# RAM_Run_Function_1 for KIT_AURIX_TC275_LK

Function running from RAM

AURIX™ TC2xx Microcontroller Training
V1.0.0

**A function is stored and executed from SRAM.**

This example implements twice the same function which toggles an LED with a wait loop. One function is implemented to be executed from SRAM and the other one from Flash memory.

The SRAM function is toggling LED1 (P00.5), while the flash function is toggling LED2 (P00.6).

# Introduction

› The Local Memory Unit (LMU) SRAM can be used for **code execution**, data storage or overlay memory

› The LMU can be accessed via cached (segment $9_H$) or via non-cached (segment $B_H$) memory addresses

› If a code is programmed to be executed from SRAM memory, it is copied from Flash to SRAM by the Start-up Software (SSW) code

# Hardware setup

This code example has been developed for the board KIT_AURIX_TC275_LITE.
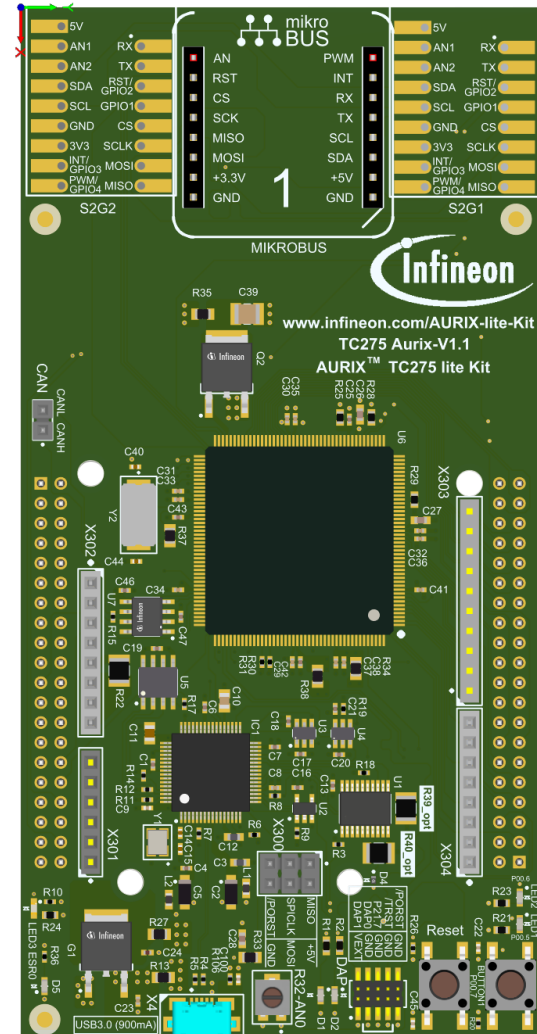
# Implementation

**SRAM code section creation**

The linker file "***Lcf_Tasking_Tricore_Tc.lsl***" is updated by adding a memory section (called ***code_lmuram_nc***) for code execution from LMURAM memory.
The memory section should be assigned to the **non-cached** memory addresses (segment $B_H$) to avoid any data inconsistency.

```
group code_lmuram_nc (ordered, attributes=rwx, copy, run_addr=mem:lmuram/not_cached)
{
    select "(.text.not_cached_lmuram*)";
    select "(.text.lmuram_not_cached*)";
}
```

**Locating function code in a specific memory section**

The ***pragma*** compiler keyword with the attribute ***section code "<section_identifier>"*** is used to specify the memory section from which the implemented function code will be fetched and executed.

The ***section code restore*** attribute is used after the function implementation to ensure that next implemented functions will be located in the default code memory section (Flash memory).

# Implementation

**LED Toggling**

Two functions are implemented, *toggleLedSram()* and *toggleLedFlash()*, to toggle two LEDs from different memory regions.
Using the previously mentioned *pragma* compiler keyword, the *toggleLedSram()* can be executed from LMURAM memory not-cached addresses segment.

Both functions are implemented as following:
- Switch On the LED by calling *IfxPort_setPinLow()*
- Wait for a one second delay
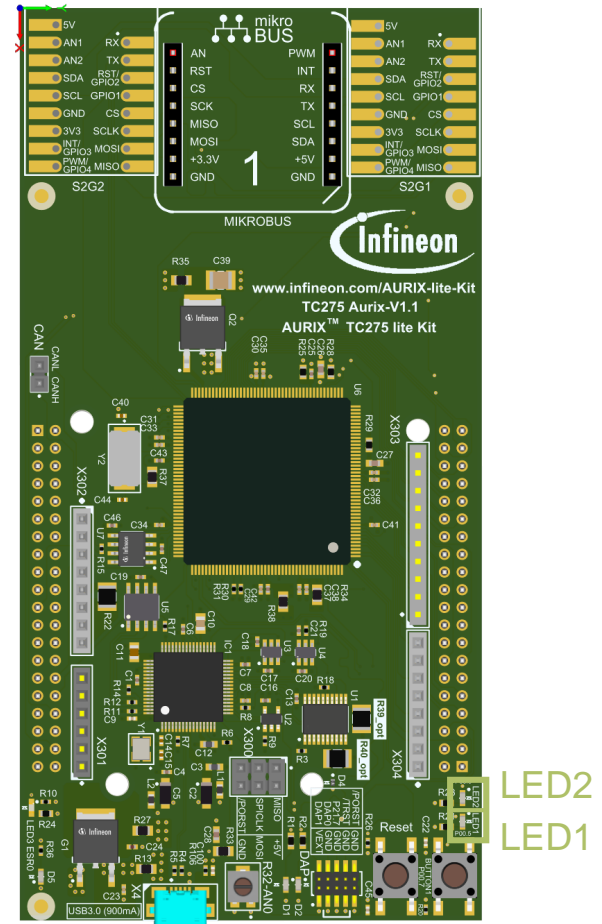- Switch Off the LED by calling *IfxPort_setPinHigh()*
- Wait for a one second delay

The above port functions can be found in the iLLD header *IfxPort.h*.

**Note:** The LEDs on the used board are low-level active.
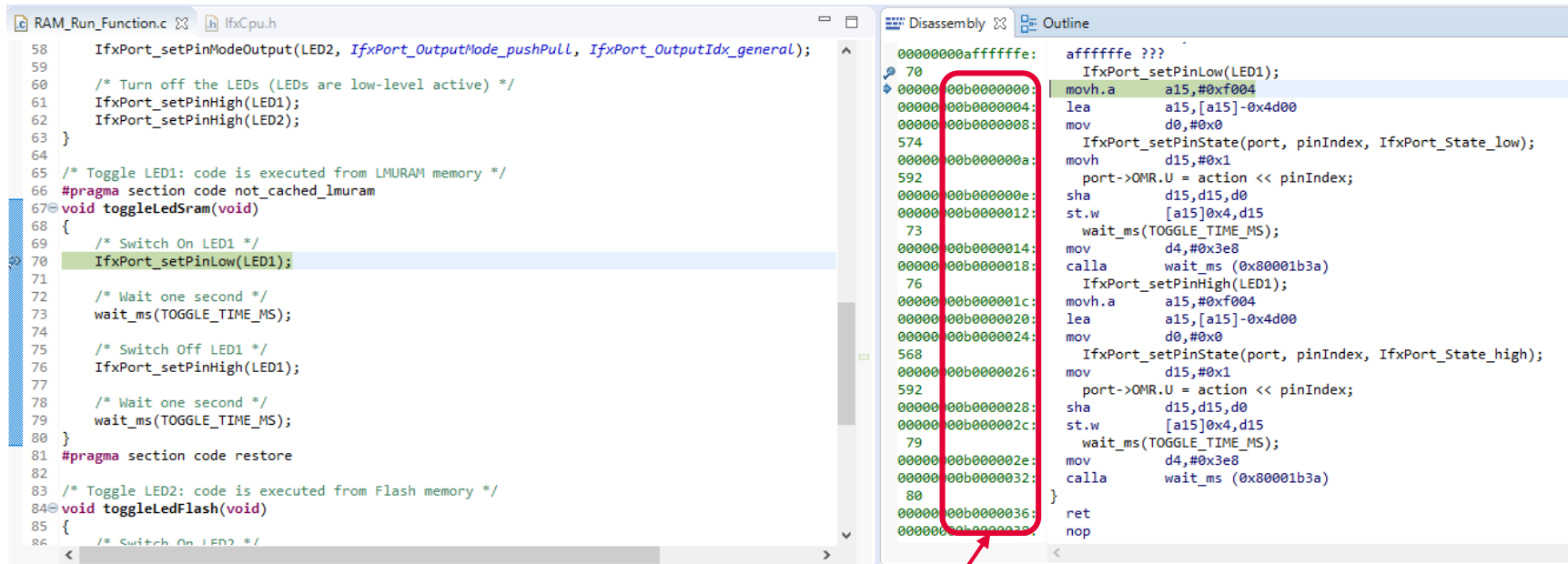
# Run and Test

After code compilation and flashing the device, check the following behavior

› Check that LED1 and LED2 are toggling

# Run and Test

Additionally, the execution from RAM can be checked by adding a breakpoint inside the ***toggleLedSram()*** function and verify in the disassembly window of the debugger that the CPU is executing it from LMURAM (Addresses segment $B_H$).



› Addresses from where the ***toggleLedSram()*** function is executed

# 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.