

# PMS\_Power\_Down\_Idle\_1 for KIT\_AURIX\_TC334\_LK

Power saving in CPU idle mode

AURIX™ TC3xx Microcontroller Training  
V1.0.0



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

## Scope of work

---

**CPU0 switches periodically between run mode and idle mode triggered by STM timer.**

The Power Management System (PMS) allows activation of power down modes for each CPU individually in case there are no active tasks to perform and power consumption needs to be reduced. In this example, the System Timer periodically generates interrupts (every two seconds). In the Interrupt Service Routine, the CPU0 is switched between idle mode and run mode. CPU0 toggles LED1 during run mode whereas the LED1 remains in the last toggling state while CPU0 is idle.

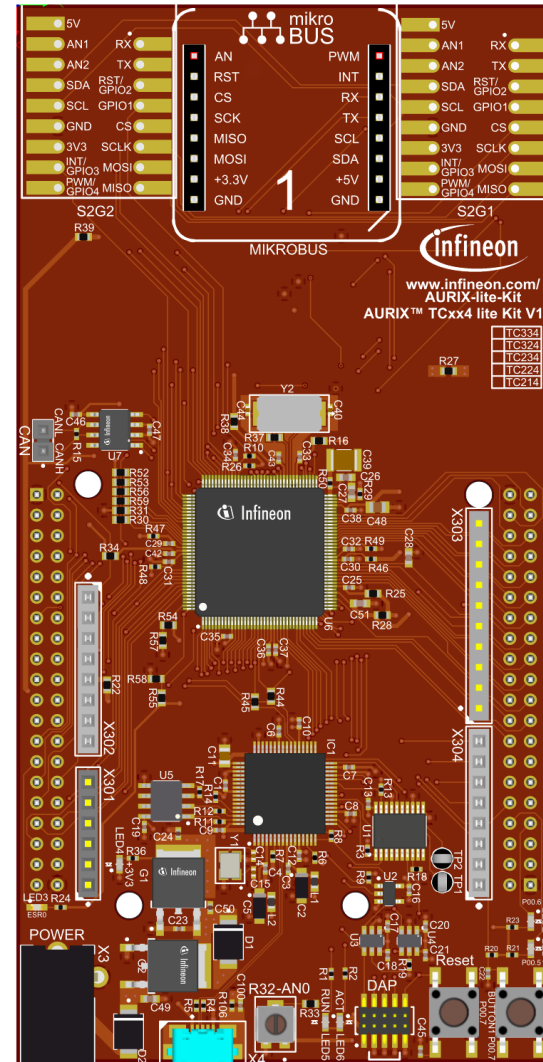
# Introduction

---

- › The **Power Management System** allows activation of power down modes so that the system operates with the minimum required power for a corresponding application state
- › **Idle, Sleep** or **Standby** are supported as power consumption modes. The **Idle** is specific to each CPU whereas the **Sleep** and **Standby** modes influence the complete system
- › The System Timer (STM) is a free running 64-bit counter which can be used for timing applications requiring both high precision and long period
- › The STM has the capability to **generate interrupts** when its count reaches a predefined **compare value**

# Hardware setup

This code example has been developed for the board KIT\_A2G\_TC334\_LITE.



# Implementation

---

## Configuring the system timer to periodically generate interrupts:

- › The configuration of the system timer is done through the function ***configSystemTimer()***.
  - The system timer immediately runs after PowerUp and needs to be configured
  - Create an instance of the structure ***lfxStm\_Timer\_Config***, which is initialized by calling ***lfxStm\_Timer\_initConfig()***. Then configure the following components:
    - Set the rate in Hz that refers to interrupt events per second
    - Define the priority for the interrupt (range from 0 to 255)
    - Define the provider for the interrupt service request as one of the CPUs or the DMA
    - Specify the comparator register that is used to store the compare value to trigger an event when timer and comparator register match
  - Use the routine ***lfxStm\_Timer\_init()*** together with the instance of structure ***lfxStm\_Timer\_Config*** to create the ***g\_myTimer*** handle

The above functions can be found in the header ***lfxStm\_Timer.h***.

# Implementation

---

## Blink the LED1:

- › CPU0 runs within the endless loop in **Cpu0\_Main.c**. Before it starts a toggle cycle, it waits for approximately 4 million CPU ticks (equals to 0x400000)
- › If the CPU0 is set to idle state, the code execution is halted and the CPU clock is disabled. This stops the toggling of the LED

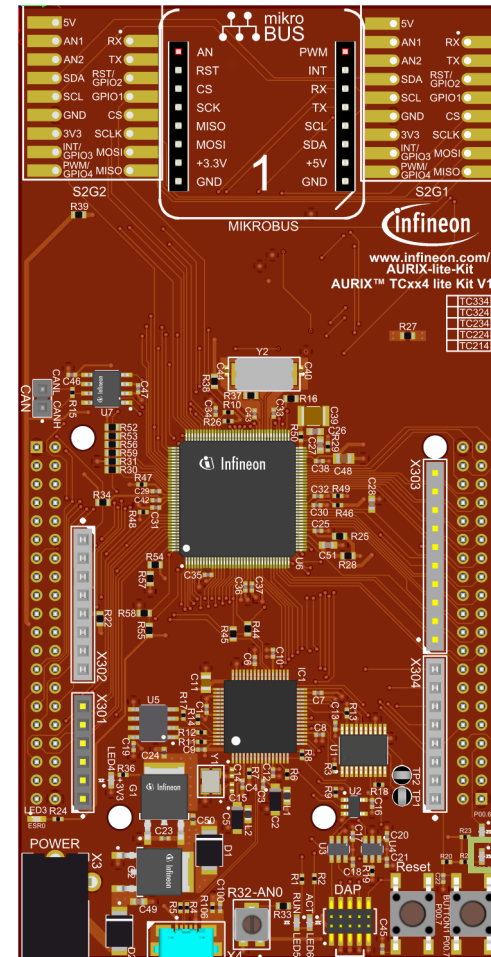
## Set/Return to/from Idle state:

- › The interrupt service routine defined in **PMS\_Power\_Down\_Idle.c** controls the CPU0, which alternates between run state and idle state. The static variable **setidle** helps to identify the next state of the CPU0, whether it remains in run state or is set into idle state
- › If the CPU0 is in idle state and the STM interrupt event occurs then the CPU returns to run state
- › The function **IfxStm\_Timer\_acknowledgeTimerIrq()** is used to acknowledge the interrupt request and set the next compare value for the next interrupt event. The function can be found in the header **IfxStm\_Timer.h**

# Run and Test

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

- › Check if LED1 (1) is blinking (equals to Run Mode) or remains on/off (equals to Idle Mode)



1

# 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 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](mailto:erratum@infineon.com)

## Document reference

**PMS\_Power\_Down\_Idle\_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](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.