

MPC57xx - Lauterbach FCCU utility

By: Peter Vlna

1. Introduction

This document describes how to use Lauterbach FCCU (fault collection and control unit) periphery extension for MPC57xx devices. It is expected that user has deep knowledge on FCCU mechanisms in order to effectively use this extension.

Contents

1.	Introduction.....	1
2.	Installation into Trace32	2
2.1.	Modifying the start script.....	2
3.	FCCU utility	3
3.1.	Clear FCCU/RGM	3
3.2.	FCCU Read/Clear Faults	4
3.3.	FCCU_CFG	7
3.4.	FCCU_NCF_CFG	8
3.5.	FCCU_NCFS_CFG	9
3.6.	Noncritical Fault Enable (FCCU_NCF_En)	10
3.7.	Noncritical Fault Timeout Enable (FCCU_NCF_TOEn).....	11
3.8.	Noncritical Fault Timeout (FCCU_NCF_TO)	12
3.9.	Delta T (FCCU_DELTA_T).....	13
3.10.	IRQ Alarm Enable (FCCU_IRQ_ALARM_ENn)	14
3.11.	NMI Enable (FCCU_NMI_ENn).....	15
3.12.	Bistable Fault-Output (EOUT) Mode Signaling Enable (FCCU_EOUT_SIG_ENn)	16



2. Installation into Trace32

Following file and folder must be copied into Trace32 main directory. Usually it is C:/T32

- FCCU_UTILITY.cmm
- custom_scripts

2.1. Modifying the start script

In order to start FCCU utility at every start of Trace32 it is necessary to modify the Trace32 start script.

1. Open t32.cmm located in main T32 directory for text editing
2. Add following call for FCCU utility

```
;FCCU Utility
```

```
CD.DO FCCUutility.cmm
```

FCCU utility is now prepared and it will be visible in navigation panel after every start of Trace32.

NOTE

The FCCU utility will only start automatically if your target board and debugger are powered and connected when your Trace32 is started. In other case, user has to manually start it by loading FCCU_utility.cmm via Trace32 interface.

2.2. Setting correct source file path

If there are more user customized scripts in Trace32, then following should be added into t32.cmm file.

```
PATH + ~\custom_scripts\FCCU_utility
```

3. FCCU utility

FCCU_utility is visible in navigation panel as figure below present.

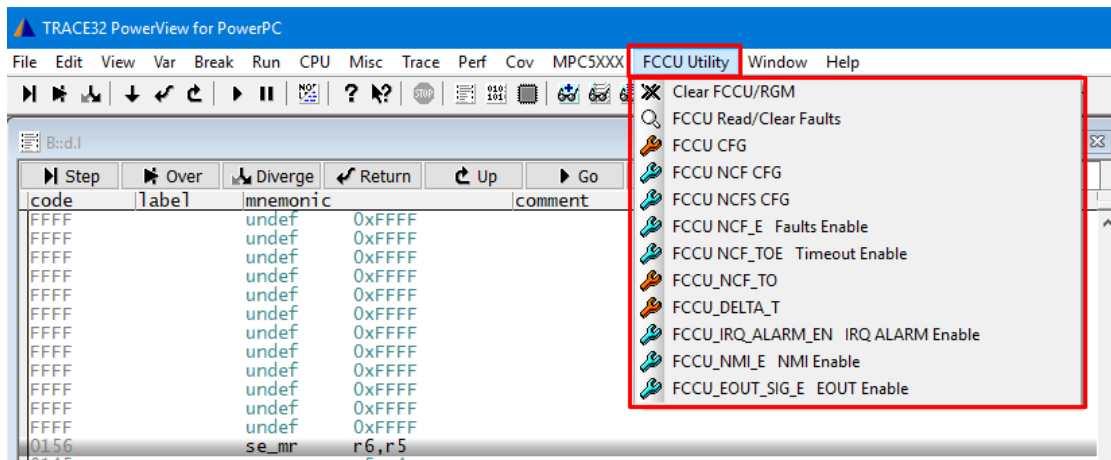


Figure 1. FCCU Utility

3.1. Clear FCCU/RGM

The first option is for clearing faults latched in FCCU and reset sources latched in RGM (reset generation module). After using this feature all faults latched in FCCU will be cleared from registers FCCU_NCFs_x and all reset sources from RGM_FES and RGM_DES registers.

NOTE

All faults latched in FCCU_NCFs_x status registers will be cleared. However, if there is active fault line from any fault source module, this particular fault will be set immediately after fault clearing. For the outside world it can look like the fault was not cleared at all, which is not true.

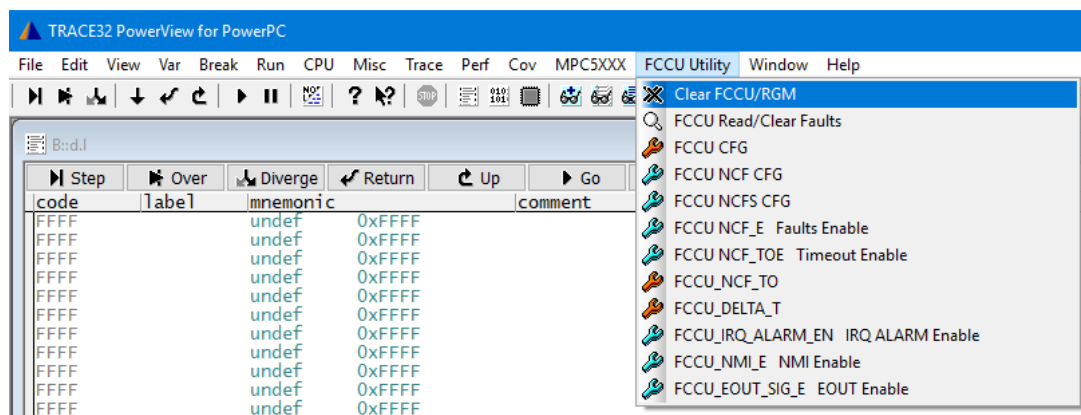


Figure 2. Clear FCCU/RGM

3.2. FCCU Read/Clear Faults

This option allow user to clear or read fault of FCCU in following way:

- Clear all fault
- Read all fault
- Clear each NCFs register separately

After calling this feature a new peripheral window will popup in Trace32 as presented on figure below.

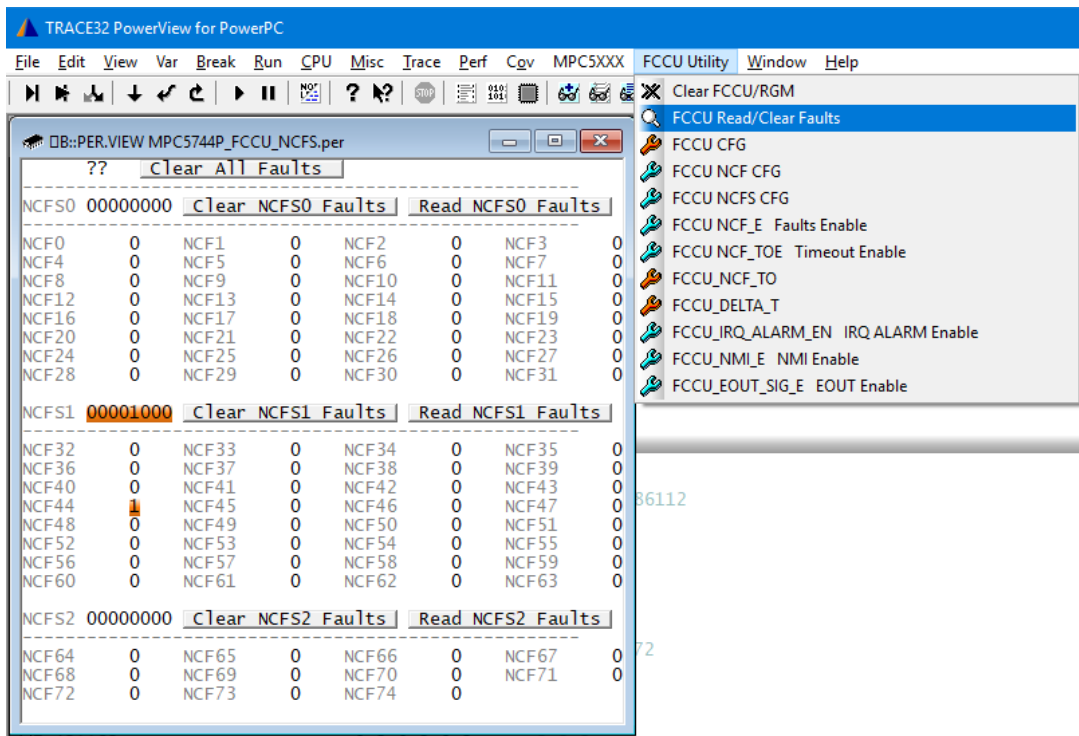


Figure 3. FCCU Read/Clear Faults

From the figure below user can see that fault NCF[44] is active. By clicking with mouse on this fault bit status user can see the description of this fault. This fault description is dynamic and will change with different microcontrollers. This will ease work with FCCU, as user doesn't need to search for fault description in device manuals. User can read fault description also in the Trace32 status bar.

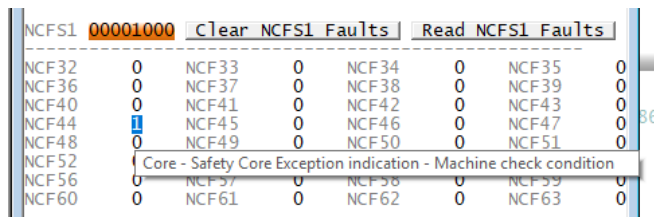


Figure 4. FCCU fault description

3.2.1. Clear all faults

By pressing clear all fault button in header of peripheral file all latched fault will be cleared.

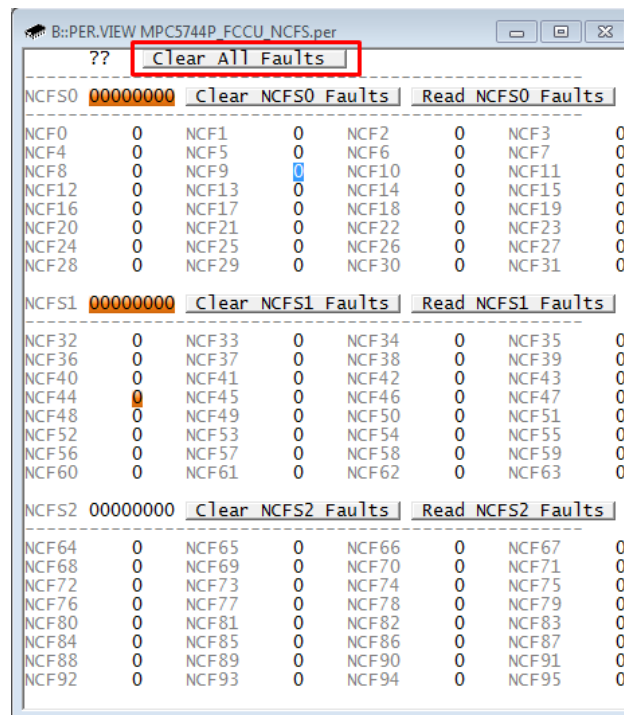


Figure 5. Clear all faults

3.2.2. Clear NCFS faults separately

There is also possibility to clear NCFS registers separately by buttons presented on picture below. The faults are not read automatically after clearing.

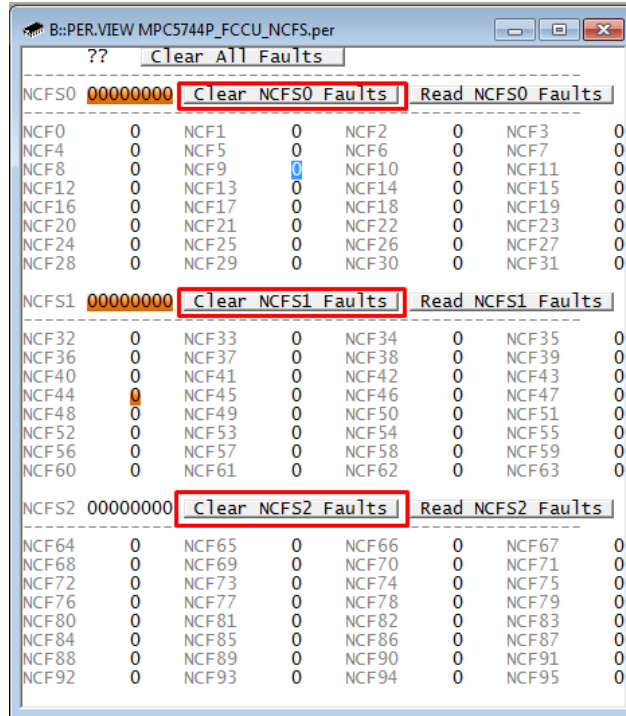


Figure 6. Clear NCF0 faults

3.2.3. Read NCF0 faults

This option allows user to effectively read all latched faults in NCF0 registers.

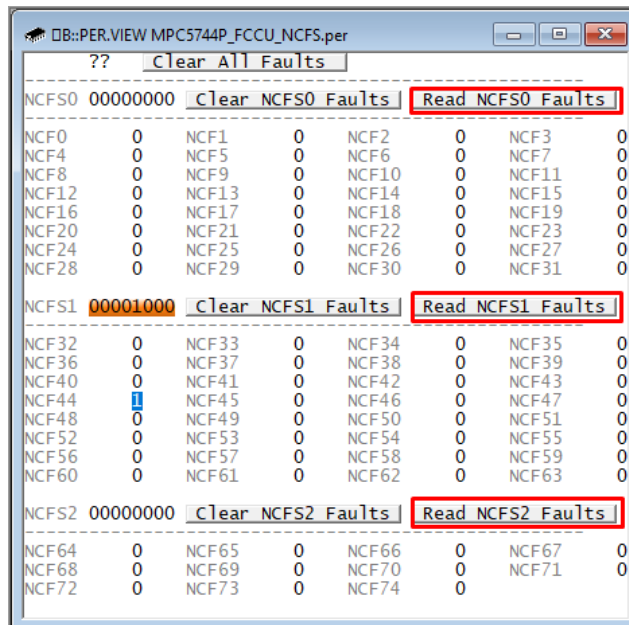


Figure 7. FCCU read faults

3.3. FCCU_CFG

User can set up FCCU configuration register (CFG) directly via s menu item FCCU_CFG. After selecting FCCU_CFG configuration peripheral window will popup. Here it is possible to set FCCU_CFG register via Set CFG option button. The value of FCCU_CFG register is entered in 32-bit format. After inserting desired value and pressing Enter, the script will automatically set FCCU_CFG value to desired one.

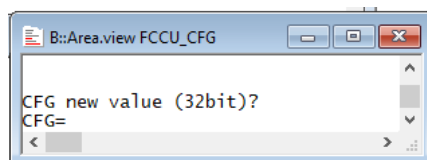
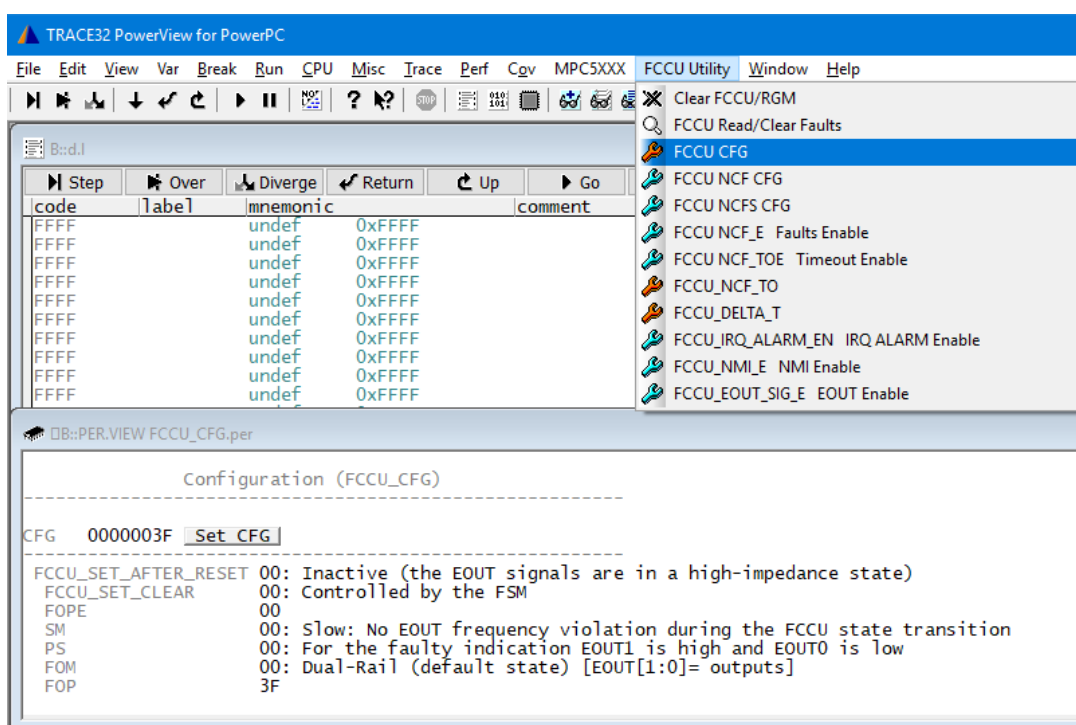


Figure 8. FCCU set CFG



3.4. FCCU_NCF_CFG

This option allows user to set up non-critical fault configuration reaction.

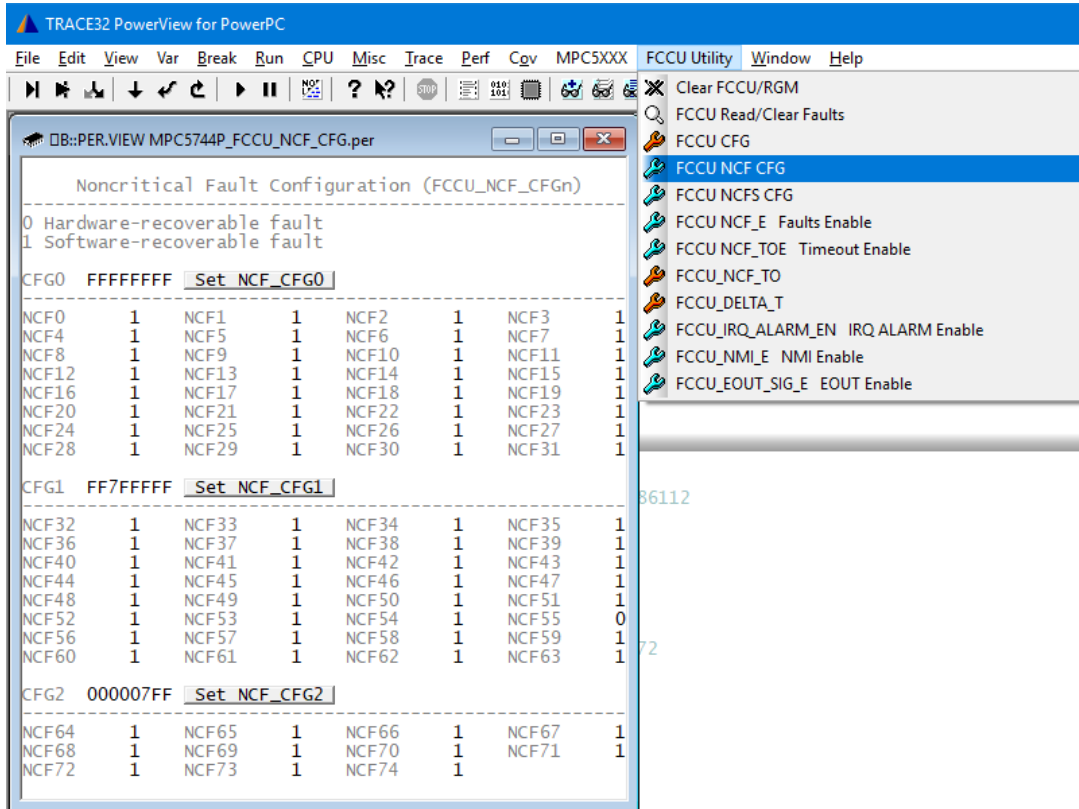


Figure 10. FCCU_NCF_CFG

3.5. FCCU_NCFS_CFG

This option offers possibility to configure FCCU noncritical fault reaction in terms of fault-recovery management.

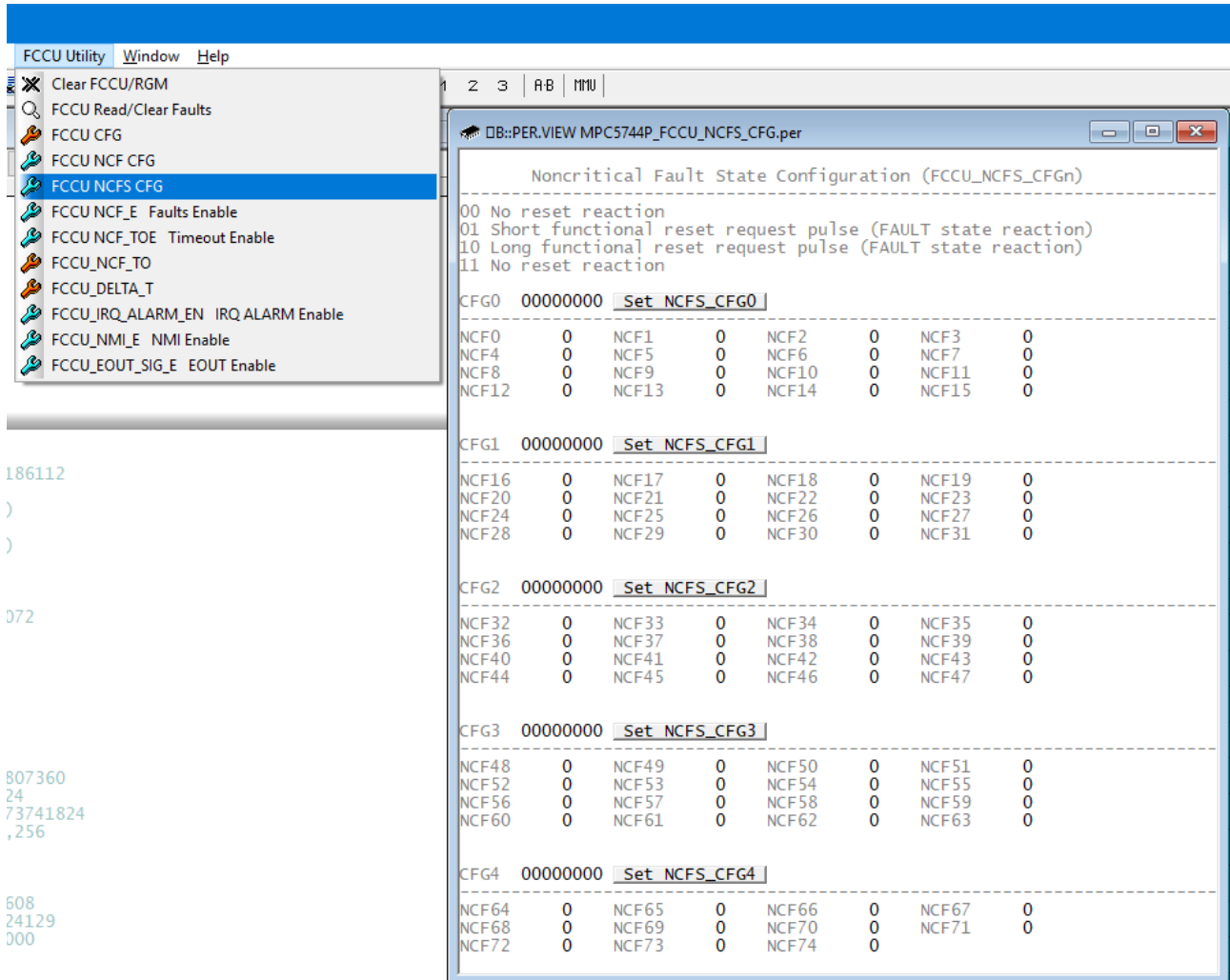


Figure 11. FCCU_NCFS_CFG

Setting of this register value is done via Set_NCFS_CFGx. After clicking on button context window will popup as shown below asking for register value:

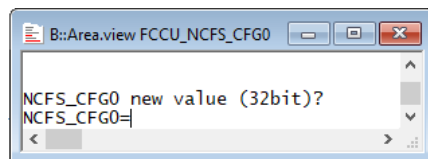


Figure 12. Set_NCFS_CFG0

3.6. Noncritical Fault Enable (FCCU_NCF_En)

The FCCU_NCF_En registers enable the fault sources to allow a transition from the NORMAL into the FAULT or ALARM state. In case of fault masking, the respective status bit into the FCCU_NCF_Sn register is set (for debugging purposes), only the reaction is masked.

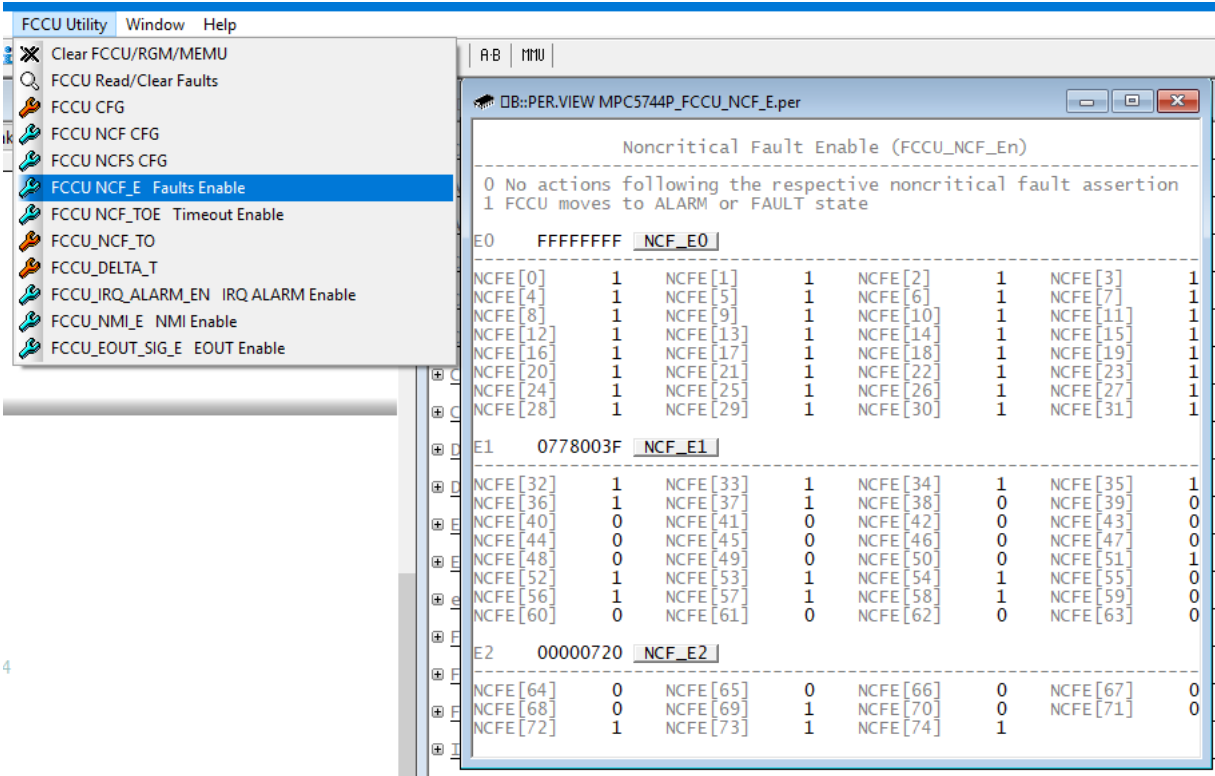


Figure 13. FCCU_NCF_En

Setting of this register value is done via NCF_Ex. After clicking on button context window will popup as shown below asking for register value:

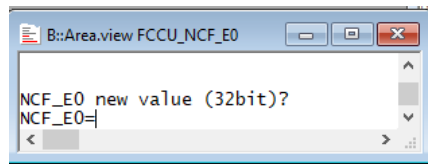


Figure 14. FCCU_NCF_E0

3.7. Noncritical Fault Timeout Enable (FCCU_NCF_TOEn)

The FCCU_NCF_TOEx registers enable a transition from the NORMAL state into the ALARM state if the respective noncritical fault is enabled (NCFEx and NCFTOEx are set). In case the respective timeout is disabled (NCFTOEx is cleared) and the noncritical fault is enabled (NCFEx is set) the FCCU moves into the FAULT state if the related noncritical fault is asserted. The timer (preset with the timeout value defined by FCCU_TO register) is started when the FCCU moves into the ALARM state. If the fault is not recovered within the timeout the FCCU moves from the ALARM state to the FAULT state.

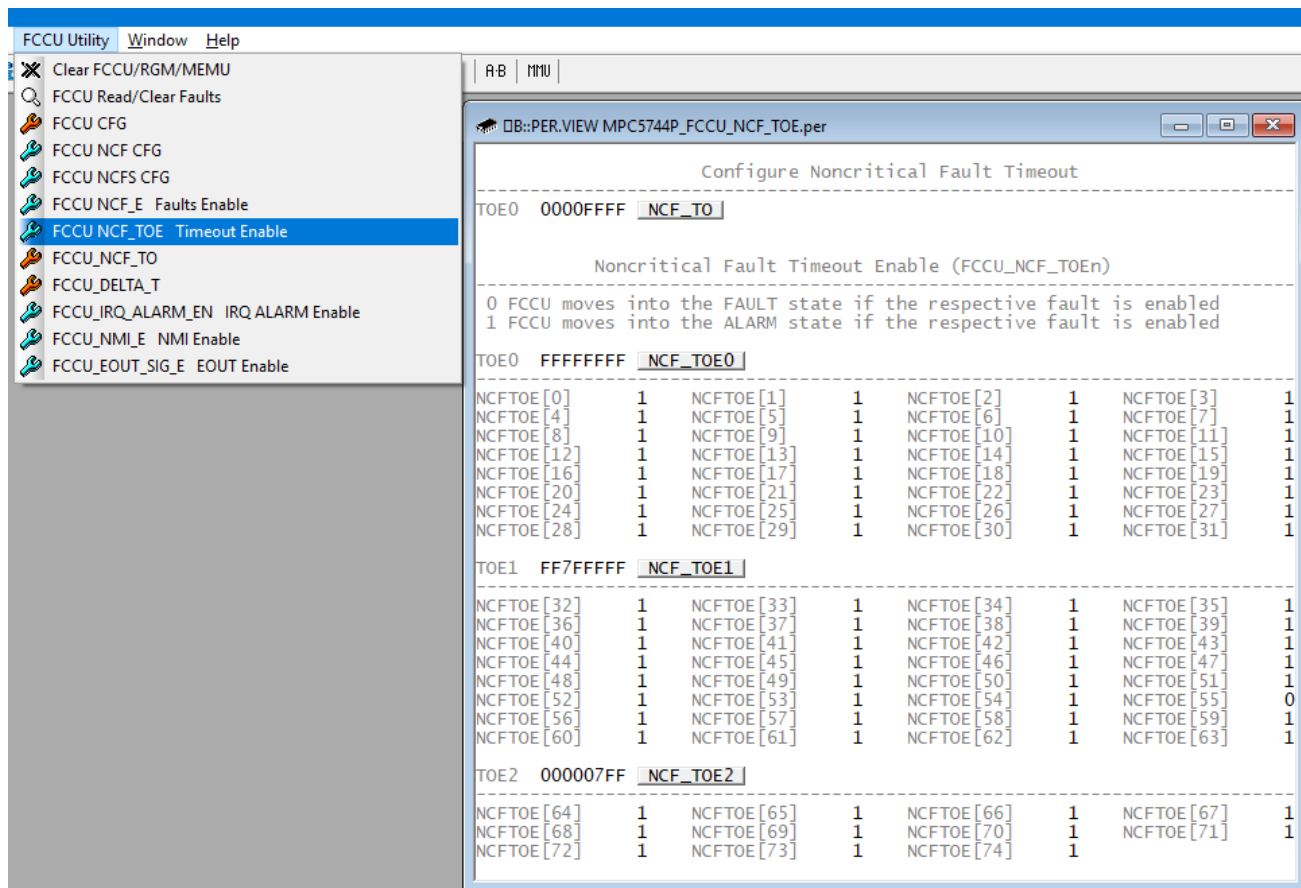


Figure 15. FCCU_NCF_TOEn

Setting of this register value is done via NCF_TO and NCF_TOEx. After clicking on button context window will popup as shown below asking for register value:

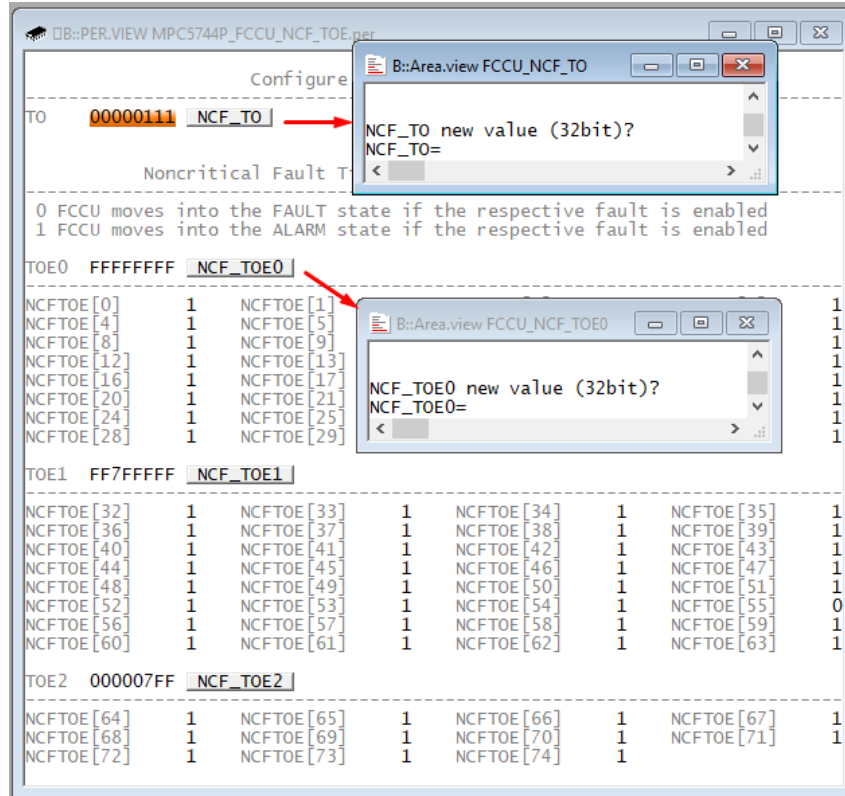


Figure 16. NCF_TO and NCF_TOEx

3.8. Noncritical Fault Timeout (FCCU_NCF_TO)

Defines the preset value of the timer for the recovery of enabled noncritical faults. Once FCCU enters Alarm state, following the assertion of a noncritical fault that is enabled (NCFEa and NCFTOEa are set), the timer starts the countdown.

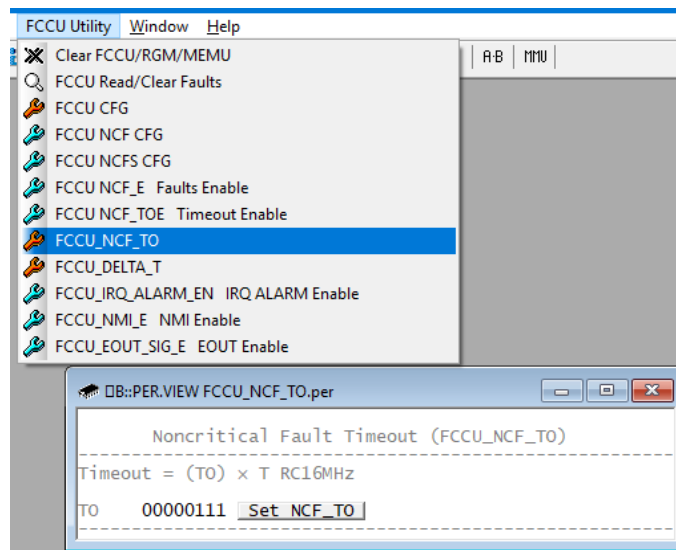


Figure 17. FCCU_NCF_TO

3.9. Delta T (FCCU_DELTA_T)

The FCCU_DELTA_T register is used for programming the value of delta_T constant, in microseconds.

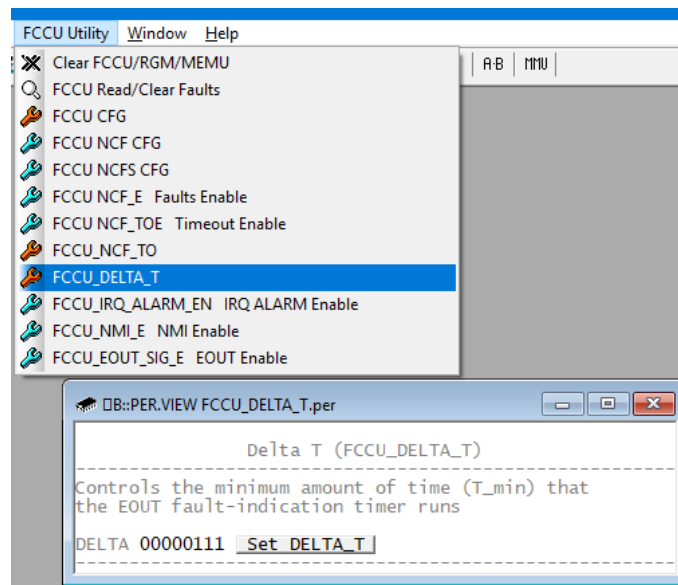


Figure 18. FCCU_DELTA_T

3.10. IRQ Alarm Enable (FCCU_IRQ_ALARM_ENn)

These registers enable the corresponding IRQ alarm.

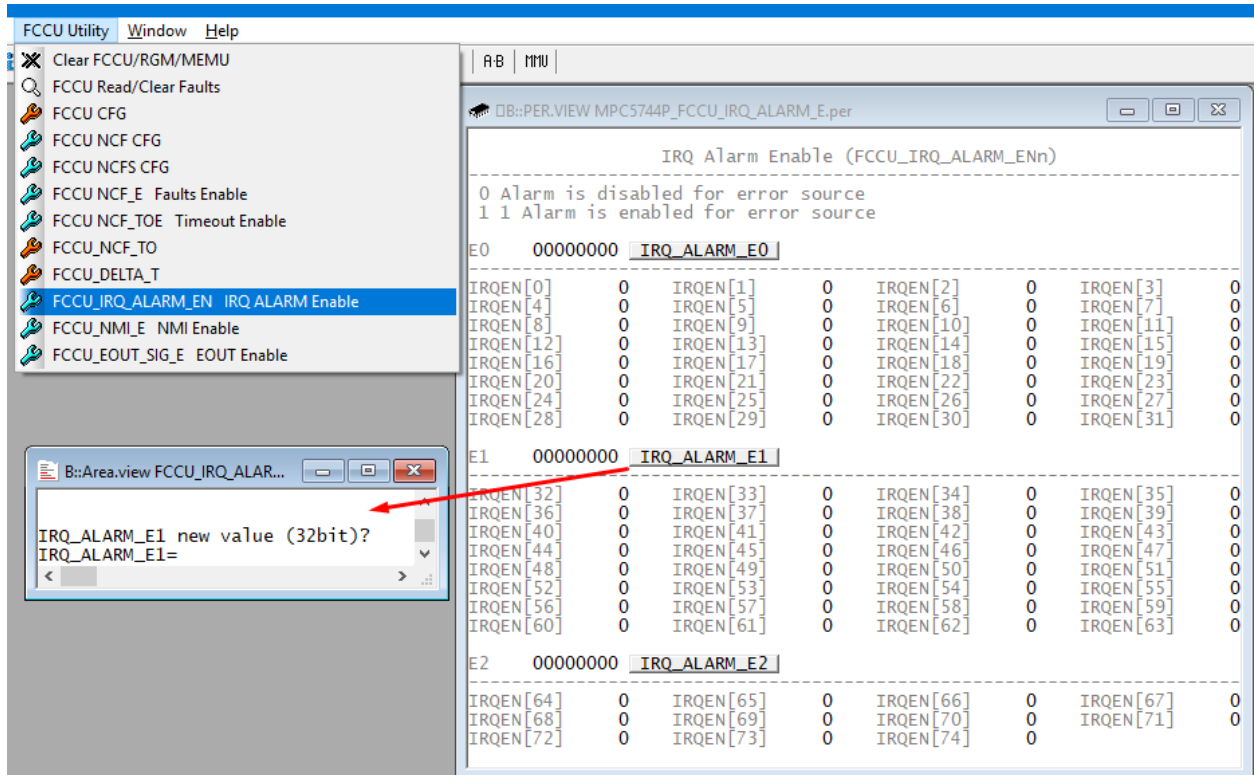


Figure 19. FCCU_IRQ_ALARM_ENn

3.11. NMI Enable (FCCU_NMI_ENn)

These registers enable the NMI reaction on FCCU fault.

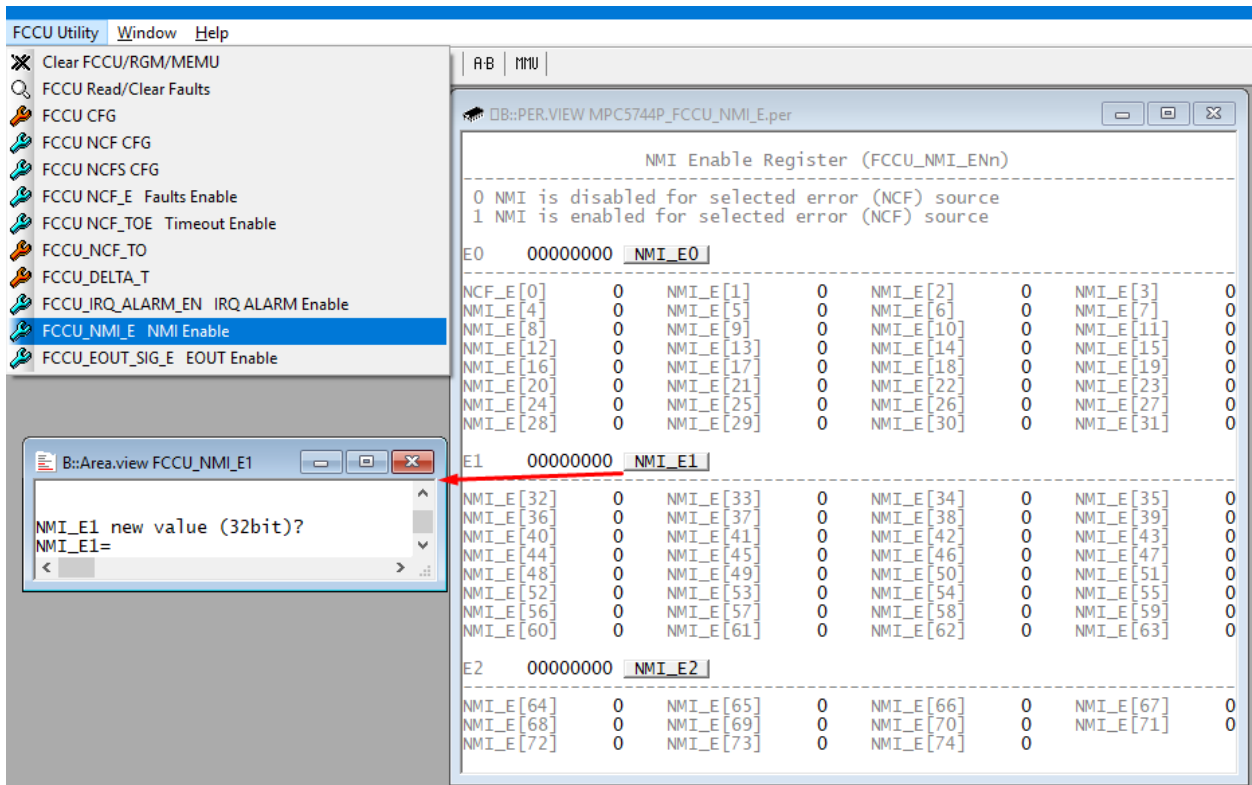


Figure 20. FCCU_NMI_ENn

3.12. Bistable Fault-Output (EOUT) Mode Signaling Enable (FCCU_EOUT_SIG_ENn)

Applies only when the EOUT signals are active (FCCU_SET_AFTER_RESET). When FCCU is configured for Bistable fault-output mode (FOM), controls whether fault-output (EOUT) signaling is enabled for the associated noncritical fault channel (x). (For other fault-output modes, fault-output signaling is always enabled, regardless of the value of this field.)

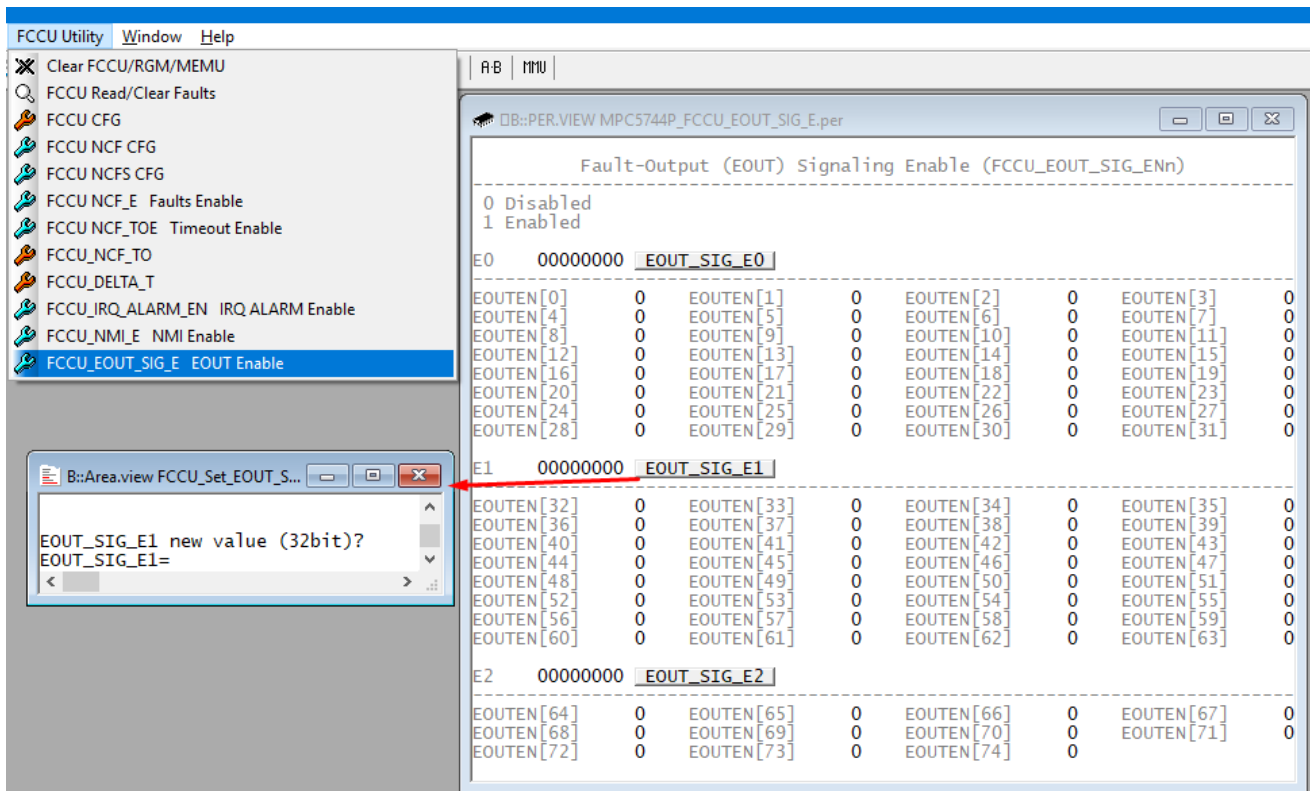


Figure 21. FCCU_EOUT_SIG_ENn



How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Registered trademarks: NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE,

JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners.

ARM, the ARM logo, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. mbed is a trademark of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

IEEE nnn, nnn, and nnn are registered trademarks of the Institute of Electrical and Electronics Engineers, Inc. (IEEE). This product is not endorsed or approved by the IEEE. Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. (Add contract language here, as necessary.)

© 2016 NXP B.V.

Document Number: MPC57xxLFCCUUG

Rev. 0
3/2018



PRELIMINARY

