

Assembly_C_Code_1 for KIT_AURIX_TC397_TFT

Assembly language in C code

AURIX™ TC3xx Microcontroller Training
V1.0.1



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

Scope of work

Inline assembler and assembler files are used in combination in a C project.

Two LEDs are switched on then switched off using assembly code functions.

Introduction

- › The TASKING compiler within the AURIX™ Development Studio offers the possibility to use the assembly language inside the project code.
- › The assembly language is based on implementing code with the CPU instruction set, TriCore™ in this case.
- › This hardware oriented method allows the application to be memory efficient and faster in term of execution time comparing to higher level programming languages (C, C++, ...).
- › Assembly code can be implemented both inside dedicated source files „*.src“ and in C source files using the `__asm()` keyword.

- › Useful Documentation:
 - TriCore™ instruction set:
 - “TriCore_TC162P_core_architecture_vol2of2_Instruction_set.pdf”
 - Tasking Compiler Assembly language:
 - “ctc_user_guide.pdf”

Implementation

LED1 state control

LED1 driven by port 13 pin 0 can be switched ON/OFF by calling the assembly function ***set_LED1_State_Assembly()***, implemented in the assembly source file ***Assembly_Code.src***.

This functionality is ensured by the following steps:

- › Check the value passed to the function:
 - If 0 (LED_OFF): write 0x1 to D0 data register
 - Else 1 (LED_ON): write 0x10000 to D0 data register
- › Load the Port 13 Output Modification Register (OMR) into A0 address register
- › Store D0 value into A0 address register (Port 13 OMR register)
- › Return from function, needed to restore the context of the caller function

Implementation

LED2 state control

LED2 driven by port 13 pin 1 can be switched ON/OFF by calling the assembly function ***set_LED2_State_Assembly()***, implemented in the C source ***Assembly_C_Code.c*** file using the ***__asm()*** keyword.

This functionality is ensured by the following steps:

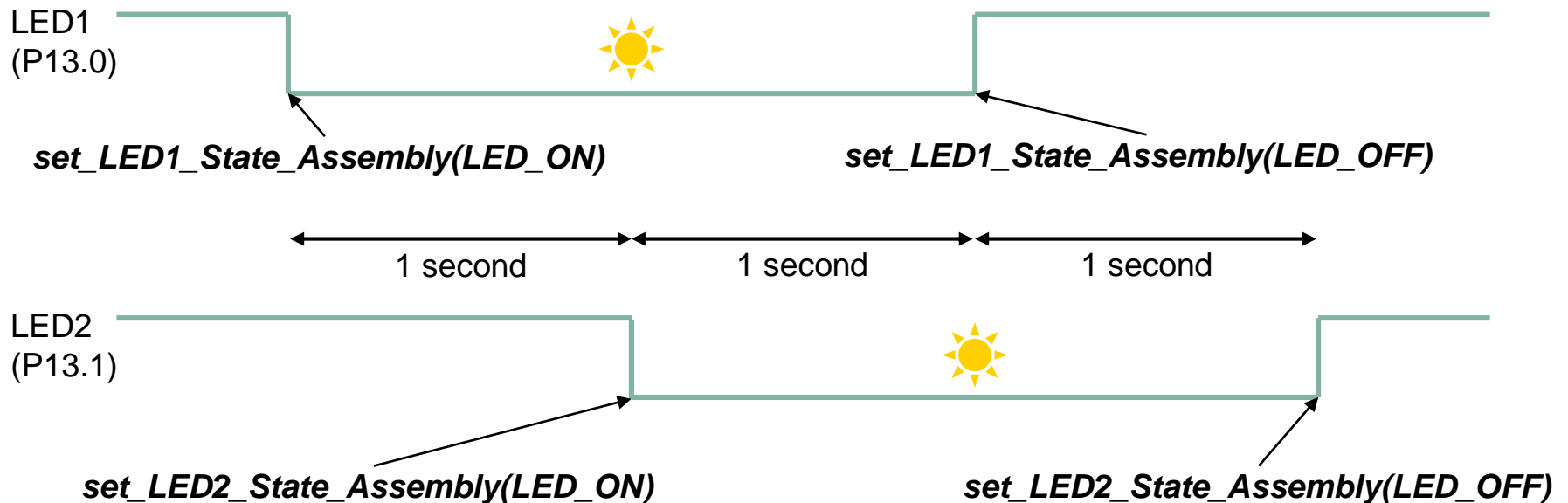
- › Check the value passed to the function:
 - If 0 (LED_OFF): write 0x2 to D0 data register
 - Else 1 (LED_ON): write 0x20000 to D0 data register
- › Load the Port 13 OMR register into A0 address register
- › Store D0 value into A0 address register (Port 13 OMR register)

Note: the return instruction is not needed in this case, because the assembly code is called inside a C code, this means the C compiler is handling restoring the context.

Implementation

Scenario:

- › P13.0 and P13.1 are configured to control respectively LED1 (D107) and LED2 (D108) using the ***ifxPort_setPinMode()*** iLLD function.
- › Then both LED1 and LED2 are switched off using the ***ifxPort_setPinHigh()*** iLLD function, please note that the LEDs are low-level active.
- › Afterwards the following is implemented:

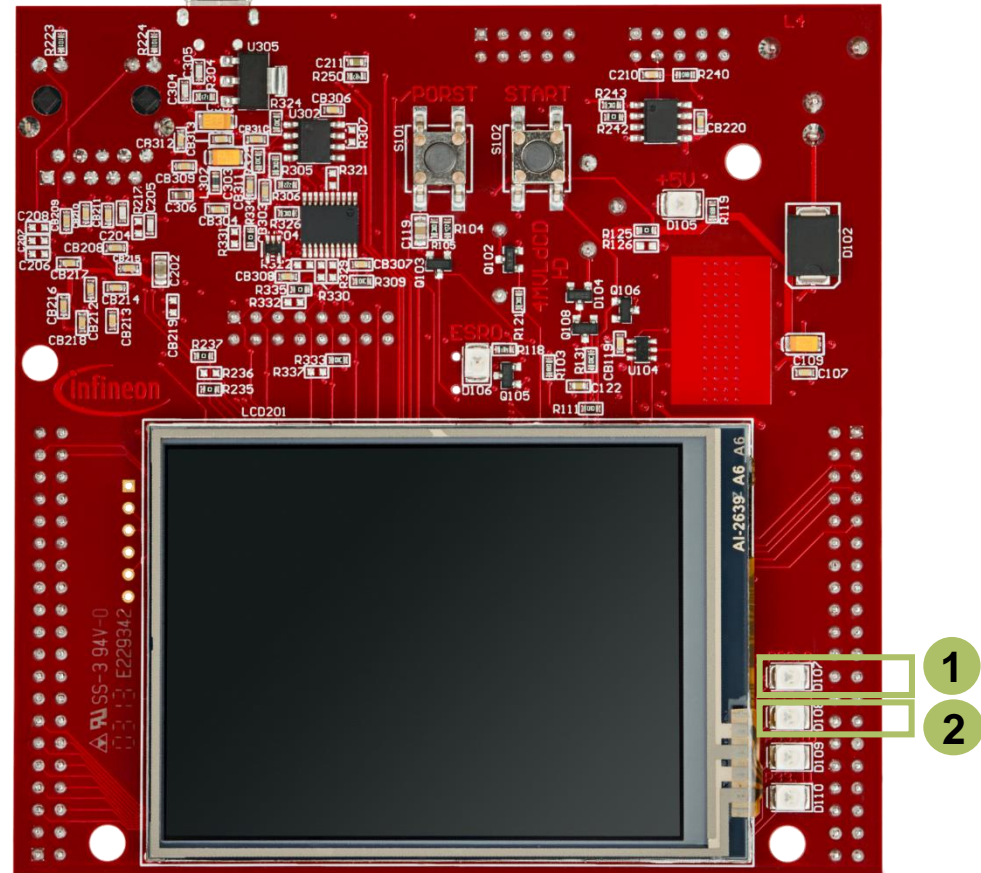


Run and Test

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

- › Firstly, LED1 (D107) is switched **on**
- › One second after, LED2 (D108) is switched **on**
- › One second after, LED1 (D107) is switched **off**
- › One second after, LED2 (D108) is switched **off**

For more details, please refer to the [previous slide](#).



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>

Revision history

Revision	Description of change
V1.0.1	Update of version to be in line with the code example's version
V1.0.0	Initial version

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2020-12

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

Document reference

Assembly_C_Code_1_KIT_TC397_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).

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.