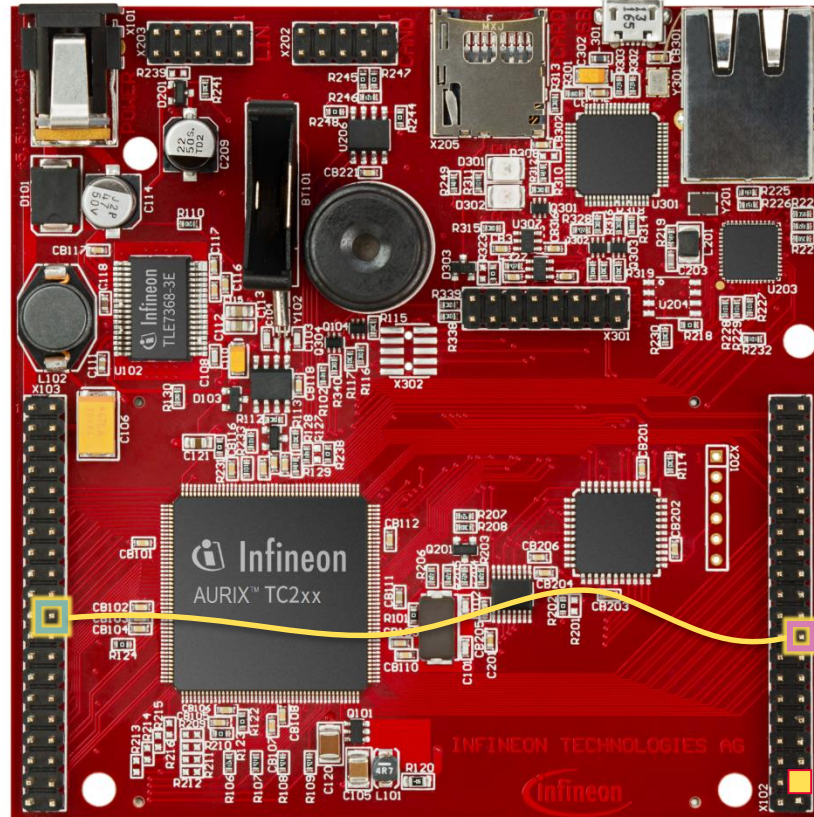
X102

P14.5	40	39	P14.4
P20.10	38	37	P20.9
P15.7	36	35	P15.6
P15.5	34	33	P15.4
P15.3	32	31	P15.2
P22.3	30	29	P22.2
P22.1	28	27	P22.0
P33.11	26	25	P23.4
P23.3	24	23	P23.2
P23.1	22	21	P23.0
P33.6	20	19	P33.8
P33.12	18	17	P33.1
P33.2	16	15	P33.3
P33.4	14	13	P33.5
AN0	12	11	AN8
AN2	10	9	AN3
AN32	8	7	AN33
AN20	6	5	AN21
GND	4		GND
V_UC(+5V)	2	1	VCC_IN

Implementation

① The GTM module generates a PWM signal

② The IOM monitors the PWM signal and compares it with an internal configured Event Window

③ If an event is generated, an SMU internal alarm is triggered

④ The IR handles the SMU alarm as software interrupt
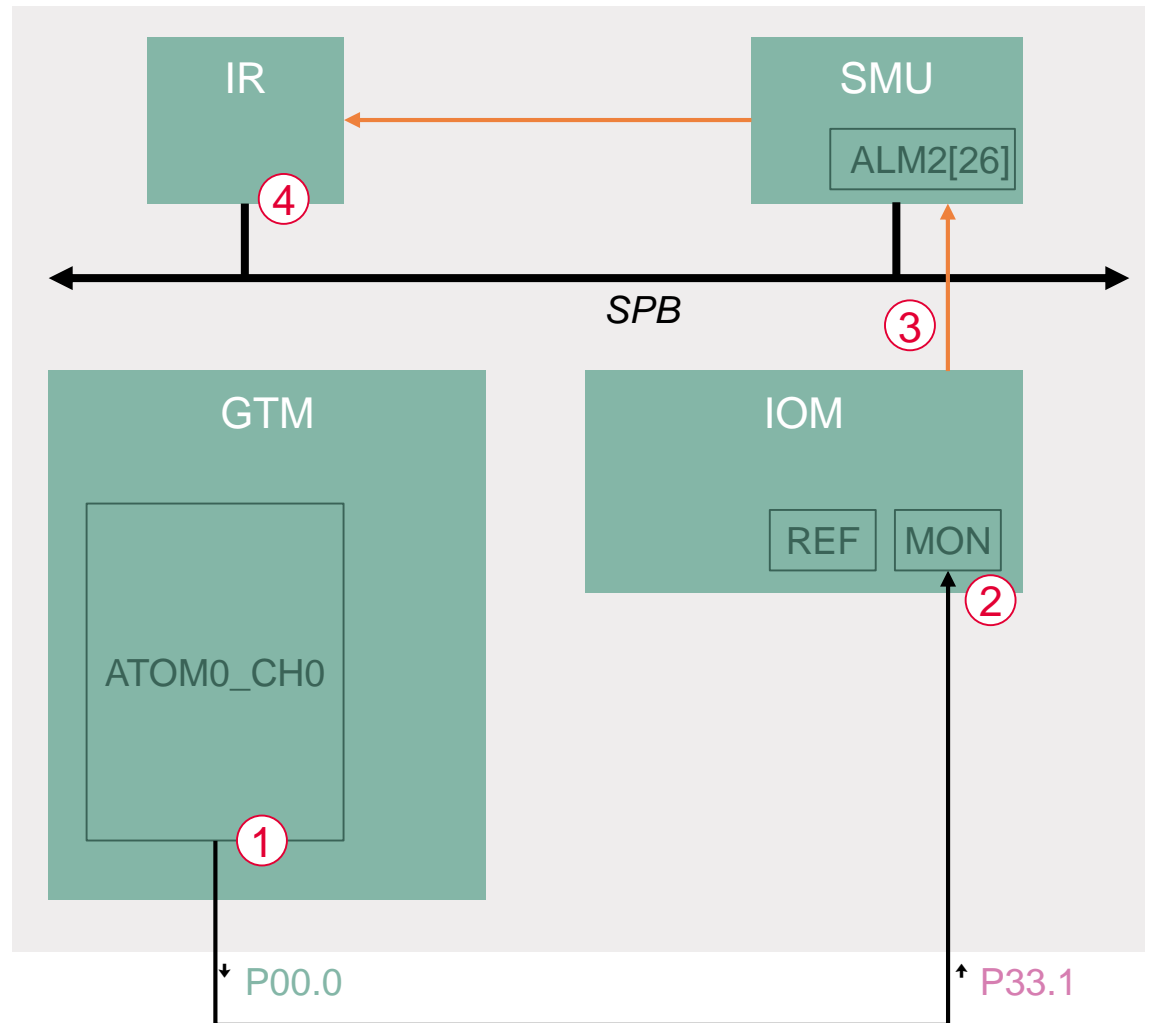
Implementation

Configuring the PWM signal from the GTM – *init_Gtm_Pwm()*

① GTM

The PWM signal is generated by the ATOM submodule. The following steps are performed at first in the example:
› Enable the GTM by calling the function ***IfxGtm_enable()***
› Set the CMU clock 0 frequency to 1 MHz with the function ***IfxGtm_Cmu_SetClkFrequency()***.
› Enable the CMU clock 0 by calling the function ***IfxGtm_Cmu_enableClocks()***.

Secondly, the function ***IfxGtm_Atom_Pwm_initConfig()*** initializes an instance of the structure ***IfxGtm_Atom_Pwm_Config*** with its default values. The user can then choose specific values for the following parameters:
› ***atom*** – Selection of the ATOM which is counting (ATOM 0 in this example)
› ***atomChannel*** – Selection of the ATOM channel (channel 0 in this example)
› ***period*** – Setting of the period for the PWM signal to the desired value (in µs)
› ***pin.outputPin*** – Selection of the port pin P00.0 as output pin
› ***synchronousUpdateEnable*** – Enabling of Synchronous Update of the timer
› ***dutyCycle*** – Selection of the PWM duty cycle length (in µs)

After applying the specified parameters via a call to ***IfxGtm_Atom_Pwm_init()***, the PWM starts by calling the function ***IfxGtm_Atom_Pwm_start()***.

The above functions can be found in the ***IfxGtm.h***, ***IfxGtm_Atom_Pwm.h*** and ***IfxGtm_Cmu.h***.

Implementation

Configuring the IOM – *init_Iom()*

Such as the other modules, the IOM needs at first to be enabled. This is done via the iLLD function ***IfxIom_enableModule()***.
Before starting to configure the parameters, the configuration drivers need to be initialized (see code for more detailed information):
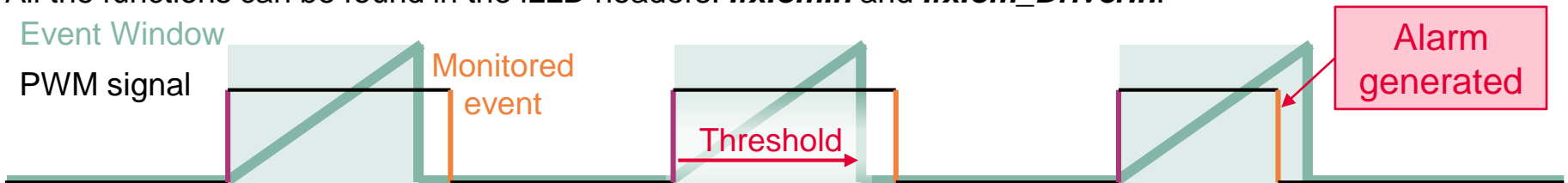› ***IfxIom_Driver*** – Declare IOM driver
› ***IfxIom_Driver_Config*** – Used to configure the IOM
› ***IfxIom_Driver_Lam*** – Declare LAM driver
› ***IfxIom_Driver_LamConfig*** – Used to configure the LAM

The configuration of the LAM continues by setting the following key parameters:
› ***mon.input*** selecting the monitor input port
› ***eventWindow.controlSource*** determining the signal generating the events
› ***eventWindow.clearEvent*** defining when to start a new event window
› ***eventWindow.threshold*** defining the event window size
› ***eventWindow.inverted*** defining whether the event should happened **before** or after the threshold
› ***event.source*** defining the sourced signal
› ***event.trigger*** defining on what is the triggering event
› ***systemEventTriggerThreshold*** setting the amount of events before generating an alarm

The configuration is thereafter applied to the module vie the iLLD function ***IfxIom_Driver_initLam()*** and the LAM module can be enabled with ***IfxIom_Driver_enableLamEvent()***.
All the functions can be found in the iLLD headers: ***IfxIom.h*** and ***IfxIom_Driver.h***.

Event Window
PWM signal
Monitored event
Threshold
Alarm generated

Implementation

Configuring the SMU – *init_Smu()*

The SMU is first enabled and started with the iLLD function ***IfxSmu_start()***.
To allow the configuration of the SMU, the module must be unlocked. This can be done with the iLLD function: ***IfxSmu_unlock()***.

The function ***IfxSmu_setAlarmConfig()*** has to be called with the following parameters:
› ***IfxSmu_Alarm_IomPinMismatchIndication*** is used to identify the alarm refering to the IOM.
› To configure the SMU behavior when an alarm occurs, an Interrupt Generation Configuration Set (IGCS) has to be selected. In this example, IGCS0 (IGCS1-2 are also available) identifies with ***IfxSmu_AlarmConfig_interruptSet0***.

Since IGCS0 has been selected, it has to be enabled in SMU Alarm Global Configuration (AGC) setting to 1 the bit 0 of the SMU_AGC register:

MODULE_SMU.AGC.U = 0x**1**;

SMU_AGC
Alarm Global Configuration

0	EFR ST	PES	0	ICS	0	IGCS2	0	IGCS1	0	IGCS0	1

31 30 29   2   1   0

Lastly, the SMU module has to be relocked with the function ***IfxSmu_lock()***.
All the above functions are contained in the header file ***IfxSmu.h***.

Implementation

Configuring the Interrupt Router – *init_IR()* ④ IR

To handle the Software interrupt generated by the SMU module, the Service Request Register (SRC) from the SMU0 (identifying the SMU Interrupt Generation Configuration Set 0) has first to be fetched.
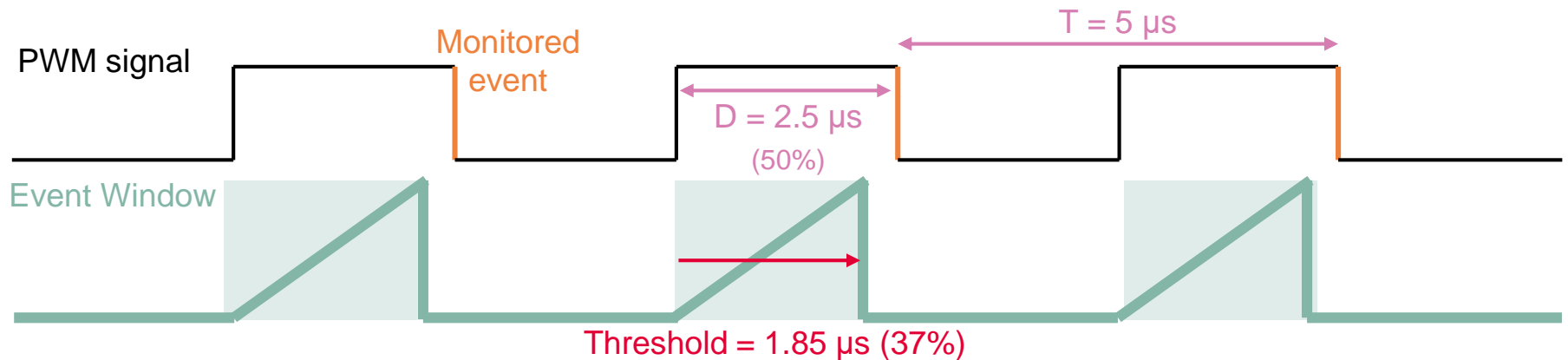Then calling ***IfxSrc_init()*** allows:
› The selection of the Service Provider (CPU0 in this example)
› The interrupt priority level (10 in this example)

The function ***IfxSrc_init()*** can be found in the header file ***IfxSrc.h***.

Example workflow

› Starting conditions:

PWM signal
Monitored event
T = 5 µs
D = 2.5 µs (50%)
Event Window
Threshold = 1.85 µs (37%)

› The duty cycle is **decreased by 5%** every second.

› When the duty cycle of the monitored PWM signal goes **below the accepted threshold of 37%**, the alarm is triggered and an LED is switched on.

› When the duty cycle reaches 25%, it is set back to 50% and the example routine starts again.

Run and Test

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

1. Make sure that the port pin P00.0 (PWM signal) and P33.1 IOM (monitor input) are connected to each other.

2. (optional)
    An oscilloscope can be used to observe the behavior of the PWM signal and that the LED is triggered when the duty cycle goes below 37%.

3. LED D109 is on when the monitored PWM signal's duty cycle is not respecting the mininum 37% ratio.

D = **40%**
T = 5µs

D109 ›
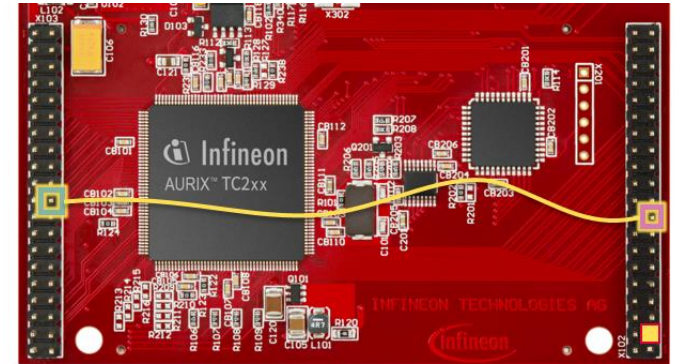
References

› 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

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.