User Manual

for S32K3 DPGA Driver

Document Number: UM34DPGAASRR21-11 Rev0000R3.0.0 Rev. 1.0

Chapter 1

Revision History

Revision	Date	Author	Description
1.0	31.03.2023	NXP RTD Team	S32K3 Real-Time Drivers AUTOSAR 4.4 & R21-11 Version 3.0.0

Chapter 2

Introduction

This User Manual describes NXP Semiconductor DPGA for S32K3. Dpga driver configuration parameters and deviations from the specification are described in Driver chapter of this document. Dpga driver requirements and APIs are described in the Dpga driver software specification document.

2.1 Supported Derivatives

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

```
- s32m274_lqfp64
- s32m276_lqfp64
```

All of the above microcontroller devices are collectively named as S32K3.

Note: MWCT part numbers contain NXP confidential IP for Qi Wireless Power.

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
CAN	Controller Area Network
C/CPP	C and C++ Source Code
CDD	Complex Device Driver
CS	Chip Select
CTU	Cross Trigger Unit
DEM	Diagnostic Event Manager
DET	Development Error Tracer
DMA	Direct Memory Access
ECU	Electronic Control Unit
FIFO	First In First Out
LSB	Least Significant Bit
MCU	Micro Controller Unit
MSB	Most Significant Bit
N/A	Not Applicable
RAM	Random Access Memory
DPGA	Differential Programmable Gain Amplifier
SIU	Systems Integration Unit
SWS	Software Specification
XML	Extensible Markup Language

2.5 Reference List

#	Title	Version
1	S32M27x Reference Manual	Rev. 2, Draft A, 2/2023
2	S32M2xx Data Sheet	Rev. 2 RC, 12/2022

Chapter 3

Driver

3.1 Requirements

Dpga 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

3.3 Hardware Resources

The Dpga driver uses the DPGA hardware IP. For more details and availability please check the device reference manual.

3.4 Deviations from Requirements

N/A.

3.5 Driver Limitations

Dpga driver limitations:

- Blanking time was not tested
- Limited Amplifier testing
- Amplifier self-test is not supported
- Voltage monitor self-test is not supported
- Voltage monitor is not tested
- Multicore is not supported
- Exlcusive areas are not implemented

3.6 Driver usage and configuration tips

3.6.1 Dpga introduction

- In order to use the Dpga driver, the logical channel which maps a hardware unit must be first initialized using Dpga_Init() function.
- Once a channel is initialized, it cannot be initialized again until it is de-initialized using Dpga_DeInit() function.
- Dpga module amplifies the voltage of the input signal, so that it can be measured with higher (compared to the non-amplified signal) resolution by an ADC.
- In order to use Dpga module, it should be activated in McuResetConfig by checking DPGA Reset checkbox (deasserts hard reset input port of the module).
- Dpga has the following features:
  - Programmable amplification by 8, 16, 24, 32, 40, 50, 65, or 80
  - Both unipolar and bipolar input voltage ranges
  - Configurable input/output common voltage
  - Blanking time, used to protect the amplifier against saturation (support added, but has not been validated)
  - Voltage monitoring (not yet implemented in the driver)
  - Functional self-test for the amplifier and the voltage monitoring (not yet implemented)

3.6.2 Dpga Amplifier

- In order to allow the highest possible resolution and accuracy in the measurement of the input signal, you have to carefully configure the amplifier.
- Adjust the common voltage of the differential signal on the input so that the voltage on the plus and minus inputs of the amplifier cannot become negative.
- Adapt the gain of the amplifier so that in case the maximum voltage on the differential input in your application is amplified, the output of the amplifier is below its maximal output voltage.
- Compensate the offset of your amplifier for the chosen gain so that the unwanted voltage shift in the output signal of the amplifier is minimized. Best is that you measure the offset while there is 0 differential input voltage.
- Shift the common voltage of the output so that the dynamic range of the output signal is within the voltage range of the amplifier.
- The usual method to measure the Dpga output voltage is to use an on-chip ADC. However, if the ADC is not used, one can use a multimeter to measure the voltage on the AMPOUT pin. Before measuring it, one should enbale (set to 1) the Dpga Out bit in IO FuncMux register in the Application Extension Controller.

3.6.3 Dpga Blanking Time

- Depending on the gain of the amplifier, that you set in your application, and on the dynamic range of the differential input, the transistor in the amplifier could saturate. In the saturation, the reaction of the output on any change on the input is slower than usual.
- One should avoid the saturation in order to maximize the usability of the DPGA. The blanking time can help you to avoid the saturation by disconnecting the input from the amplifier during the time when high voltages can occur on the differential input, e.g. due to ringing caused by switching on the input signal.
- The inputs of the amplifier are connected to ground while the blanking time is active.
- There are 6 blanking time trigger signals. You can select which event, so a low to high or a high to low transition on any of the 6 trigger signals, can start the blanking time. Once a trigger occurs, it always starts the full blanking time defined by the number of clock cycles, no matter whether the blanking time was already active when the trigger occurred.
- During the blanking time, the differential inputs to the DPGA are disconnected from the amplifier itself.

3.7 Runtime errors

The driver generates the following DET errors at runtime.

Function	Error Code	Condition triggering the error
Dpga_Init	DPGA_E_ALREADY_INITIALIZED	Calling Dpga_Init twice in a row
Dpga_Init	DPGA_E_INIT_FAILED	Dpga module is not initialized successfully
Dpga_DeInit	DPGA_E_NOT_INITIALIZED	Dpga_Init() was not called before
Dpga_ConfigureAmplifier	DPGA_E_NOT_INITIALIZED	Dpga_Init() was not called before
Dpga_ConfigureAmplifier	DPGA_E_INVALID_POINTER	The AmplifierConfig pointer is NULL_PTR
Dpga_GetVersionInfo	DPGA_E_INVALID_POINTER	The VersionInfo pointer is NULL_PTR

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 Dpga
  - Container GeneralConfiguration
    - Parameter DpgaEnableVoltageMonitoring
    - Parameter DpgaDevErrorDetect
    - Parameter DpgaIpDevErrorDetect
    - Parameter DpgaMulticoreSupport
    - Parameter DpgaEnableUserModeSupport
    - Parameter DpgaVersionInfoApi
    - Parameter DpgaTimeoutMethod
    - Parameter DpgaTimeoutValue
  - Container DpgaChannel
    - Parameter DpgaChannelId
    - Parameter DpgaHwChannel
    - Container DpgaConfiguration
      - Parameter DpgaAmplifierGain
      - Parameter DpgaOutCommonModeVoltage
      - Parameter DpgaOffset
      - Parameter DpgaInCommonModeCoarse
      - Parameter DpgaInCommonModeFine
      - Parameter DpgaBlankingTimeDuration
      - Parameter DpgaEnableBipolarDetector
      - Parameter DpgaLowDetectLimit
      - Parameter DpgaLowDetectFilterDuration
      - Parameter DpgaHighDetectLimit
      - Parameter DpgaHighDetectFilterDuration
      - Parameter DpgaCallback
      - Container DpgaBlankingTimeTriggerList
      - Parameter DpgaBTTriggerState
  - Container CommonPublishedInformation

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

4.1 Module Dpga

Configuration of the Differential Programmable Gain Amplifier (DPGA) module.

Included containers:

- GeneralConfiguration
- DpgaChannel
- CommonPublishedInformation

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

4.2 Container GeneralConfiguration

This container contains the global configuration parameters of the Non-Autosar Dpga driver.

Included subcontainers:

- None

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

4.3 Parameter DpgaEnableVoltageMonitoring

Enable/Disable Voltage Monitoring

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

4.4 Parameter DpgaDevErrorDetect

Switches the Development Error Detection and Notification ON or OFF.

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

4.5 Parameter DpgaIpDevErrorDetect

Enables/disables development error detection for Dpga Ip.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
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

4.6 Parameter DpgaMulticoreSupport

Multicore is not supported for DPGA module.

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

4.7 Parameter DpgaEnableUserModeSupport

User Mode is not supported for DPGA module.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

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

4.8 Parameter DpgaVersionInfoApi

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
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

4.9 Parameter DpgaTimeoutMethod

Configures the timeout method for Dpga

Based on this selection a certain timeout method from OsIf will be used in the driver.

Note: If OSIF_COUNTER_SYSTEM or OSIF_COUNTER_CUSTOM are selected make sure the corresponding timer is enabled in OsIf General configuration.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

Property	Value
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	OSIF_COUNTER_DUMMY
literals	['OSIF_COUNTER_DUMMY', 'OSIF_COUNTER_SYSTEM', 'OSIF_COU↩ NTER_CUSTOM']

4.10 Parameter DpgaTimeoutValue

This is a timeout (microseconds) value which is used to wait for each synchronization transfer

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: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	65535
max	4294967295
min	1

4.11 Container DpgaChannel

This container contains the configuration (parameters) of the Dpga Controller(s).

Included subcontainers:

- DpgaConfiguration

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.12 Parameter DpgaChannelId

Identifies the Dpga channel

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	true
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	1
min	0

4.13 Parameter DpgaHwChannel

Selects hardware channel.

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	DPGA_0
literals	['DPGA_0']

4.14 Container DpgaConfiguration

This container contains the configuration (parameters) of the Dpga Controller(s).

Included subcontainers:

- DpgaBlankingTimeTriggerList

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

4.15 Parameter DpgaAmplifierGain

Select the gain of the differential amplifier.

GAIN_8: Amplify by 8

GAIN_16: Amplify by 16

GAIN_24: Amplify by 24

GAIN_32: Amplify by 32

GAIN_40: Amplify by 40

GAIN_50: Amplify by 50

GAIN_65: Amplify by 65

GAIN_80: Amplify by 80

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	GAIN_8
literals	['GAIN_8', 'GAIN_16', 'GAIN_24', 'GAIN_32', 'GAIN_40', 'GAIN_50', 'G↩ AIN_65', 'GAIN_80']

4.16 Parameter DpgaOutCommonModeVoltage

Configure the common mode voltage of the output of the amplifier.

VREF_DIV_12: Reference voltage divided by 12

VREF_DIV_6: Reference voltage divided by 6

VREF_DIV_4: Reference voltage divided by 4

VREF_DIV_2: Reference voltage divided by 2

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	VREF_DIV_2
literals	['VREF_DIV_12', 'VREF_DIV_6', 'VREF_DIV_4', 'VREF_DIV_2']

4.17 Parameter DpgaOffset

Configure the value to compensate the offset of the differential amplifier.

0: No offset

1: +3mV

2: +6mV

7: +21mV

8: No offset

9: -3mV

10: -6mV

15: -21mV

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

4.18 Parameter DpgaInCommonModeCoarse

Configure the common mode of the differential inputs of the amplifier.

NO_SHIFT: No shift

SHIFT_200: Shift with 200 uA current out of both inputs

SHIFT_100: Shift with 100 uA current out of both inputs

SHIFT_50: Shift with 50 uA current out of both inputs

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	SHIFT_0
literals	['SHIFT_0', 'SHIFT_200', 'SHIFT_100', 'SHIFT_50']

4.19 Parameter DpgaInCommonModeFine

Adjust the common mode of the differential inputs of the amplifier by

changing the current out of one of the differential inputs.

0: Input Common Mode Coarse * 1.0000 out of minus input

1: Input Common Mode Coarse * 1.0025 out of minus input

2: Input Common Mode Coarse * 1.0050 out of minus input

31: Input Common Mode Coarse * 1.0775 out of minus input

32: Input Common Mode Coarse * 1.0000 out of plus input

33: Input Common Mode Coarse * 1.0025 out of plus input

34: Input Common Mode Coarse * 1.0050 out of plus input

63: Input Common Mode Coarse * 1.0775 out of plus input

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	63
min	0

4.20 Parameter DpgaBlankingTimeDuration

Blanking Time duration, in number of module clock cycles.

The actual number of clock cycles is this number + 1.

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	255
min	0

4.21 Parameter DpgaEnableBipolarDetector

Enable the low detector. Use the low detector for bipolar measurements.

Property	Value
type	ECUC-BOOLEAN-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	false

4.22 Parameter DpgaLowDetectLimit

When the output of the amplifier is lower than the limit defined here, a low detect event is generated.

The low detect limit is actually (1 + this value) / 64 of the supply voltage.

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

Property	Value
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

4.23 Parameter DpgaLowDetectFilterDuration

The minimal time duration that the low detect must be activer before a low detect output flag is set.

This value + 1 is the actual minimal time duration in number of module clock cycles.

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	255
min	0

4.24 Parameter DpgaHighDetectLimit

When the output of the amplifier is higher than the limit defined here, a high detect event is generated.

The high detect limit is actually (48 + this value) / 64 of the supply voltage.

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

4.25 Parameter DpgaHighDetectFilterDuration

The minimal time duration that the high detect must be activer before a high detect output flag is set.

This value + 1 is the actual minimal time duration in number of module clock cycles.

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	255
min	0

4.26 Parameter DpgaCallback

Dpga callback. This function will be called for all Dpga events.

Property	Value
type	ECUC-FUNCTION-NAME-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: POST-BUILD
defaultValue	NULL_PTR

4.27 Container DpgaBlankingTimeTriggerList

List of the Blanking Time triggers

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	6
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD

4.28 Parameter DpgaBTTriggerState

Select the state of the blanking time trigger that enables the blanking time counter.

NO_EDGE: This trigger cannot start the BT counter

FALLING_EDGE: The falling edge of this trigger starts the BT counter from 0

RISING_EDGE: The rising edge of this trigger starts the BT counter from 0

BOTH_EDGES: Both edges of this trigger start the BT counter from 0

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	NO_EDGE
literals	['NO_EDGE', 'FALLING_EDGE', 'RISING_EDGE', 'BOTH_EDGES']

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

Property	Value
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	7
max	7
min	7

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

Property	Value
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
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false

Property	Value
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	3
max	3
min	3

4.35 Parameter SwMinorVersion

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

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 vendor specific.

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

Property	Value
defaultValue	0
max	0
min	0

4.37 Parameter VendorApiInfix

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that the 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	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
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

Property	Value
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

This chapter describes the Tresos configuration plug-in for the Dpga Driver. The most of the parameters are described below.

Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

Chapter 6

Module Documentation

6.1 Dpga

6.1.1 Detailed Description

Macros

- #define DPGA_INIT_ID

    APIs service IDs.
- #define DPGA_DEINIT_ID
- #define DPGA_CONFIGURE_AMPLIFIER_ID

Types Reference

- typedef Dpga_Ip_AmplifierConfigType Dpga_AmplifierConfigType

    Differential Amplifier configuration structure This structure is used to provide the configuration parameters for the Differential Amplifier.
- typedef Dpga_Ip_ConfigType Dpga_ConfigType

    Dpga module configuration structure This structure is used to provide the configuration parameters for the Dpga module.

Function Reference

- Std_ReturnType Dpga_Init (const Dpga_ConfigType ∗const Config)

    Initialize the DPGA module.
- Std_ReturnType Dpga_DeInit (void)

    Deinitialize the DPGA module.
- Std_ReturnType Dpga_ConfigureAmplifier (const Dpga_AmplifierConfigType ∗AmplifierConfig)

    Reload Amplifier configuration.

6.1.2 Macro Definition Documentation

6.1.2.1 DPGA_INIT_ID

```
#define DPGA_INIT_ID
```

APIs service IDs.

Service IDs for the DPGA driver APIs. Dpga_Init() ID

Definition at line 134 of file CDD_Dpga.h.

6.1.2.2 DPGA_DEINIT_ID

```
#define DPGA_DEINIT_ID
```

Dpga_Deinit() ID

Definition at line 135 of file CDD_Dpga.h.

6.1.2.3 DPGA_CONFIGURE_AMPLIFIER_ID

```
#define DPGA_CONFIGURE_AMPLIFIER_ID
```

Dpga_ConfigureAmplifier() ID

Definition at line 136 of file CDD_Dpga.h.

6.1.3 Types Reference

6.1.3.1 Dpga_AmplifierConfigType

```
typedef Dpga_Ip_AmplifierConfigType Dpga_AmplifierConfigType
```

Differential Amplifier configuration structure This structure is used to provide the configuration parameters for the Differential Amplifier.

Definition at line 121 of file CDD_Dpga_Types.h.

6.1.3.2 Dpga_ConfigType

```
typedef Dpga_Ip_ConfigType Dpga_ConfigType
```

Dpga module configuration structure This structure is used to provide the configuration parameters for the Dpga module.

Definition at line 128 of file CDD_Dpga_Types.h.

6.1.4 Function Reference

6.1.4.1 Dpga_Init()

```
Std_ReturnType Dpga_Init (
            const Dpga_ConfigType *const Config )
```

Initialize the DPGA module.

This function initializes the DPGA module:

- Maps the logical channel to the hardware channel
- Initializes the channel

Parameters

in	*Config*	Pointer to the configuration structure

6.1.4.2 Dpga_DeInit()

```
Std_ReturnType Dpga_DeInit (
            void )
```

Deinitialize the DPGA module.

This function deinitializes the DPGA module to the reset values. The driver must be initialized before calling Dpga_DeInit().

6.1.4.3 Dpga_ConfigureAmplifier()

```
Std_ReturnType Dpga_ConfigureAmplifier (
            const Dpga_AmplifierConfigType * AmplifierConfig )
```

Reload Amplifier configuration.

This function is used to configure the Amplifier at runtime. The user can store specific configuration values after device power up and reload them after each device reset.

Parameters

`in`	*AmplifierConfig*	Pointer to the Amplifier configuration structure

Returns

Std_ReturnType

6.2 Dpga_ip

6.2.1 Detailed Description

Data Structures

- struct Dpga_Ip_AmplifierConfigType

  Differential Amplifier configuration structure. More...
- struct Dpga_Ip_ConfigType

  Dpga Ip module configuration structure. More...

Types Reference

- typedef void(∗ Dpga_Ip_CallbackType) (Dpga_Ip_EventType Event)

  Callback used for DPGA module notifications. Possible notifications: low/high voltage detect, parity check.

Enum Reference

- enum Dpga_Ip_StatusType

  Dpga Ip Status.
- enum Dpga_Ip_EventType

  Events which can trigger DPGA notification callback.

Function Reference

- Dpga_Ip_StatusType Dpga_Ip_Init (const uint8 Instance, const Dpga_Ip_ConfigType ∗const Config)

  Initialize the DPGA module.
- Dpga_Ip_StatusType Dpga_Ip_DeInit (const uint8 Instance)

  Deinitialize the DPGA module.
- Dpga_Ip_StatusType Dpga_Ip_ConfigureAmplifier (const uint8 Instance, const Dpga_Ip_AmplifierConfigType ∗AmplifierConfig)

  Reload Amplifier configuration.
- Dpga_Ip_StatusType Dpga_Ip_IRQHandler (const uint8 Instance)

  Handle DPGA interrupt.

6.2.2 Data Structure Documentation

6.2.2.1 struct Dpga_Ip_AmplifierConfigType

Differential Amplifier configuration structure.

This structure is used to provide the configuration parameters for the Differential Amplifier

Definition at line 136 of file Dpga_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	AmplifierGain	The gain of the Amplifier
uint8	OutCommonModeVoltage	Common Mode Voltage of the output of the Amplifier
uint8	AmplifierOffset	Amplifier offset
uint8	InCommonModeCoarse	The common mode of the differential inputs of the Amplifier
uint8	InCommonModeFine	Adjust the common mode of the differential inputs of the amplifier

6.2.2.2 struct Dpga_Ip_ConfigType

Dpga Ip module configuration structure.

This structure is used to provide the configuration parameters for the Dpga Ip module.

Definition at line 151 of file Dpga_Ip_Types.h.

Data Fields

Type	Name	Description
Dpga_Ip_AmplifierConfigType *	AmplifierConfig	Pointer to the Amplifier Config structure
uint8	BlankingTimeDuration	Blanking time duration, in number of module cycles
uint8	BTTriggerStateArray[DPGA_IP_BT_NUM_OF_TRIGGERS]	State of the Blanking Time Triggers 0 - 5: on which edge to start the counter for each trigger
Dpga_Ip_CallbackType	DpgaCallback	Dpga callback function. This will be called for all Dpga events.

6.2.3 Types Reference

6.2.3.1 Dpga_Ip_CallbackType

```
typedef void(* Dpga_Ip_CallbackType) (Dpga_Ip_EventType Event)
```

Callback used for DPGA module notifications. Possible notifications: low/high voltage detect, parity check.

Definition at line 128 of file Dpga_Ip_Types.h.

6.2.4 Enum Reference

6.2.4.1 Dpga_Ip_StatusType

```
enum Dpga_Ip_StatusType
```

Dpga Ip Status.

Definition at line 96 of file Dpga_Ip_Types.h.

6.2.4.2 Dpga_Ip_EventType

```
enum Dpga_Ip_EventType
```

Events which can trigger DPGA notification callback.

Definition at line 109 of file Dpga_Ip_Types.h.

6.2.5 Function Reference

6.2.5.1 Dpga_Ip_Init()

```
Dpga_Ip_StatusType Dpga_Ip_Init (
             const uint8 Instance,
             const Dpga_Ip_ConfigType *const Config )
```

Initialize the DPGA module.

This function initializes the DPGA module.

Parameters

in	*Instance*	DPGA instance
in	*Config*	Pointer to the configuration structure

6.2.5.2 Dpga_Ip_DeInit()

```
Dpga_Ip_StatusType Dpga_Ip_DeInit (
             const uint8 Instance )
```

Deinitialize the DPGA module.

This function deinitializes the DPGA module. The driver can't be used until initialized again.

Parameters

in	*Instance*	DPGA instance

6.2.5.3 Dpga_Ip_ConfigureAmplifier()

```
Dpga_Ip_StatusType Dpga_Ip_ConfigureAmplifier (
            const uint8 Instance,
            const Dpga_Ip_AmplifierConfigType * AmplifierConfig )
```

Reload Amplifier configuration.

This function is used to configure the Amplifier at runtime. The user can store specific configuration values after device power up and reload them after each device reset.

Parameters

in	*Instance*	DPGA instance
in	*AmplifierConfig*	Pointer to the Amplifier configuration structure

6.2.5.4 Dpga_Ip_IRQHandler()

```
Dpga_Ip_StatusType Dpga_Ip_IRQHandler (
            const uint8 Instance )
```

Handle DPGA interrupt.

This function handles DPGA interrupt events.

Parameters

in	*Instance*	DPGA instance

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support