

# IOM\_Signals\_Comparison\_1 for KIT\_AURIX\_TC297\_TFT

Comparison of signals via IOM

AURIX™ TC2xx Microcontroller Training  
V1.0.0



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

## Scope of work

---

### **Two PWM signals are compared by the IOM module.**

Two PWM signals are routed through the port pins to the Input Output Module (IOM) as monitor and reference respectively. The signals are continuously compared using exclusive OR. If the newly formed signal (XOR) is in high state outside the acceptable propagation window, 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

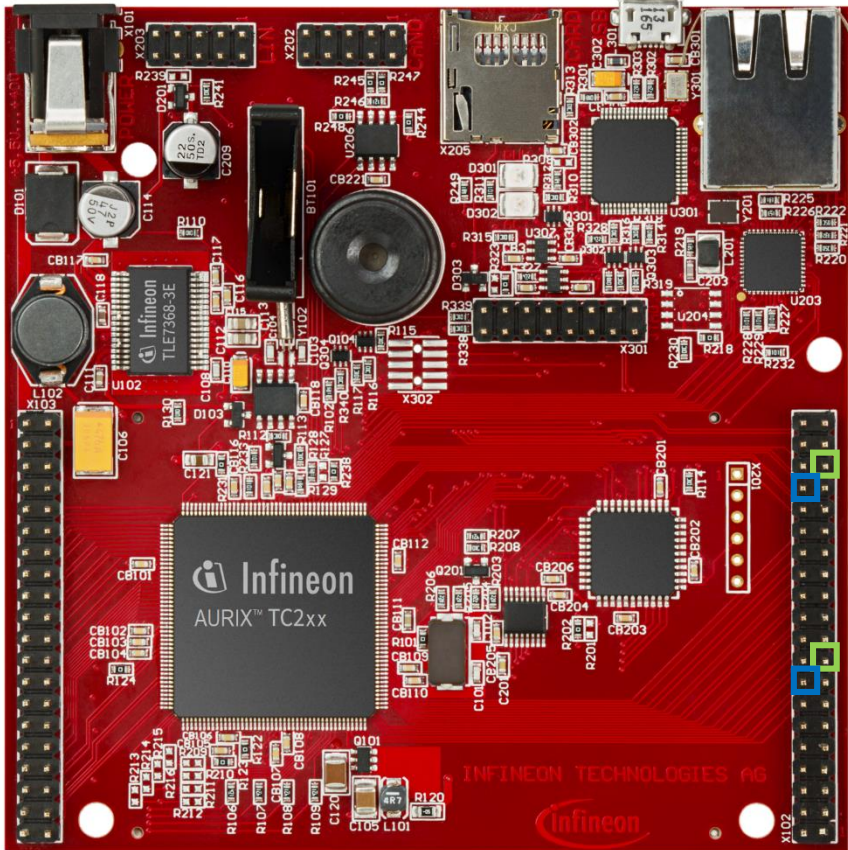
SMU

IR

GTM

- › The Generic Timer Module (GTM) is a modular timer unit designed to accommodate many timer applications.
- › It has an in-built Timer Output Module (TOM) that can offers 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\_AURIX\_TC297\_TFT\_BC-Step.

For the first signal, connect the two port pins **P15.6 PWM signal** and **P33.1 IOM reference input** together. For the second signal, connect the two port pins **P15.5 PWM signal** and **P33.2 IOM monitor input** to each other.

	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	3	GND
V_UC(+5V)	2	1	VCC_IN





# Implementation

## Configuring the TOM

① GTM

- › The ***lfxGtm\_Tom\_Pwm\_Config*** structure allows to set the following parameters to initialize the module:
  - ***tom*** – Selection of the TOM which is counting (TOM 0 and TOM 1 in this training)
  - ***tomChannel*** – Selection of the channel which is driving the LED (Channel 0 in this training)
  - ***period*** – Setting of the period for the PWM signal to the desired value
  - ***pin.outputPin*** – Selection of the LED as the output pin
  - ***synchronousUpdateEnable*** – Enabling of Synchronous Update of the timers
  
- › After configuration, the function ***lfxGtm\_Tom\_Pwm\_init()*** initializes and activates the TOM with the user configuration
  
- › Start the PWM with the function ***lfxGtm\_Tom\_Pwm\_start()***

All the functions used for the configuration of the TOM are provided by the iLLD header ***lfxGtm\_Tom\_Pwm.h***.

# Implementation

---

## Configuring the IOM – *initIOM()*

② IOM

Such as the other modules, the IOM needs at first to be enabled. This is done via the iLLD function ***ifxlom\_enableModule()***.

Before starting to configure the parameters, the configuration drivers need to be initialized (see code for more detailed information):

- › ***ifxlom\_Driver*** – Declare IOM driver
- › ***ifxlom\_Driver\_Config*** – Used to configure the IOM
- › ***ifxlom\_Driver\_Lam*** – Declare LAM driver
- › ***ifxlom\_Driver\_LamConfig*** – Used to configure the LAM

# Implementation

## Configuring the IOM – *initIOM()*

② IOM

Then, the configuration of the LAM regarding the example continues by setting the key parameters as following:

- › ***mon.input*** selecting the monitor input port
- › ***ref.input*** selecting the reference 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 happen before or after the threshold
- › ***event.source*** defining the sourced signal
- › ***event.trigger*** defining on which condition the event is triggered
- › ***systemEventTriggerThreshold*** setting the amount of events before generating an alarm

The configuration is thereafter applied to the module through the iLLD function ***ifxlom\_Driver\_initLam()*** and the LAM module can be enabled with ***ifxlom\_Driver\_enableLamEvent()***.

# Implementation

## Configuring the SMU – *initSmu()*



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 referring 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 = 0x1;
```

SMU\_AGC  
Alarm Global Configuration



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 – *initIR()*



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 ***lfxSrc\_init()*** allows:

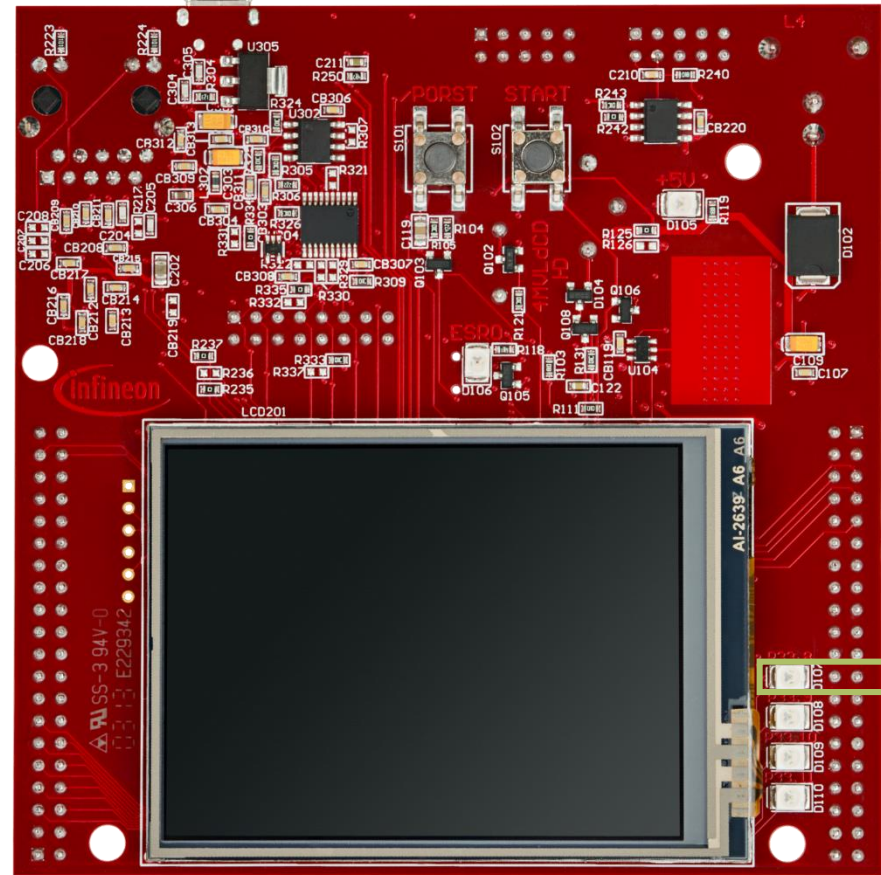
- › The selection of the Service Provider (CPU0 in this example)
- › The setting of the interrupt priority level (10 in this example)

The function ***lfxSrc\_init()*** can be found in the header file ***lfxSrc.h***.

# Run and Test

After code compilation and flashing the device, observe the behavior of the LED:

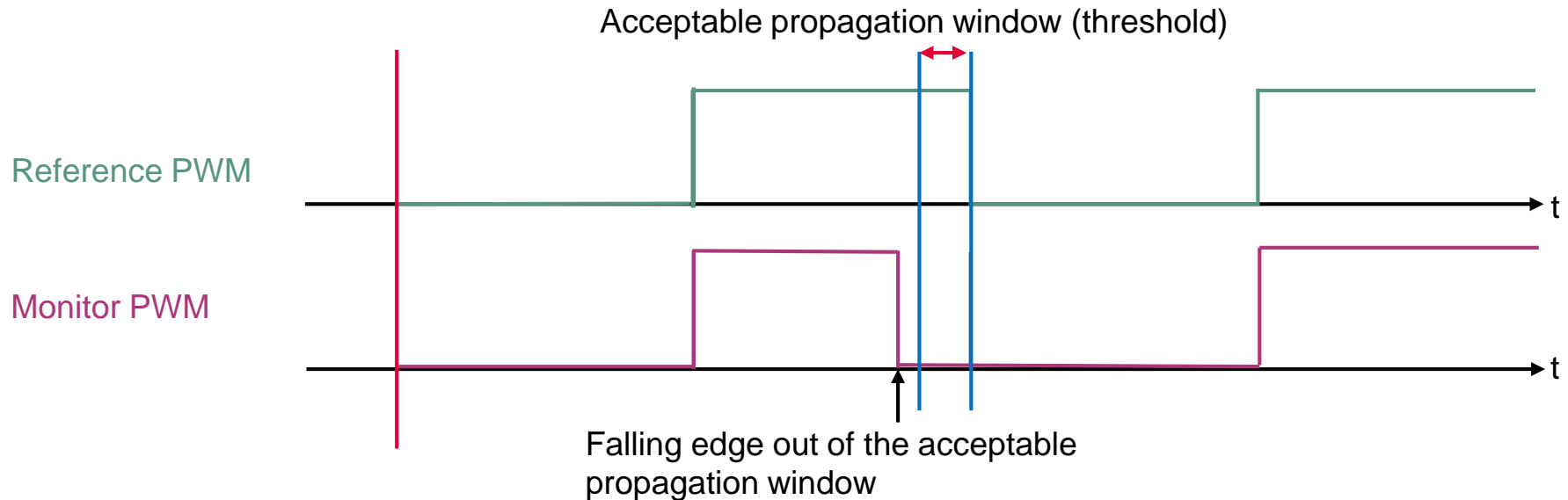
- › Check if the LED D107 (1) is switched off when the duty cycle of the monitored PWM is inside the acceptable propagation window and switched on when outside.



# Run and Test

## Graphic explanation

When the falling edge of the monitored PWM is out of the acceptable propagation window, an event is generated, triggering an internal SMU alarm.





# 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](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 2020-02**

**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**

**© 2020 Infineon Technologies AG.**

**All Rights Reserved.**

**Do you have a question about this document?**

**Email: [erratum@infineon.com](mailto:erratum@infineon.com)**

**Document reference**

**IOM\_Signals\_Comparison\_1**

**\_KIT\_TC297\_TFT**

## 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](http://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.