# FCE_CRC_1
# for KIT_AURIX_TC297_TFT
## FCE CRC calculation

AURIX™ TC2xx Microcontroller Training
V1.0.0

# Scope of work

**The FCE module is used to calculate the CRC of the same message with different algorithms.**

This training shows how to configure the FCE to calculate CRC of a known message with all supported algorithms (CRC32, CRC16 and CRC8).
The FCE interrupt is enabled to report execution errors.
Any CRC kernel calculation error is indicated by switching ON the related LED.

# Introduction

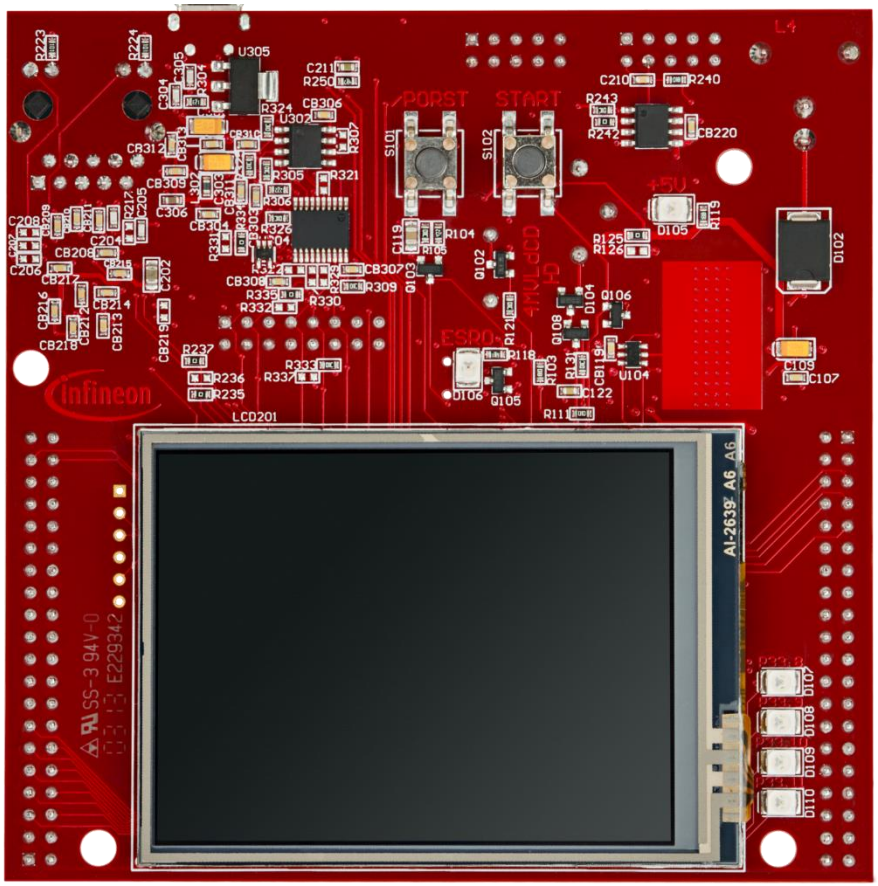
- The Flexible CRC Engine (FCE) provides a parallel implementation of Cyclic Redundancy Code (CRC) algorithms.

- FCE module supported algorithms:
  - IEEE 802.3 Ethernet CRC32 polynomial
  - CCITT CRC16 polynomial
  - SAE J1850 CRC8 polynomial

- CRC algorithms are used to calculate message signatures that can be used to check message integrity during transport over communication.

# Hardware setup

This code example has been developed for the board KIT_AURIX_TC297_TFT_BC-Step.



# Implementation

**Demo Initialization**

The initialization of the module is done via ***Fce_Crc_Init()***, which contains:

- the FCE module initialization, using the function ***IfxFce_Crc_initModule()***
- the CRC algorithms initialization, using the function ***IfxFce_Crc_initCrc()***

**Demo execution**

The execution is started with the function ***Fce_Crc_Demo_Run()***, which calculates:

- CRC32 algorithm, using ***IfxFce_Crc_calculateCrc32()***
- CRC16 algorithm, using ***IfxFce_Crc_calculateCrc16()***
- CRC8 algorithm, using ***IfxFce_Crc_calculateCrc8()***

All functions, needed for using the FCE CRC calculation, are provided by the iLLD header ***IfxFce_Crc.h***.

**FCE Error Interrupt Service Routine**

The ISR will be executed in case of a CRC calculation error. It will scan all kernels status registers and check if error flags are set.

# Implementation

**Configure and control the LEDs**

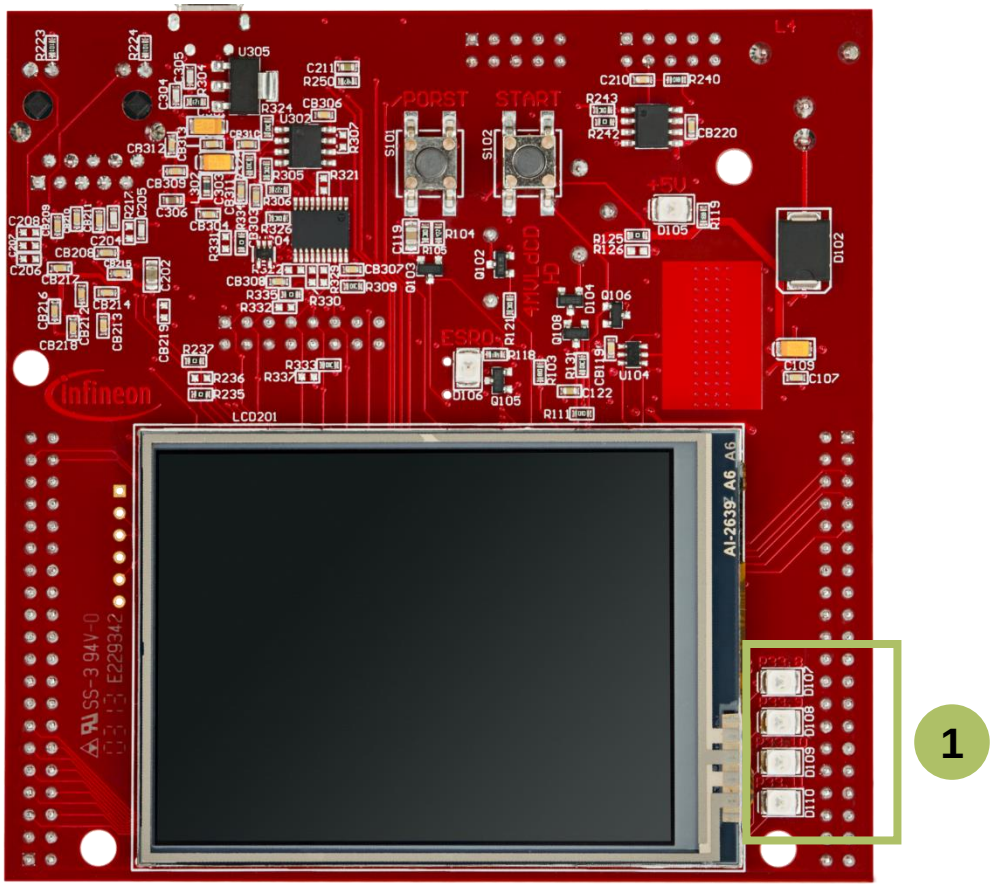
Four LEDs are configured to be switched on/off by the **controlling port pins** to which they are connected using methods from the iLLD header ***IfxPort.h***.

In the setup phase, the port pins of the LEDs have to be **configured as push-pull output** using the function ***IfxPort_setPinMode()***.

If CRC calculation errors occur, the related LEDs are **switched on** using the function ***IfxPort_setPinLow()***.

# Run and Test

After code compilation and flashing the device, observe the LEDs' behavior. All LEDs (1) should be switched **Off** if all CRC algorithms calculation are correct (Result = Expected)



# Run and Test

- LEDs behavior in case of Error:
  - CRC32 Kernel 0 calculation error: **D107** switches On
    - Could be tested by setting the macro ***CRC32_KERNEL0_WRONG_CHECK_VAL*** to 1
  - CRC32 Kernel 1 calculation error: **D108** switches On
    - Could be tested by setting the macro ***CRC32_KERNEL1_WRONG_CHECK_VAL*** to 1
  - CRC16 calculation error: **D109** switches On
    - Could be tested by setting the macro ***CRC16_WRONG_CHECK_VAL*** to 1
  - CRC8 calculation error: **D110** switches On
    - Could be tested by setting the macro ***CRC8_WRONG_CHECK_VAL*** to 1

- All macros mentioned above are provided only for test purpose. They allow to pass a wrong expected value to the FCE CRC kernel which leads to a mismatch with the calculated one, therefore the error flag will be set and the error interrupt will be triggered.

# 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

**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**



**Do you have a question about this document?**
**Email: erratum@infineon.com**

**Document reference**
**FCE_CRC_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).

**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.