Run_Function_RAM_1 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 (P13.0), while the flash function is toggling LED2 (P13.1).



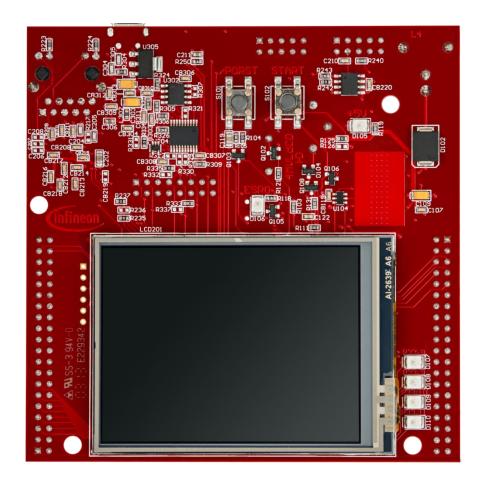
Introduction

- 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_TC297_TFT_BC-Step.







SRAM code section creation

The linker file "*Lcf_Tasking_Tricore_Tc.lsl*" is updated by adding a memory section (called *code_Imuram_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.

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).





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*.

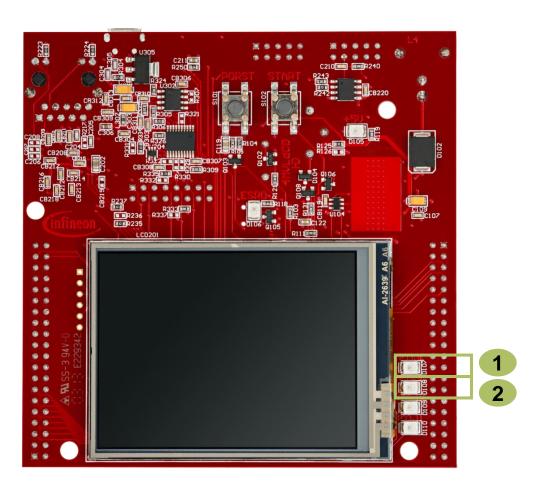
Note: The LEDs on the used board are low-level active.



Run and Test

After code compilation and flashing the device:

 Check that LED1 (D107) and LED2 (D108) 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).

```
Fnter location he V 8 6 6 8 6 13 13
            RAM_FUNCTION.c 
☐ IfxCpu.h
                                         Lcf Tasking ...

    Disassembly 
    B Outline

       IfxPort_setPinHigh(LED2);
                                                                             00000000afffffe:
69 }
                                                                             00000000afffffff:
                                                                                                             Unable to retrieve disassembly data from backend.
70
                                                                                                                IfxPort setPinLow(LED1);
71 /* Toggle LED1: code is executed from LMURAM memory */
                                                                                                0xff004091

    0000000 0b0000000

                                                                                                                          a15,#0xf004
72 #pragma section code not cached 1muram
                                                                             0000000 0b0000004
                                                                                                0xcb00ffd9
                                                                                                                           a15,[a15]-0x4d00
73 void toggleLedSram(void)
                                                                             0000000 0b0000008
                                                                                                0x00000082
74 {
                                                                                                                IfxPort_setPinState(port, pinIndex, IfxPort_State_low);
        /* Switch On LED1 */
75
                                                                             0000000 0b000000a
                                                                                                0xf000107b
                                                                                                                          d15.#0x1
76
       IfxPort setPinLow(LED1);
                                                                                                                port->OMR.U = action << pinIndex;
77
                                                                             0000000 0b000000e
                                                                                                0xf0100f0f
                                                                                                                          d15,d15,d0
                                                                                                              sha
78
       /* Wait one second */
                                                                             0000000 0b0000012
                                                                                                0x0000f16c
                                                                                                                           [a15]0x4,d15
79
       wait ms(TOGGLE TIME MS);
                                                                                                                wait_ms(TOGGLE_TIME_MS);
80
                                                                             0000000 0b0000014
                                                                                                0x403e803b
                                                                                                                          d4,#0x3e8
81
       /* Switch Off LED1 */
                                                                             0000000 0b0000018
                                                                                                0x0d9d80ed
                                                                                                                          wait ms (0x80001b3a)
                                                                                                              calla
       IfxPort_setPinHigh(LED1);
                                                                                                                IfxPort_setPinHigh(LED1);
                                                                              82
83
                                                                             0000000 0b000001c
                                                                                                0xff004091
                                                                                                                          a15,#0xf004
                                                                                                              movh.a
84
       /* Wait one second */
                                                                             0000000 0b0000020
                                                                                                0xcb00ffd9
                                                                                                                           a15,[a15]-0x4d00
85
       wait_ms(TOGGLE_TIME_MS);
                                                                             0000000 0b0000024
                                                                                                0x00000082
86 }
                                                                                                                IfxPort_setPinState(port, pinIndex, IfxPort_State_high)
87 #pragma section code restore
                                                                             594
                                                                                                                port->OMR.U = action << pinIndex;
89 /* Toggle LED2: code is executed from Flash memory */
                                                                             0000000 0b0000028
                                                                                                0xf0100f0f
                                                                                                                          d15,d15,d0
90 void toggleLedFlash(void)
                                                                             0000000 0b000002c
                                                                                                0x0000f16c
                                                                                                                          [a15]0x4,d15
91 {
                                                                                                                wait ms(TOGGLE_TIME_MS);
        /* Switch On LED2 */
                                                                             0000000 0b0000002e
                                                                                                0x403e803b
                                                                                                                          d4,#0x3e8
93
       IfxPort_setPinLow(LED2);
                                                                             0000000 0b0000032
                                                                                                0x0d9d80ed
                                                                                                              calla
                                                                                                                          wait ms (0x80001b3a)
   <
```

Addresses from where the toggleLedSram() function is executed

References







- 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 2020-01 Published by Infineon Technologies AG 81726 Munich, Germany

© 2020 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: <u>erratum@infineon.com</u>

Document reference Run_Function_RAM_1

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.