User Manual

for S32K1 QDEC Driver

Document Number: UM2QDEC4.4 Rev0000R1.0.1 Rev. 1.0

Chapter 1

Revision History

Revision	Date	Author	Description
1.0	24.02.2022	NXP RTD Team	Prepared for release RTD S32K1 Version 1.0.1

Chapter 2

Introduction

This User Manual describes NXP Semiconductor QDEC driver for S32K1. The QDEC driver configuration parameters and deviations from the specification are described in QDEC Driver chapter of this document. QDEC driver requirements and APIs are vendor-specific.

2.1 Supported Derivatives

The software described in this document is intended to be used with the following microcontroller devices of NXP Semiconductors:

- s32k116_qfn32
- s32k116_lqfp48
- s32k118_lqfp48
- s32k118_lqfp64
- s32k142_lqfp48
- s32k142_lqfp64
- s32k142_lqfp100
- s32k142w_lqfp48
- s32k142w_lqfp64
- s32k144_lqfp48

- s32k144_lqfp64
- s32k144_lqfp100
- s32k144_mapbga100
- s32k144w_lqfp48
- s32k144w_lqfp64
- s32k146_lqfp64
- s32k146_lqfp100
- s32k146_mapbga100
- s32k146_lqfp144
- s32k148_lqfp100
- s32k148_mapbga100
- s32k148_lqfp144
- s32k148_lqfp176

All of the above microcontroller devices are collectively named as S32K1.

2.2 Overview

AUTOSAR (AUTomotive Open System ARchitecture) is an industry partnership working to establish standards for software interfaces and software modules for automobile electronic control systems.

AUTOSAR:

- paves the way for innovative electronic systems that further improve performance, safety and environmental friendliness.
- is a strong global partnership that creates one common standard: "Cooperate on standards, compete on implementation".
- is a key enabling technology to manage the growing electrics/electronics complexity. It aims to be prepared for the upcoming technologies and to improve cost-efficiency without making any compromise with respect to quality.
- facilitates the exchange and update of software and hardware over the service life of the vehicle.

2.3 About This Manual

This Technical Reference employs the following typographical conventions:

- **Boldface** style: Used for important terms, notes and warnings.
- *Italic* style: Used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

Notes and warnings are shown as below:

Note

This is a note.

Warning

This is a warning

2.4 Acronyms and Definitions

Term	Definition
API	Application Programming Interface
ASM	Assembler
BSMI	Basic Software Make file Interface
C/CPP	C and C++ Source Code
CS	Chip Select
CTU	Cross Trigger Unit
DMA	Direct Memory Access
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECU	Electronic Control Unit
FIFO	First In First Out
LSB	Least Significant Bit
MCU	Micro Controller Unit
MIDE	Multi Integrated Development Environment
MSB	Most Significant Bit
N/A	Not Applicable
QDEC	Quadrature Decoder
RAM	Random Access Memory
SIU	Systems Integration Unit
SWS	Software Specification
VLE	Variable Length Encoding
XML	Extensible Markup Language

2.5 Reference List

#	Title	Version
1	S32K1XX Reference Manual	S32K1xx Series Reference Manual, Rev. 14, 09/2021
2	Datasheet	S32K1xx Data Sheet, Rev. 14, 08/2021
3	Errata	S32K116_0N96V Rev. 22/OCT/2021
		S32K118_0N97V Rev. 22/OCT/2021
		S32K142_0N33V Rev. 22/OCT/2021
		S32K144_0N57U Rev. 22/OCT/2021
		S32K144W_0P64A Rev. 22/OCT/2021
		S32K146_0N73V Rev. 22/OCT/2021
		S32K148_0N20V Rev. 22/OCT/2021

Chapter 3

Driver

3.1 Requirements

Requirements for this driver are detailed in the Autosar Driver Software Specification document (See Table Reference List).

For CDD: QDEC Driver is a Complex Device Driver (CDD), so there are no AUTOSAR requirements regarding this module.

It has vendor-specific requirements and implementation.

3.2 Driver Design Summary

Quadrature decoder with input filters, relative position counting, and interrupt on position count or capture of position count on external event

3.3 Hardware Resources

#	Hardware IP	Description
1	FTM	FlexTimer

3.4 Deviations from Requirements

None.

3.5 Driver Limitations

The QDEC driver software have some following limitations for RTD S32K1 :

- Variant aware support not validated.
- Overflow Interrupt is supported but not validated.
- Limited validation of S32 Design Studio components for High Level Driver (HLD) and Low Level Drivers (IPL).

3.6 Driver usage and configuration tips

In this chapter, the extra features from our drivers that are not described in the AutoSAR standard are detailed.

3.7 Runtime errors

No runtime errors used.

3.8 Symbolic Names Disclaimer

All containers having symbolicNameValue set to TRUE in the AUTOSAR schema will generate defines like:

#define <Mip>Conf_<Container_ShortName>_<Container_ID>

For this reason it is forbidden to duplicate the names of such containers across the RTD configurations or to use names that may trigger other compile issues (e.g. match existing `#ifdefs` arguments).

Chapter 4

Tresos Configuration Plug-in

This chapter describes the Tresos configuration plug-in for the driver. All the parameters are described below.

- Module Qdec
  - Container QdecConfigurationOfOptionalApis
    - Parameter QdecDeInitApi
  - Container QdecInstanceConfig
    - Parameter QdecHardwareModuleId
    - Parameter QdecFtmModule
    - Parameter QdecFtmClockSource
    - Parameter QdecFtmPrescaler
    - Parameter QdecFtmPrescale_Alternate
    - Parameter QdecEncodingMode
    - Parameter QdecOverflowNotification
    - Parameter QdecMinCounterValue
    - Parameter QdecMaxCounterValue
    - Parameter QdecmEnAPhaseFilter
    - Parameter QdecmAPhaseFilterVal
    - Parameter QdecmAPhasePolarity
    - Parameter QdecmEnBPhaseFilter
    - Parameter QdecmBPhaseFilterVal
    - Parameter QdecmBPhasePolarity
    - Parameter QdecEnTofIsr
    - Reference QdecInstanceEcucPartitionRef
  - Container QdecGeneral
    - Parameter QdecMulticoreEnabled
    - Parameter QdecDevErrorDetect
    - Parameter QdecEnableUserModeSupport
    - Parameter QdecEnableDualClockMode
    - Parameter QdecOverFlowNotificationSupported
    - Reference QdecEcucPartitionRef
  - Container CommonPublishedInformation

* Parameter ArReleaseMajorVersion
* Parameter ArReleaseMinorVersion
* Parameter ArReleaseRevisionVersion
* Parameter ModuleId
* Parameter SwMajorVersion
* Parameter SwMinorVersion
* Parameter SwPatchVersion
* Parameter VendorApiInfix
* Parameter VendorId

4.1 Module Qdec

Configuration of QDEC (Quadrature Decoder) module.

Included containers:

- QdecConfigurationOfOptionalApis
- QdecInstanceConfig
- QdecGeneral
- CommonPublishedInformation

Property	Value
type	ECUC-MODULE-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantSupport	true
supportedConfigVariants	VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

4.2 Container QdecConfigurationOfOptionalApis

Configuration of optional APIs.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.3 Parameter QdecDeInitApi

Adds / removes the service Qdec_DeInit() from the code.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.4 Container QdecInstanceConfig

This container contains Qdec-specific parameters for selecting optional settings.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	2
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.5 Parameter QdecHardwareModuleId

ID of the QDEC hardware module.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	2
min	0

4.6 Parameter QdecFtmModule

Selects one Ftm modules available on the platform.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	FTM_1
literals	['FTM_1', 'FTM_2']

4.7 Parameter QdecFtmClockSource

Select origin of clock source used by current instance of FTM.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	SYSTEM_CLOCK
literals	['SYSTEM_CLOCK', 'EXTERNAL_CLOCK', 'FIXED_FREQUENCY_CLOCK']

4.8 Parameter QdecFtmPrescaler

Optional QDEC driver specific clock prescale factor, if supported by hardware. Implementation is defined vendor specific.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	DIV1
literals	['DIV1', 'DIV2', 'DIV4', 'DIV8', 'DIV16', 'DIV32', 'DIV64', 'DIV128']

4.9 Parameter QdecFtmPrescale_Alternate

Select input count source (clock) used for this Ftm module. This parameter will be used by the Qdec_SetClockMode function.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	False
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	DIV1
literals	['DIV1', 'DIV2', 'DIV4', 'DIV8', 'DIV16', 'DIV32', 'DIV64', 'DIV128']

4.10 Parameter QdecEncodingMode

Encoding mode used by FTM in the QDEC mode.

-> MODE_PHASE_ENCODE < Phase A and phase B encoding mode >

-> MODE_COUNT_AND_DIR_ENCODE < Count and direction encoding mode >

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	MODE_COUNT_AND_DIR_ENCODE
literals	['MODE_PHASE_ENCODE', 'MODE_COUNT_AND_DIR_ENCODE']

4.11 Parameter QdecOverflowNotification

User callback notification function.

This option is only activated when QdecGeneral/QdecNotificationSupported is checked.

<note>

Use NULL_PTR without any quotes to determine no notification function is used.

If the string is different from above, it will be used as the notification function name.

</note>

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	NULL_PTR

4.12 Parameter QdecMinCounterValue

Minimum counter value.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	65534
min	0

4.13 Parameter QdecMaxCounterValue

Maximum counter value.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	65535
max	65535
min	1

4.14 Parameter QdecmEnAPhaseFilter

Enables the filter for the quadrature decoder phase A,B inputs

False: disable phase filter

True : enable phase filter

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.15 Parameter QdecmAPhaseFilterVal

Filter value (if input filter is enabled)

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	15
min	0

4.16 Parameter QdecmAPhasePolarity

PHASE_NORMAL Phase input signal is not inverted before identifying the rising and falling edges of this signal

PHASE_INVERT Phase input signal is inverted before identifying the rising and falling edges of this signal <]/p>

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	PHASE_NORMAL
literals	['PHASE_NORMAL', 'PHASE_INVERT']

4.17 Parameter QdecmEnBPhaseFilter

Enables the filter for the quadrature decoder phase A,B inputs

False: disable phase filter

True : enable phase filter

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.18 Parameter QdecmBPhaseFilterVal

Filter value (if input filter is enabled)

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	15
min	0

4.19 Parameter QdecmBPhasePolarity

PHASE_NORMAL Phase input signal is not inverted before identifying the rising and falling edges of this signal

PHASE_INVERT Phase input signal is inverted before identifying the rising and falling edges of this signal <]/p>

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	PHASE_NORMAL
literals	['PHASE_NORMAL', 'PHASE_INVERT']

4.20 Parameter QdecEnTofIsr

True: Timer Overflow Interrupt is enabled.

False: Timer Overflow Interrupt is disabled.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.21 Reference QdecInstanceEcucPartitionRef

Maps one or multiple QDEC module to an ECUC partition.

The ECUC partitions referenced are a subset of the ECUC partitions where the QDEC driver is mapped to.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0

Property	Value
upperMultiplicity	Infinite
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.22 Container QdecGeneral

This container contains the module-wide configuration parameters of the QDEC Driver.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.23 Parameter QdecMulticoreEnabled

Switch to enable/disable multi-core feature.

User can choose ENABLE multi-core feature by checking this option, this will force to configure at least 1 ECUC partition in

QdecInstanceEcucPartitionRef, and each QDEC channel in QdecInstance to configure at least 1 ECUC partition reference

in QdecInstanceEcucPartitionRef container to fulfill generating code condition; OR unchecked this option to DISABLE

multi-core feature, performing this action will force user to remove all ECUC partition reference in every Qdec instance contained

in QdecInstace and in QdecInstanceEcucPartitionRef.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.24 Parameter QdecDevErrorDetect

Enables/Disables development error detection.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.25 Parameter QdecEnableUserModeSupport

Enable or disable user mode support for quadrature decoder.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.26 Parameter QdecEnableDualClockMode

Enables prescaler settings at mode transition.

true: Enabled.

false: Disabled.

Note: This feature is not required by Autosar.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.27 Parameter QdecOverFlowNotificationSupported

Switch to indicate that the notifications are supported.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.28 Reference QdecEcucPartitionRef

Maps the QDEC driver to zero or multiple ECUC partitions to make the driver API available in the according partition.

Depending on the addressed timer resource the interfaces operate as follows.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.29 Container CommonPublishedInformation

Common container, aggregated by all modules. It contains published information about vendor and versions.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.30 Parameter ArReleaseMajorVersion

Major version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.31 Parameter ArReleaseMinorVersion

Minor version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.32 Parameter ArReleaseRevisionVersion

Revision version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.33 Parameter ModuleId

Module ID of this module from Module List.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	255
max	255
min	255

4.34 Parameter SwMajorVersion

Major version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	1
max	1
min	1

4.35 Parameter SwMinorVersion

Minor version number of the vendor specific implementation of the module. The numbering is vendor pecific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.36 Parameter SwPatchVersion

Patch level version number of the vendor specific implementation of the module. The numbering is endor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	1
max	1
min	1

4.37 Parameter VendorApiInfix

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that he name of APIs is extended by the VendorId and a vendor specific name.

This parameter is used to specify the vendor specific name. In total, the implementation specific name is generated as follows:

<ModuleName>_>VendorId>_<VendorApiInfix>.

E.g. assuming that the VendorId of the implementor is 123 and the implementer chose a VendorApiInfix of "v11r456" a api name Can_Write defined in the SWS will translate to Can_123_v11r456Write.

This parameter is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	

4.38 Parameter VendorId

Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	43
max	43
min	43

Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

Chapter 6

Module Documentation

6.1 Ftm_Qdec_Ip Driver

6.1.1 Detailed Description

Data Structures

- struct Ftm_Qdec_Ip_CallbackCfgType
  Structure for notification. More...
- struct Ftm_Qdec_Ip_PhaseParamsType
  Quadrature decoder channel parameters. More...
- struct Ftm_Qdec_Ip_ConfigType
  Quadrature configure structure. More...
- struct Ftm_Qdec_Ip_StateType
  FTM Quadrature state (counter value and flags) More...

Types Reference

- typedef FTM_Type Ftm_Qdec_HwAddrType
  FTM - Register Layout Typedef.
- typedef void(* Ftm_Qdec_Ip_CallbackType) (uint8 callbackParam1)
  Callback type for each instance.

Enum Reference

- enum Ftm_Qdec_Ip_ClockSourceType
  FlexTimer Clock Source Selection.
- enum Ftm_Qdec_Ip_ClockPrescType
  FlexTimer Prescale Factor Selections bits (PS).
- enum Ftm_Qdec_Ip_EncodingModeType
  Encoding mode used by FTM in the QDEC mode.
- enum Ftm_Qdec_Ip_PhasePolarityType
  Quadrature phase polarities, normal or inverted polarity.
- enum Ftm_ModeType
  Qdec mode type include: QDEC_MODE_NORMAL and QDEC_MODE_SLEEP.

Function Reference

- void Ftm_Qdec_Ip_Init (uint8 module, const Ftm_Qdec_Ip_ConfigType *config)

  Configures the quadrature mode and starts measurement.
- void Ftm_Qdec_Ip_Deinit (uint8 module)

  De-activates the Quadrature Decoder Mode.
- Ftm_Qdec_Ip_StateType Ftm_Qdec_Ip_GetState (uint8 module)

  Return the current quadrature decoder state (counter value, timer overflow, TOF direction and counting direction flags)
- void Ftm_Qdec_Ip_CounterState (const uint8 module)

  Set state of counter.
- void Ftm_Qdec_Ip_ResetState (const uint8 module)

  Reset state of module to default value.
- void Ftm_Qdec_Ip_SetMode (const uint8 module, Ftm_ModeType mode)

  Set mode of Qdec module.

6.1.2 Data Structure Documentation

6.1.2.1 struct Ftm_Qdec_Ip_CallbackCfgType

Structure for notification.

The structure used to notification

Definition at line 205 of file Ftm_Qdec_Ip_Types.h.

6.1.2.2 struct Ftm_Qdec_Ip_PhaseParamsType

Quadrature decoder channel parameters.

Definition at line 214 of file Ftm_Qdec_Ip_Types.h.

Data Fields

Type	Name	Description
boolean	mEnPhaseFilter	Enables the filter for the quadrature decoder phase A,B inputs False: disable phase filter, True : enable phase filter
uint8	mPhaseFilterVal	Filter value (if input filter is enabled)
Ftm_Qdec_Ip_PhasePolarityType	mPhasePolarity	Phase polarity

6.1.2.3 struct Ftm_Qdec_Ip_ConfigType

Quadrature configure structure.

Definition at line 226 of file Ftm_Qdec_Ip_Types.h.

Data Fields

Type	Name	Description
Ftm_Qdec_Ip_EncodingModeType	mEncMode	FTM_MODE_PHASE_ENCODE or FTM_MODE_COUNT_AND_DIR_ENCODE
uint16	mMinCntVal	Minimum counter value
uint16	mModCntVal	Maximum counter value
boolean	mEnTofIsr	True: Timer Overflow Interrupt is enabled, False: Timer Overflow Interrupt is disabled
Ftm_Qdec_Ip_PhaseParamsType	mPhaseAConfig	Configuration of input phase A
Ftm_Qdec_Ip_PhaseParamsType	mPhaseBConfig	Configuration of input phase B
Ftm_Qdec_Ip_CallbackCfgType	overflowCb	overflow callback
Ftm_Qdec_Ip_ClockSourceType	eSourceClock	FlexTimer Clock Source

6.1.2.4 struct Ftm_Qdec_Ip_StateType

FTM Quadrature state (counter value and flags)

Definition at line 253 of file Ftm_Qdec_Ip_Types.h.

Data Fields

Type	Name	Description
uint16	mCounter	FTM Counter value
boolean	mTimerOverflow	True if timer overflow occurred, False otherwise
boolean	mTimerOverflowDir	Timer Overflow Direction: False if there was an FTM counter decrement and FTM counter changes from its minimum value to its maximum value, True if there was an FTM counter increment and FTM counter changes from its maximum value to its minimum value
boolean	mCountDir	FTM Counter Direction in QDEC mode: False if counting direction is decreasing (counter decrement), True if counting direction is increasing (counter increment)

6.1.3 Types Reference

6.1.3.1 Ftm_Qdec_HwAddrType

```
typedef FTM_Type Ftm_Qdec_HwAddrType
```

FTM - Register Layout Typedef.

Definition at line 196 of file Ftm_Qdec_Ip_Types.h.

6.1.3.2 Ftm_Qdec_Ip_CallbackType

```
typedef void(* Ftm_Qdec_Ip_CallbackType) (uint8 callbackParam1)
```

Callback type for each instance.

Definition at line 199 of file Ftm_Qdec_Ip_Types.h.

6.1.4 Enum Reference

6.1.4.1 Ftm_Qdec_Ip_ClockSourceType

```
enum Ftm_Qdec_Ip_ClockSourceType
```

FlexTimer Clock Source Selection.

Enumerator

FTM_CLOCK_SOURCE_NONE	None use clock for FTM
FTM_CLOCK_SOURCE_SYSTEMCLK	System clock
FTM_CLOCK_SOURCE_FIXEDCLK	Fixed clock
FTM_CLOCK_SOURCE_EXTERNALCLK	External clock

Definition at line 124 of file Ftm_Qdec_Ip_Types.h.

6.1.4.2 Ftm_Qdec_Ip_ClockPrescType

```
enum Ftm_Qdec_Ip_ClockPrescType
```

FlexTimer Prescale Factor Selections bits (PS).

Selects one of 8 division factors for the clock source selected by CLKS. The new prescaler factor affects the clock source on the next FTM input clock cycle after the new value is updated into the register bits. This field is write protected. It can be written only when MODE[WPDIS] = 1. 000b - Divide by 1 001b - Divide by 2 010b - Divide by 4 011b - Divide by 8 100b - Divide by 16 101b - Divide by 32 110b - Divide by 64 111b - Divide by 128

Enumerator

FTM_CLOCK_DIVID_BY_1	Divide by 1
FTM_CLOCK_DIVID_BY_2	Divide by 2

Enumerator

FTM_CLOCK_DIVID_BY_4	Divide by 4
FTM_CLOCK_DIVID_BY_8	Divide by 8
FTM_CLOCK_DIVID_BY_16	Divide by 16
FTM_CLOCK_DIVID_BY_32	Divide by 32
FTM_CLOCK_DIVID_BY_64	Divide by 64
FTM_CLOCK_DIVID_BY_128	Divide by 128

Definition at line 147 of file Ftm_Qdec_Ip_Types.h.

6.1.4.3 Ftm_Qdec_Ip_EncodingModeType

enum Ftm_Qdec_Ip_EncodingModeType

Encoding mode used by FTM in the QDEC mode.

Enumerator

FTM_MODE_PHASE_ENCODE	Phase A and phase B encoding mode
FTM_MODE_COUNT_AND_DIR_ENCODE	Count and direction encoding mode

Definition at line 162 of file Ftm_Qdec_Ip_Types.h.

6.1.4.4 Ftm_Qdec_Ip_PhasePolarityType

enum Ftm_Qdec_Ip_PhasePolarityType

Quadrature phase polarities, normal or inverted polarity.

Enumerator

FTM_PHASE_NORMAL	Phase input signal is not inverted before identifying the rising and falling edges of this signal
FTM_PHASE_INVERT	Phase input signal is inverted before identifying the rising and falling edges of this signal

Definition at line 171 of file Ftm_Qdec_Ip_Types.h.

6.1.4.5 Ftm_ModeType

```
enum Ftm_ModeType
```

Qdec mode type include: QDEC_MODE_NORMAL and QDEC_MODE_SLEEP.

Enumerator

FTM_QDEC_MODE_NORMAL	Normal operation, all used interrupts are enabled according to the notification requests.
FTM_QDEC_MODE_SLEEP	Reduced power operation.

Definition at line 182 of file Ftm_Qdec_Ip_Types.h.

6.1.5 Function Reference

6.1.5.1 Ftm_Qdec_Ip_Init()

```
void Ftm_Qdec_Ip_Init (
            uint8 module,
            const Ftm_Qdec_Ip_ConfigType * config )
```

Configures the quadrature mode and starts measurement.

Parameters

in	*module*	FTM hardware instance number.
in	*config*	Configuration structure(Quadrature Decoder Mode, polarity for both phases, minimum and maximum value for the counter, filter configuration).

Returns

No return.

6.1.5.2 Ftm_Qdec_Ip_Deinit()

```
void Ftm_Qdec_Ip_Deinit (
            uint8 module )
```

De-activates the Quadrature Decoder Mode.

Parameters

in	*module*	FTM hardware instance number.

Returns

No return.

6.1.5.3 Ftm_Qdec_Ip_GetState()

```
Ftm_Qdec_Ip_StateType Ftm_Qdec_Ip_GetState (
            uint8 module )
```

Return the current quadrature decoder state (counter value, timer overflow, TOF direction and counting direction flags)

Parameters

in	*module*	FTM hardware instance number.

Returns

Structure with the state of FTM module

6.1.5.4 Ftm_Qdec_Ip_CounterState()

```
void Ftm_Qdec_Ip_CounterState (
            const uint8 module )
```

Set state of counter.

Parameters

in	*module*	FTM hardware number.

Returns

void

6.1.5.5 Ftm_Qdec_Ip_ResetState()

```
void Ftm_Qdec_Ip_ResetState (
             const uint8 module )
```

Reset state of module to default value.

Parameters

in	*module*	FTM hardware number.

Returns

void

6.1.5.6 Ftm_Qdec_Ip_SetMode()

```
void Ftm_Qdec_Ip_SetMode (
             const uint8 module,
             Ftm_ModeType mode )
```

Set mode of Qdec module.

Parameters

in	*module*	FTM hardware number.
in	*mode*	Given mode: QDEC_NORMAL_MODE or QDEC_SLEEP_MODE.

Returns

void

6.2 Qdec Driver

6.2.1 Detailed Description

Data Structures

- struct Qdec_StateType
  Quadrature decode state (counter value and flags) *More...*
- struct Qdec_PhaseParamsType
  Quadrature decoder channel parameters. *More...*
- struct Qdec_InstanceConfigType
  Quadrature instance configuration. *More...*
- struct Qdec_ConfigType
  Qdec high level configuration structure. *More...*

Enum Reference

- enum Qdec_EncodingModeType
  Encoding mode used in the Quadrature Decoder Mode.
- enum Qdec_PhasePolarityType
  Quadrature phase polarities, normal or inverted polarity.
- enum Qdec_CounterStateType
  Qdec counter state type.
- enum Qdec_ModeType
  Qdec mode type include: QDEC_MODE_NORMAL and QDEC_MODE_SLEEP.

Function Reference

- void Qdec_Init (const Qdec_ConfigType *Config)
  This function it is used to intitialize a QDEC instance.
- void Qdec_DeInit (void)
  This function it is used to disable QDEC Mode.
- Qdec_StateType Qdec_GetState (const uint8 module)
  Return the current quadrature decoder state (counter value and flags).
- void Qdec_CounterState (const uint8 module, Qdec_CounterStateType counterState)
  This function will set the state of counter.
- void Qdec_ResetState (const uint8 module)
  Reset state of module to default value.
- void Qdec_SetMode (const uint8 module, Qdec_ModeType mode)
  Set mode.

6.2.1.1 MISRA-C:2012 violations

6.2.2 Data Structure Documentation

6.2.2.1 struct Qdec_StateType

Quadrature decode state (counter value and flags)

Definition at line 183 of file Qdec_Types.h.

Data Fields

Type	Name	Description
uint32	mCounter	Counter value
boolean	mTimerOverflow	True if timer overflow occurred, False otherwise
boolean	mTimerOverflowDir	Timer Overflow Direction: False if there was a counter decrement and counter changes from its minimum value to its maximum value, True if there was a counter increment and counter changes from its maximum value to its minimum value
boolean	mCountDir	Counter Direction in QDEC mode: False if counting direction is decreasing (counter decrement), True if counting direction is increasing (counter increment)
boolean	mOverRun	True if overrun has occurred, False otherwise
boolean	mMatchEv	True if a match event in the comparators occurred, False otherwise

6.2.2.2 struct Qdec_PhaseParamsType

Quadrature decoder channel parameters.

Definition at line 200 of file Qdec_Types.h.

Data Fields

Type	Name	Description
boolean	mEnPhaseFilter	Enables the filter for the quadrature decoder phase A,B inputs False: disable phase filter, True : enable phase filter
uint8	mPhaseFilterVal	Filter value (if input filter is enabled)
Qdec_PhasePolarityType	mPhasePolarity	Phase polarity

6.2.2.3 struct Qdec_InstanceConfigType

Quadrature instance configuration.

Definition at line 220 of file Qdec_Types.h.

Data Fields

Type	Name	Description
Qdec_EncodingModeType	mEncMode	MODE_PHASE_ENCODE or MODE_COUNT_AND_DIR_ENCODE
uint16	mMinCntVal	Minimum counter value
uint16	mMaxCntVal	Maximum counter value
boolean	mEnTofIsr	True: Timer Overflow Interrupt is enabled, False: Timer Overflow Interrupt is disabled
Qdec_PhaseParamsType	mPhaseAConfig	Configuration of input phase A
Qdec_PhaseParamsType	mPhaseBConfig	Configuration of input phase B

6.2.2.4 struct Qdec_ConfigType

Qdec high level configuration structure.

Definition at line 237 of file Qdec_Types.h.

Data Fields

Type	Name	Description
uint8	CoreIndex	Core Index
uint8	Instances	Number of QDEC instances used
const Qdec_InstanceConfigType(∗	InstanceConfigType)[]	Pointer to user configuration(HLD)
const Qdec_Ipw_ConfigType ∗	Ipw_ConfigType	Pointer to wrapper layer

6.2.3 Enum Reference

6.2.3.1 Qdec_EncodingModeType

enum Qdec_EncodingModeType

Encoding mode used in the Quadrature Decoder Mode.

Enumerator

MODE_NOT_INITIALIZED	driver is not initialized
MODE_PHASE_ENCODE	Phase A and phase B encoding mode
MODE_COUNT_AND_DIR_ENCODE	Count and direction encoding mode

Definition at line 124 of file Qdec_Types.h.

6.2.3.2 Qdec_PhasePolarityType

enum Qdec_PhasePolarityType

Quadrature phase polarities, normal or inverted polarity.

Enumerator

PHASE_NORMAL	Phase input signal is not inverted before identifying the rising and falling edges of this signal
PHASE_INVERT	Phase input signal is inverted before identifying the rising and falling edges of this signal

Definition at line 134 of file Qdec_Types.h.

6.2.3.3 Qdec_CounterStateType

```
enum Qdec_CounterStateType
```

Qdec counter state type.

Enumerator

QDEC_RESUME_COUNTER	When the counting starts again to count
QDEC_PAUSE_COUNTER	When counting is paused

Definition at line 159 of file Qdec_Types.h.

6.2.3.4 Qdec_ModeType

```
enum Qdec_ModeType
```

Qdec mode type include: QDEC_MODE_NORMAL and QDEC_MODE_SLEEP.

Enumerator

QDEC_MODE_NORMAL	Normal operation, all used interrupts are enabled according to the notification requests.
QDEC_MODE_SLEEP	Reduced power operation.

Definition at line 170 of file Qdec_Types.h.

6.2.4 Function Reference

6.2.4.1 Qdec_Init()

```
void Qdec_Init (
            const Qdec_ConfigType * Config )
```

This function it is used to intitialize a QDEC instance.

This function activates your Quadrature Decoder Mode on the selected module.

Parameters

in	*Config*	- Qdec configuration structure.

Returns

void

Precondition

None.

6.2.4.2 Qdec_DeInit()

```
void Qdec_DeInit (
            void  )
```

This function it is used to disable QDEC Mode.

This function disable your Quadrature Decoder Mode on the selected module.

Returns

void

Precondition

None.

6.2.4.3 Qdec_GetState()

```
Qdec_StateType Qdec_GetState (
            const uint8 module )
```

Return the current quadrature decoder state (counter value and flags).

This function get the current quadrature decoder state (counter value and flags).

Parameters

in	*module*	- Name of your QDEC instance.

Returns

Qdec_StateType

Precondition

None.

6.2.4.4 Qdec_CounterState()

```
void Qdec_CounterState (
            const uint8 module,
            Qdec_CounterStateType counterState )
```

This function will set the state of counter.

This function will set the state of counter in 2 states:

- PAUSE when counting is paused.
- RESUME when the counting starts again to count.

Parameters

in	*module*	Name of your QDEC instance.
in	*counterState*	State of counter.

Returns

void

Precondition

None.

6.2.4.5 Qdec_ResetState()

```
void Qdec_ResetState (
            const uint8 module )
```

Reset state of module to default value.

Parameters

in	*module*	- Qdec instance ID.

Returns

void

Precondition

None.

6.2.4.6 Qdec_SetMode()

```
void Qdec_SetMode (
            const uint8 module,
            Qdec_ModeType mode )
```

Set mode.

Normal operation, all used interrupts are enabled according to the notification requests. Reduced power operation.

Parameters

in	*module*	- Qdec instance ID. mode - select QDEC_MODE_NORMAL or QDEC_MODE_SLEEP

Returns

void

Precondition

None.

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.