# User Manual

for S32K3 AE Driver

Document Number: UM34AEASRR21-11 Rev0000R 3.0.0 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 Semiconductors' AE Driver for S32K3XX.

AE Driver configuration parameters description can be found in the Tresos Configuration Plugin section.

## 2.1 Supported Derivatives

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

- s32k310_mqfp100
- s32k310_lqfp48
- s32k311_mqfp100 / MWCT2015S_mqfp100
- s32k311_lqfp48
- s32k312_mqfp100 / MWCT2016S_mqfp100
- s32k312_mqfp172 / MWCT2016S_mqfp172
- s32k314_mqfp172
- s32k314_mapbga257
- s32k322_mqfp100 / MWCT2D16S_mqfp100
- s32k322_mqfp172 / MWCT2D16S_mqfp172

- s32k324_mqfp172 / MWCT2D17S_mqfp172
- s32k324_mapbga257
- s32k341_mqfp100
- s32k341_mqfp172
- s32k342_mqfp100
- s32k342_mqfp172
- s32k344_mqfp172
- s32k344_mapbga257
- s32k394_mapbga289
- s32k396_mapbga289
- s32k358_mqfp172
- s32k358_mapbga289
- s32k328_mqfp172
- s32k328_mapbga289
- s32k338_mqfp172
- s32k338_mapbga289
- s32k348_mqfp172
- s32k348_mapbga289
- 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 |
| AUTOSAR | Automotive Open System Architecture |
| DEM | Diagnostic Event Manager |
| DET | Default Error Tracer |
| AE | Application Extension |
| AEC | Application Extension Controller |
| HVM | High Voltage Module |
| VM | Voltage Monitor |
| HVI | High Voltage Input |
| SPI | Serial Peripheral Interface |
| MCU | Micro controller Unit |
| N/A | Not Available |
| RAM | Random Access Memory |

- The term "Ae Driver" is related to the software handling the Appilcation Extension.
- The term "Application" is used for the software utilizing the AE Driver.

## 2.5 Reference List

| # | Title | Version |
|---|---|---|
| 1 | S32M27x Reference Manual | Rev.2, Draft A, — 02/2023 |
| 2 | S32M2xx Data Sheet | Rev. 2 RC — 12/2022 |

# Chapter 3

# Driver



## 3.1 Requirements

None.

## 3.2 Driver Design Summary

The AE Driver controls the Application Extension Controller (AEC) module and High Voltage Monitor(HVM) of the S32K3XX device. It provides the following features:

- For AEC:
- A SPI interface for microcontroller communication to read/ write to AE's registers for polling and interrupt mode.
- Read Frame Counter status.
- Allow to control Raw bit.
- Decode global heath status of frame.

- Read status of faults and events through interrupt_out raised.

- For HVM:
- A voltage divider to divide down different input voltage levels to the system voltage.
- Event generation, when the input logic level changes. This works also when the clock is off, so you can use this digital input interrupt for example to wake up the device from sleep mode.
- Event observation through interrupt generation, reading of an interrupt flag in register, or polling the event status from a register.
- Analog output, e.g. for reading the input voltage with an on-chip ADC.
- Selectable pull-up and pull-down connections on the input for open input detection.
- Fixed input voltage division by 11 in the connected HVI when a voltage monitor is enabled.
- Event generation, when the input voltage is out of customer-defined limits. You can configure the upper and lower limits of the allowed voltage range of the input signal.
- Observation of any voltage out of defined limits event through interrupt generation, reading of an interrupt flag in a register, or polling the event status from a register.

## 3.3 Hardware Resources

The SPI interface communication to microcontroller pin mapping can be done using the file **S32M27x_IOMU↩ X.xlsx** attached to the Reference Manual.

## 3.4 Deviations from Requirements

None.

## 3.5 Driver Limitations

None.

## 3.6 Driver usage and configuration tips

None

## 3.7 Runtime errors

None.

## 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 Ae
  - Container AeGeneral
    - Parameter AeVersionInfoApi
    - Parameter AeDevErrorDetect
    - Parameter AeRawbitEn
    - Container AeGeneralVendorSpecific
      - Parameter AeTimeoutMethod
      - Parameter AeTimeoutDuration
  - Container AeHvmVmConfig
    - Parameter AeHvmVoltageMonitorInputSelect
    - Parameter AeVmHighVoltageDetectInterruptEn
    - Parameter AeVmLowVoltageDetectInterruptEn
    - Parameter AeHvmHighVoltageDetectEn
    - Parameter AeHvmLowVoltageDetectEn
    - Parameter AeHvmLowDetectRefVoltage
    - Parameter AeHvmHighDetectRefVoltage
  - Container AeHvmHviConfig
    - Container AeHvmHviGeneralConfig
      - Parameter AeHvmHviAnalogchannel
      - Parameter AeHvmAnalogInput
      - Parameter AeHvmUnderControlIsrCallback
    - Container AeHvmHviInstanceConfig
      - Parameter AeHvmHviChannelIdx
      - Parameter AeHVIInputDigitalEn
      - Parameter AeHVIInterruptInputDigitalEn
      - Parameter AeHviDigitalInputPolarity
      - Parameter AeHviDividerSelect
      - Parameter AeHVIPullUpEn
      - Parameter AeHVIPullDownEn

- Container AeSpi
  - Container AeSpiSequence
    - Parameter AeSpiAccessSynchronous
    - Reference AeSpiSequenceName
- Container AeIrqConfig
  - Container AeIntConfig
    - Parameter AeIntName
    - Parameter AeIsrEnabled
    - Parameter AeIsrPriority
    - Parameter AeIntHandler
- Container CommonPublishedInformation
  - Parameter ModuleId
  - Parameter VendorId
  - Parameter VendorApiInfix
  - Parameter ArReleaseMajorVersion
  - Parameter ArReleaseMinorVersion
  - Parameter ArReleaseRevisionVersion
  - Parameter SwMajorVersion
  - Parameter SwMinorVersion
  - Parameter SwPatchVersion

## 4.1 Module Ae

Configuration of the Ae (Application Extension Driver) module.

Included containers:

- AeGeneral
- AeHvmVmConfig
- AeHvmHviConfig
- AeSpi
- AeIrqConfig
- CommonPublishedInformation

| Property | Value |
|---|---|
| type | ECUC-MODULE-DEF |
| lowerMultiplicity | 0 |
| upperMultiplicity | 1 |
| postBuildVariantSupport | true |
| supportedConfigVariants | VARIANT-PRE-COMPILE, VARIANT-POST-BUILD |

## 4.2 Container AeGeneral

General configuration of Ae Driver module.

Included subcontainers:

- AeGeneralVendorSpecific

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

## 4.3 Parameter AeVersionInfoApi

Enables / Disables Ae_GetVersionInfo API.

| 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 |

## 4.4 Parameter AeDevErrorDetect

Enables / Disables development error detection.

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

| Property | Value |
|---|---|
| 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 AeRawbitEn

Enables / Disables read after write

| 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 |

## 4.6 Container AeGeneralVendorSpecific

General vendor-specific configuration of Ae Driver.

Included subcontainers:

- None

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

## 4.7 Parameter AeTimeoutMethod

Configures the timeout method.

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

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

Note: Implementation Specific Parameter.

| Property | Value |
|---|---|
| type | ECUC-ENUMERATION-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 | OSIF_COUNTER_DUMMY |
| literals | ['OSIF_COUNTER_SYSTEM', 'OSIF_COUNTER_CUSTOM', 'OSIF_CO↩ UNTER_DUMMY'] |

## 4.8 Parameter AeTimeoutDuration

The unit of measurement is given in number of microseconds.

| 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 | 1000 |
| max | 65535 |
| min | 1 |

## 4.9 Container AeHvmVmConfig

General Configuration Volatage Monitor.

Included subcontainers:

- None

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

## 4.10 Parameter AeHvmVoltageMonitorInputSelect

Select the voltage that is observed by the voltage monitor.

Note: Implementation Specific Parameter.

| 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 | VM_HIGH_VOLTAGE_INPUT |
| literals | ['VM_HIGH_VOLTAGE_INPUT', 'VM_SENSE_VOLTAGE'] |

## 4.11 Parameter AeVmHighVoltageDetectInterruptEn

This parameter is used to enable/disable the digital input interrupt of High Voltage Detect.

| 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 |

## 4.12 Parameter AeVmLowVoltageDetectInterruptEn

This parameter is used to enable/disable the digital input interrupt of Low Voltage Detect.

| 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 |

## 4.13 Parameter AeHvmHighVoltageDetectEn

Enables / Disables the voltage monitor for high voltage detection.

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

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

## 4.14 Parameter AeHvmLowVoltageDetectEn

Enables / Disables the voltage monitor for Low voltage detection.

| 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 |

## 4.15 Parameter AeHvmLowDetectRefVoltage

Configures the reference voltage level for low voltage detection.

This reference voltage level is in datasheet.

Note: Implementation Specific Parameter.

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

| Property | Value |
|---|---|
| multiplicityConfigClasses | N/A |
| postBuildVariantValue | true |
| valueConfigClasses | VARIANT-PRE-COMPILE: PRE-COMPILE |
| | VARIANT-POST-BUILD: POST-BUILD |
| defaultValue | VM_LBI1 |
| literals | ['VM_LBI1', 'VM_LBI2', 'VM_LBI3', 'VM_LBI4'] |

## 4.16 Parameter AeHvmHighDetectRefVoltage

Configures the reference voltage level for high voltage detection.

This reference voltage level is in datasheet.

Note: Implementation Specific Parameter.

| 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 | VM_HBI1 |
| literals | ['VM_HBI1', 'VM_HBI2'] |

## 4.17 Container AeHvmHviConfig

Configuration of High Voltage Input for each instances.

Included subcontainers:

- AeHvmHviGeneralConfig
- AeHvmHviInstanceConfig

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

## 4.18 Container AeHvmHviGeneralConfig

Included subcontainers:

- None

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

## 4.19 Parameter AeHvmHviAnalogchannel

Configures the Analog channel.

Enable the analog input for one HVI.

Note: Implementation Specific Parameter.

| 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 | HVI0_CHANNEL |
| literals | ['HVI0_CHANNEL', 'HVI1_CHANNEL'] |

## 4.20 Parameter AeHvmAnalogInput

Configures the Analog Input.

Select the analog input that is given out on the analog output of the instance that is selected.

Note: Implementation Specific Parameter.

| 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 | HVI_ANALOG_INPUT_BUFFERED |
| literals | ['HVI_ANALOG_OUT_OHMIC', 'HVI_ANALOG_INPUT_BUFFERED', 'HVI_ANALOG_DIVIDE_DOWN_HIGH_INPUT_BUFFERED', 'HVI_A↩ NALOG_INPUT_UNBUFFERED'] |

## 4.21 Parameter AeHvmUnderControlIsrCallback

Enables callback function was handled by user for Hvm_Ip_pfReportCallback, if is disable, driver doesn't do anything, callback function is empty.

| 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 |

## 4.22 Container AeHvmHviInstanceConfig

Container information to congiure for each HVI instance.

Included subcontainers:

- None

| Property | Value |
|---|---|
| type | ECUC-PARAM-CONF-CONTAINER-DEF |
| lowerMultiplicity | 0 |
| upperMultiplicity | 255 |
| postBuildVariantMultiplicity | true |
| multiplicityConfigClasses | VARIANT-POST-BUILD: POST-BUILD |
| | VARIANT-PRE-COMPILE: PRE-COMPILE |

## 4.23 Parameter AeHvmHviChannelIdx

High Voltage Input channel Index

| 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 | N/A |
| min | 0 |

## 4.24 Parameter AeHVIInputDigitalEn

This parameter is used to enable/disable the digital input.

| 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 |

## 4.25 Parameter AeHVIInterruptInputDigitalEn

This parameter is used to enable/disable the digital input interrupt of HVI channel.

| 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 |

## 4.26 Parameter AeHviDigitalInputPolarity

Configures the Digital Input Polarity.

Configure whether the high or the low state on the digital input triggers an event.

Note: Implementation Specific Parameter.

| 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 | HVI_DIGITAL_INPUT_POLARITY_PULLDOWN |
| literals | ['HVI_DIGITAL_INPUT_POLARITY_PULLDOWN', 'HVI_DIGITAL_IN↩ PUT_POLARITY_PULLUP'] |

## 4.27 Parameter AeHviDividerSelect

Configures the ratio of the voltage division.

The pull down must be enabled to divide by the ratios selectable here. To set a divider ratio of 1, the pull down must be disabled.

Note: Implementation Specific Parameter.

| 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 | HVI_VOLTAGE_DIVISION_2 |
| literals | ['HVI_VOLTAGE_DIVISION_2', 'HVI_VOLTAGE_DIVISION_6', 'HVI_↩ VOLTAGE_DIVISION_11', 'HVI_VOLTAGE_DIVISION_16'] |

## 4.28 Parameter AeHVIPullUpEn

This parameter is used to enable/disable the pull up on the input of each instance HVI.

| 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 |

## 4.29 Parameter AeHVIPullDownEn

This parameter is used to enable/disable the pull down on the input of each instance HVI.

| 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 |

## 4.30 Container AeSpi

Configuration of one Application Extension SPI access (if SPI is used).

Included subcontainers:

- AeSpiSequence

| Property | Value |
|---|---|
| type | ECUC-PARAM-CONF-CONTAINER-DEF |

| Property | Value |
|---|---|
| lowerMultiplicity | 1 |
| upperMultiplicity | 1 |
| postBuildVariantMultiplicity | N/A |
| multiplicityConfigClasses | N/A |

## 4.31 Container AeSpiSequence

Container gives Ae driver information about one SPI sequence.

Included subcontainers:

- None

| Property | Value |
|---|---|
| type | ECUC-PARAM-CONF-CONTAINER-DEF |
| lowerMultiplicity | 1 |
| upperMultiplicity | 255 |
| postBuildVariantMultiplicity | true |
| multiplicityConfigClasses | VARIANT-POST-BUILD: PRE-COMPILE |
| | VARIANT-PRE-COMPILE: PRE-COMPILE |

## 4.32 Parameter AeSpiAccessSynchronous

This parameter is used to define whether the access to the Spi sequence is synchronous or asynchronous.

| 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 |

## 4.33 Reference AeSpiSequenceName

Reference to a Spi sequence configuration container.

| Property | Value |
|---|---|
| type | ECUC-REFERENCE-DEF |
| origin | NXP |
| lowerMultiplicity | 1 |
| upperMultiplicity | 1 |
| postBuildVariantMultiplicity | N/A |
| multiplicityConfigClasses | N/A |
| postBuildVariantValue | false |
| valueConfigClasses | VARIANT-POST-BUILD: PRE-COMPILE |
| | VARIANT-PRE-COMPILE: PRE-COMPILE |
| requiresSymbolicNameValue | False |
| destination | /AUTOSAR/EcucDefs/Spi/SpiDriver/SpiSequence |

## 4.34 Container AeIrqConfig

Generic configuration for the interrupts (events, faults).

Included subcontainers:

- AeIntConfig

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

## 4.35 Container AeIntConfig

Vendor specific:

Configuration for interrupt requests.

Included subcontainers:

- None

| Property | Value |
|---|---|
| type | ECUC-PARAM-CONF-CONTAINER-DEF |
| lowerMultiplicity | 25 |
| upperMultiplicity | 25 |
| postBuildVariantMultiplicity | N/A |
| multiplicityConfigClasses | N/A |

## 4.36 Parameter AeIntName

Vendor specific:

Interrupt Name caused by faults or events.

| Property | Value |
|---|---|
| type | ECUC-ENUMERATION-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 | LIN_INT |
| literals | ['LIN_INT', 'OCD_VDDE', 'TEMP_WDG_PMC', 'TEMP_WDG_PHY', 'CAN_INT', 'LVD_VDDC', 'WAKEUP', 'HVI_ACTIVE', 'HVI_AE_SUPPLY', 'STATERESET', 'FRAMEWIDTH', 'GDU_FAULT_PROT', 'GDU_INT', 'DPGA_OC_NEG', 'DPGA_OC_POS', 'PMC_VLS', 'RAW_FAILED', 'CHKSUM_ERR', 'OBSCHK_ERR', 'XFER_ERR', 'EVENT_EXPIRED', 'MCU_SUPPLY', 'HVD_AE_INTERN', 'ILL_TEST', 'ALIVE_WDG'] |

## 4.37 Parameter AeIsrEnabled

Vendor specific: Enable to raise interrupt_out for events/faults which owned by AEC.

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

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

## 4.38 Parameter AeIsrPriority

Set priority to handle for event or fault which caused interrupt.

Priority's value smaller that will be handled first.

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

## 4.39 Parameter AeIntHandler

Function to be installed as the interrupt handler.

| Property | Value |
|---|---|
| type | ECUC-FUNCTION-NAME-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 | undefined_handler |

## 4.40 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 |

## 4.41 Parameter ModuleId

Module ID of this module from the BSW 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 | 250 |
| max | 250 |
| min | 250 |

## 4.42 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 |

## 4.43 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][API name from SWS].

E.g. Assuming that the VendorId of the implementor is 123 and the implementer chose a

VendorApiInfix of 'v11r456' an API named 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 |

| Property | Value |
|---|---|
| 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 | |

## 4.44 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 |

## 4.45 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 |

| Property | Value |
|---|---|
| postBuildVariantValue | false |
| valueConfigClasses | VARIANT-POST-BUILD: PUBLISHED-INFORMATION |
| | VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION |
| defaultValue | 7 |
| max | 7 |
| min | 7 |

## 4.46 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 |

## 4.47 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 |
| valueConfigClasses | VARIANT-POST-BUILD: PUBLISHED-INFORMATION |

| Property | Value |
|---|---|
| | VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION |
| defaultValue | 3 |
| max | 3 |
| min | 3 |

## 4.48 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 |

## 4.49 Parameter SwPatchVersion

Patch 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 |

| Property | Value |
|---|---|
| max | 0 |
| min | 0 |

This chapter describes the Tresos configuration plug-in for the Ae 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 AEC Driver

### 6.1.1 Detailed Description

#### 6.1.1.1 Functions Expectation called for read/write value from/to AE's registers in polling/interrupt mode:

- Aec_Ip_Init
- Aec_Ip_SpiWrite
- Aec_Ip_SpiRead

### Data Structures

- struct Aec_Ip_SpiConfigType

  *SPI configuration for access Implements : Aec_Ip_SpiConfigType_Class.* *More...*
- struct Aec_Ip_IntRouteConfigType

  *Structure storing the routing and handler configuration for an interrupt request.* *More...*

### Types Reference

- typedef void(∗ Aec_Ip_IntHandlerType) (void)

  *Interrupt handler type.*

### Enum Reference

- enum Aec_Ip_StatusType

### Function Reference

- Aec_Ip_StatusType Aec_Ip_Init (const Aec_Ip_ConfigType *AecConfig)
  *Initializes the Aec Driver.*
- Aec_Ip_StatusType Aec_Ip_SpiWrite (uint32 RegAddr, uint8 DataWidth, uint32 Data)
  *Write Value to indexed register by Register address through SPI.*
- Aec_Ip_StatusType Aec_Ip_SpiRead (uint32 RegAddr, uint8 DataWidth, uint32 *Data)
  *Read Value return from indexed register by Register address through SPI.*
- boolean Aec_Ip_Spi_Frame_Counter (boolean IsVerify, uint8 ExpectedFrameCount, uint8 *ActualFrameCount)
  *Verify for correctness the received frame counter or just read number of frames.*
- uint16 Aec_Ip_DecodeGlobalHeathStatus (void)
  *Return status of global heath.*

## 6.1.2 Data Structure Documentation

### 6.1.2.1 struct Aec_Ip_SpiConfigType

SPI configuration for access Implements : Aec_Ip_SpiConfigType_Class.

Definition at line 88 of file Aec_Ip_Types.h.

### 6.1.2.2 struct Aec_Ip_IntRouteConfigType

Structure storing the routing and handler configuration for an interrupt request.

Definition at line 114 of file Aec_Ip_Types.h.

#### Data Fields

- uint16 IsrMask
  *Name of event/fault.*
- uint8 IsrPriority
  *Priority for event/fault.*
- Aec_Ip_IntHandlerType pfHandler
  *Interrupt handler.*

#### 6.1.2.2.1 Field Documentation

##### 6.1.2.2.1.1 IsrMask `uint16 IsrMask`

Name of event/fault.

Definition at line 117 of file Aec_Ip_Types.h.

#### 6.1.2.2.1.2 IsrPriority uint8 IsrPriority

Priority for event/fault.

Definition at line 119 of file Aec_Ip_Types.h.

#### 6.1.2.2.1.3 pfHandler Aec_Ip_IntHandlerType pfHandler

Interrupt handler.

Definition at line 121 of file Aec_Ip_Types.h.

### 6.1.3 Types Reference

#### 6.1.3.1 Aec_Ip_IntHandlerType

```
typedef void(* Aec_Ip_IntHandlerType) (void)
```

Interrupt handler type.

Definition at line 108 of file Aec_Ip_Types.h.

### 6.1.4 Enum Reference

#### 6.1.4.1 Aec_Ip_StatusType

```
enum Aec_Ip_StatusType
```

Enumerator

| | |
|---|---|
| AEC_STATUS_SUCCESS | Generic operation success status |
| AEC_STATUS_ERROR | Generic operation failure status |
| AEC_STATUS_TIMEOUT | Generic operation timeout status |

Definition at line 95 of file Aec_Ip_Types.h.

### 6.1.5 Function Reference

#### 6.1.5.1 Aec_Ip_Init()

```
Aec_Ip_StatusType Aec_Ip_Init (
            const Aec_Ip_ConfigType * AecConfig )
```

Initializes the Aec Driver.

The configuration pointer is internally stored and the driver is initialized.

Return AEC_STATUS_SUCCESS if initialized successful. AEC_STATUS_TIMEOUT if initialized failed.

#### 6.1.5.2 Aec_Ip_SpiWrite()

```
Aec_Ip_StatusType Aec_Ip_SpiWrite (
            uint32 RegAddr,
            uint8 DataWidth,
            uint32 Data )
```

Write Value to indexed register by Register address through SPI.

Parameters

| | | |
|---|---|---|
| in | *RegAddr* | Register address want to write to. |
| in | *DataWidth* | Size of register in AEC that want to read/write. |
| in | *Data* | Value want to write to indexed register. |

#### 6.1.5.3 Aec_Ip_SpiRead()

```
Aec_Ip_StatusType Aec_Ip_SpiRead (
            uint32 RegAddr,
            uint8 DataWidth,
            uint32 * Data )
```

Read Value return from indexed register by Register address through SPI.

Parameters

| | | |
|---|---|---|
| in | *RegAddr* | Register address want to read from. |
| in | *DataWidth* | Size of register in AEC that want to read/write. |
| out | *Data* | Value want to read from indexed register. |

#### 6.1.5.4 Aec_Ip_Spi_Frame_Counter()

```
boolean Aec_Ip_Spi_Frame_Counter (
            boolean IsVerify,
```

```
            uint8 ExpectedFrameCount,
            uint8 * ActualFrameCount )
```

Verify for correctness the received frame counter or just read number of frames.

Parameters

| | | |
|---|---|---|
| in | *IsVerify* | want to verify for correctness for frame counter. |
| in | *ExpectedFrameCount* | Number of frame expected get. |
| out | *ActualFrameCount* | Number of frame get in real. |

return TRUE if verify correct.

#### 6.1.5.5 Aec_Ip_DecodeGlobalHeathStatus()

```
uint16 Aec_Ip_DecodeGlobalHeathStatus (
            void  )
```

Return status of global heath.

## 6.2 HVM Driver

### 6.2.1 Detailed Description

#### 6.2.1.1 Functions Expectation called first for configuration initilized:

- Hvm_Ip_Init

### Enum Reference

- enum Hvm_Ip_StatusType
- enum Hvm_Ip_OutputStatusType

### Function Reference

- Hvm_Ip_StatusType Hvm_Ip_Init (const Hvm_Ip_ConfigType *pHvmConfig)

  *Initializes the Hvm Driver.*
- Hvm_Ip_StatusType Hvm_Ip_AdcMeasureConfig (const Ae_HviHighVoltageInputConfig *pHviHighVolt↩ Config, Ae_AnalogInputType AnalogInputSelect)

  *Initializes the configuration for analog input for measurement.*
- Hvm_Ip_StatusType Hvm_Ip_DigitalInputEventConfig (const Ae_HviChannelConfig *pHviConfig)

  *Initializes the configuration for digital input.*
- Hvm_Ip_StatusType Hvm_Ip_VoltageMonitorConfig (const Ae_VmConfig *pAeVmConfig, const Ae_Hvi↩ HighVoltageInputConfig *pHighVoltageInput)

  *Initializes the configuration for Voltage Monitor.*
- void Hvm_Ip_ReadFaultsEventsStatus (void)

  *Check status flag used in polling mode.*
- void Hvm_Ip_FaultsEventsProccess (void)

  *Check and clear interrupt flag used in interrupt mode.*

### 6.2.2 Enum Reference

#### 6.2.2.1 Hvm_Ip_StatusType

```
enum Hvm_Ip_StatusType
```

Enumerator

| | |
|---|---|
| HVM_STATUS_SUCCESS | Generic operation success status |
| HVM_STATUS_ERROR | Generic operation failure status |
| HVM_STATUS_TIMEOUT | Generic operation timeout status |

Definition at line 71 of file Hvm_Ip_Types.h.

#### 6.2.2.2 Hvm_Ip_OutputStatusType

```
enum Hvm_Ip_OutputStatusType
```

Enumerator

| | |
|---|---|
| HVM_STATUS_NOTHING_DETECT | Generic operation nothing detect status |
| HVM_STATUS_HIGH_DETECT | Generic operation high voltage detect status |
| HVM_STATUS_LOW_DETECT | Generic operation low voltage detect status |
| HVM_STATUS_DIGITAL_INPUT_ACTIVE | Generic operation digital input active status |
| HVM_STATUS_OUTPUT_ERROR | Generic operation failure status |

Definition at line 80 of file Hvm_Ip_Types.h.

### 6.2.3 Function Reference

#### 6.2.3.1 Hvm_Ip_Init()

```
Hvm_Ip_StatusType Hvm_Ip_Init (
            const Hvm_Ip_ConfigType * pHvmConfig )
```

Initializes the Hvm Driver.

The configuration pointer is internally stored and the driver is initialized.

Parameters

| | | |
|---|---|---|
| in | *HvmConfig* | Pointer to the module configuration structure |

Return values

| | |
|---|---|
| *HVM_STATUS_SUCCESS* | The initialization was successful. |

#### 6.2.3.2 Hvm_Ip_AdcMeasureConfig()

```
Hvm_Ip_StatusType Hvm_Ip_AdcMeasureConfig (
            const Ae_HviHighVoltageInputConfig * pHviHighVoltConfig,
            Ae_AnalogInputType AnalogInputSelect )
```

Initializes the configuration for analog input for measurement.

Base on configuration to select analog input, HVI channel, divider if high voltage input selected.

Parameters

| | | |
|---|---|---|
| in | *pHviHighVoltConfig* | Pointer to configuration for High Voltage Input. AnalogInputSelect Type of analog input. |

Return values

| | |
|---|---|
| *HVM_STATUS_SUCCESS* | The initialization was successful. |

#### 6.2.3.3 Hvm_Ip_DigitalInputEventConfig()

```
Hvm_Ip_StatusType Hvm_Ip_DigitalInputEventConfig (
            const Ae_HviChannelConfig * pHviConfig )
```

Initializes the configuration for digital input.

Initializes the configuration for digital input include: voltage input source, divider, mode polling or interrupt used.

Parameters

| | | |
|---|---|---|
| in | *pHviConfig* | HVI channels configuration. |

Return values

| | |
|---|---|
| *HVM_STATUS_SUCCESS* | The initialization was successful. |

#### 6.2.3.4 Hvm_Ip_VoltageMonitorConfig()

```
Hvm_Ip_StatusType Hvm_Ip_VoltageMonitorConfig (
            const Ae_VmConfig * pAeVmConfig,
            const Ae_HviHighVoltageInputConfig * pHighVoltageInput )
```

Initializes the configuration for Voltage Monitor.

Initializes the configuration for High/Low voltage detect.

Parameters

| | | |
|---|---|---|
| in | *pAeVmConfig* | Pointer to configuration of VM. |
| in | *pHighVoltageInput* | Pointer to configuration of High Voltage Input. |

Return values

| | |
|---|---|
| *HVM_STATUS_SUCCESS* | The initialization was successful. |

#### 6.2.3.5 Hvm_Ip_ReadFaultsEventsStatus()

```
void Hvm_Ip_ReadFaultsEventsStatus (
            void  )
```

Check status flag used in polling mode.

This function used to read status flag for digital input, high voltage, low voltage detect.

#### 6.2.3.6 Hvm_Ip_FaultsEventsProccess()

```
void Hvm_Ip_FaultsEventsProccess (
            void  )
```

Check and clear interrupt flag used in interrupt mode.

Check interrupt flag, clear flag if event (digital input, high voltage, low voltage detected) occured.

## 6.3 Ae Driver

### 6.3.1 Detailed Description

### Macros

- #define AE_SID_INIT

  *Service ID of Ae_Init.*
- #define AE_SID_WRITEREGISTER

  *Service ID of Ae_WriteRegister.*
- #define AE_SID_READREGISTER

  *Service ID of Ae_ReadRegister.*
- #define AE_SID_GETVERSIONINFO

  *Service ID of Ae_GetVersionInfo.*
- #define AE_SID_ADCMEASURECONFIG

  *Service ID of Ae_AdcMeasureConfig.*
- #define AE_SID_DIGITALINPUTCONFIG

  *Service ID of Ae_DigitalInputEventConfig.*
- #define AE_SID_VOLTAGEMONITORCONFIG

  *Service ID of Ae_VoltageMonitorConfig.*
- #define AE_SID_DECODEGLOBALHEATHSTATUS

  *Service ID of Ae_DecodeGlobalHeathStatus.*
- #define AE_SID_FRAMECOUNTER

  *Service ID of Ae_Frame_Counter.*
- #define AE_SID_FAULTSEVENTSPROCCESS

  *Service ID of Ae_FaultsEventsProccess.*
- #define AE_SID_READFAULTSEVENTSSTATUS

  *Service ID of Ae_ReadFaultsEventsStatus.*
- #define AE_E_INV_CTRL_IDX

  *Development Error ID for "Invalid Controller Index".*
- #define AE_E_UNINIT

  *Development Error ID for "Uninitialized Controller".*
- #define AE_E_PARAM_POINTER

  *Development Error ID for "Invalid (Null) Pointer Parameter".*
- #define AE_E_INV_PARAM

  *Development Error ID for "Invalid Parameter".*
- #define AE_E_INV_MODE

  *Development Error ID for "Invalid Controller Mode".*
- #define AE_E_INIT_FAILED

  *Development Error ID for "Invalid Configuration pointer argument".*

## Function Reference

- void Ae_Init (const Ae_ConfigType *CfgPtr)
  *Initializes the Ae Driver.*
- Std_ReturnType Ae_ReadRegister (uint32 RegAddr, uint8 DataWidth, uint32 *RegValPtr)
  *Read register value.*
- Std_ReturnType Ae_WriteRegister (uint32 RegAddr, uint8 DataWidth, uint32 RegVal)
  *Write register value.*
- uint16 Ae_DecodeGlobalHeathStatus (void)
  *Return status of global heath.*
- boolean Ae_Frame_Counter (boolean IsVerify, uint8 ExpectedFrameCount, uint8 *ActualFrameCount)
  *Verify for correctness the received frame counter or just number of frames.*
- Std_ReturnType Ae_AdcMeasureConfig (const Ae_HviHighVoltageInputConfig *pHviHighVoltConfig, Ae_AnalogInputType AnalogInputSelect)
  *Initializes the configuration for analog input for measurement.*
- Std_ReturnType Ae_DigitalInputEventConfig (const Ae_HviChannelConfig *pAeHviChannelConfig)
  *Initializes the configuration for digital input.*
- Std_ReturnType Ae_VoltageMonitorConfig (const Ae_VmConfig *pAeVmConfig, const Ae_HviHighVoltageInputConfig *pHighVoltageInput)
  *Initializes the configuration for High/Low Voltage detect.*
- void Ae_FaultsEventsProccess (void)
  *Check and clear interrupt flag used in interrupt mode.*
- void Ae_ReadFaultsEventsStatus (void)
  *Check status flag used in polling mode for digital input, high/low voltage detect.*
- void Ae_GetVersionInfo (Std_VersionInfoType *VersionInfoPtr)
  *Returns the version information of this module.*

### 6.3.2 Macro Definition Documentation

#### 6.3.2.1 AE_SID_INIT

```
#define AE_SID_INIT
```

Service ID of Ae_Init.

Definition at line 80 of file Ae.h.

#### 6.3.2.2 AE_SID_WRITEREGISTER

```
#define AE_SID_WRITEREGISTER
```

Service ID of Ae_WriteRegister.

Definition at line 82 of file Ae.h.

#### 6.3.2.3 AE_SID_READREGISTER

```
#define AE_SID_READREGISTER
```

Service ID of Ae_ReadRegister.

Definition at line 84 of file Ae.h.

#### 6.3.2.4 AE_SID_GETVERSIONINFO

```
#define AE_SID_GETVERSIONINFO
```

Service ID of Ae_GetVersionInfo.

Definition at line 86 of file Ae.h.

#### 6.3.2.5 AE_SID_ADCMEASURECONFIG

```
#define AE_SID_ADCMEASURECONFIG
```

Service ID of Ae_AdcMeasureConfig.

Definition at line 88 of file Ae.h.

#### 6.3.2.6 AE_SID_DIGITALINPUTCONFIG

```
#define AE_SID_DIGITALINPUTCONFIG
```

Service ID of Ae_DigitalInputEventConfig.

Definition at line 90 of file Ae.h.

#### 6.3.2.7 AE_SID_VOLTAGEMONITORCONFIG

```
#define AE_SID_VOLTAGEMONITORCONFIG
```

Service ID of Ae_VoltageMonitorConfig.

Definition at line 92 of file Ae.h.

#### 6.3.2.8 AE_SID_DECODEGLOBALHEATHSTATUS

```
#define AE_SID_DECODEGLOBALHEATHSTATUS
```

Service ID of Ae_DecodeGlobalHeathStatus.

Definition at line 94 of file Ae.h.

#### 6.3.2.9 AE_SID_FRAMECOUNTER

```
#define AE_SID_FRAMECOUNTER
```

Service ID of Ae_Frame_Counter.

Definition at line 96 of file Ae.h.

#### 6.3.2.10 AE_SID_FAULTSEVENTSPROCCESS

```
#define AE_SID_FAULTSEVENTSPROCCESS
```

Service ID of Ae_FaultsEventsProccess.

Definition at line 98 of file Ae.h.

#### 6.3.2.11 AE_SID_READFAULTSEVENTSSTATUS

```
#define AE_SID_READFAULTSEVENTSSTATUS
```

Service ID of Ae_ReadFaultsEventsStatus.

Definition at line 100 of file Ae.h.

#### 6.3.2.12 AE_E_INV_CTRL_IDX

```
#define AE_E_INV_CTRL_IDX
```

Development Error ID for "Invalid Controller Index".

Definition at line 106 of file Ae.h.

#### 6.3.2.13 AE_E_UNINIT

```
#define AE_E_UNINIT
```

Development Error ID for "Uninitialized Controller".

Definition at line 108 of file Ae.h.

#### 6.3.2.14 AE_E_PARAM_POINTER

```
#define AE_E_PARAM_POINTER
```

Development Error ID for "Invalid (Null) Pointer Parameter".

Definition at line 110 of file Ae.h.

#### 6.3.2.15 AE_E_INV_PARAM

```
#define AE_E_INV_PARAM
```

Development Error ID for "Invalid Parameter".

Definition at line 112 of file Ae.h.

#### 6.3.2.16 AE_E_INV_MODE

```
#define AE_E_INV_MODE
```

Development Error ID for "Invalid Controller Mode".

Definition at line 114 of file Ae.h.

#### 6.3.2.17 AE_E_INIT_FAILED

```
#define AE_E_INIT_FAILED
```

Development Error ID for "Invalid Configuration pointer argument".

Definition at line 116 of file Ae.h.

### 6.3.3 Function Reference

#### 6.3.3.1 Ae_Init()

```
void Ae_Init (
             const Ae_ConfigType * CfgPtr )
```

Initializes the Ae Driver.

The configuration pointer is internally stored and the driver is initialized.

#### 6.3.3.2 Ae_ReadRegister()

```
Std_ReturnType Ae_ReadRegister (
             uint32 RegAddr,
             uint8 DataWidth,
             uint32 * RegValPtr )
```

Read register value.

Read register value.

Parameters

| | | |
|---|---|---|
| in | *RegAddr* | Index of the AE register to be read. |
| in | *DataWidth* | Size of register in AEC that want to read/write. |
| out | *RegValPtr* | Filled with the register content of the indexed register |

Return E_OK if read value from register successful. E_NOT_OK E_OK if read value from register failure.

#### 6.3.3.3 Ae_WriteRegister()

```
Std_ReturnType Ae_WriteRegister (
             uint32 RegAddr,
             uint8 DataWidth,
             uint32 RegVal )
```

Write register value.

Write register value.

Parameters

| | | |
|---|---|---|
| in | *RegAddr* | Index of the AE register to be write. |
| in | *DataWidth* | Size of register in AEC that want to read/write. |
| in | *RegVal* | Filled with the register content of the indexed register |

Return E_OK if write value from register successful. E_NOT_OK E_OK if write value from register failure.

#### 6.3.3.4 Ae_DecodeGlobalHeathStatus()

```
uint16 Ae_DecodeGlobalHeathStatus (
            void  )
```

Return status of global heath.

#### 6.3.3.5 Ae_Frame_Counter()

```
boolean Ae_Frame_Counter (
            boolean IsVerify,
            uint8 ExpectedFrameCount,
            uint8 * ActualFrameCount )
```

Verify for correctness the received frame counter or just number of frames.

Parameters

| | | |
|---|---|---|
| in | *IsVerify* | want to verify for correctness for frame counter. |
| in | *ExpectedFrameCount* | Number of frame expected get. |
| out | *ActualFrameCount* | Number of frame get in real. |

Return counter status of respond frame from AE_Robin.

#### 6.3.3.6 Ae_AdcMeasureConfig()

```
Std_ReturnType Ae_AdcMeasureConfig (
            const Ae_HviHighVoltageInputConfig * pHviHighVoltConfig,
            Ae_AnalogInputType AnalogInputSelect )
```

Initializes the configuration for analog input for measurement.

Parameters

| | | |
|---|---|---|
| in | *pHviHighVoltConfig* | Pointer for configuration of High Voltage Input. |
| in | *AnalogInputSelect* | Type of Analog Input. |

Return E_OK The initialization was successful.

#### 6.3.3.7 Ae_DigitalInputEventConfig()

```
Std_ReturnType Ae_DigitalInputEventConfig (
            const Ae_HviChannelConfig * pAeHviChannelConfig )
```

Initializes the configuration for digital input.

Parameters

| | | |
|---|---|---|
| in | *pAeHviChannelConfig* | Pointer to configuration of HVI channel. |

Return E_OK The initialization was successful.

#### 6.3.3.8 Ae_VoltageMonitorConfig()

```
Std_ReturnType Ae_VoltageMonitorConfig (
            const Ae_VmConfig * pAeVmConfig,
            const Ae_HviHighVoltageInputConfig * pHighVoltageInput )
```

Initializes the configuration for High/Low Voltage detect.

Parameters

| | | |
|---|---|---|
| in | *pAeVmConfig* | Pointer to configuration of VM. |
| in | *pHighVoltageInput* | Pointer to configuration of High Voltage Input. |

Return E_OK The initialization was successful.

#### 6.3.3.9 Ae_FaultsEventsProccess()

```
void Ae_FaultsEventsProccess (
            void  )
```

Check and clear interrupt flag used in interrupt mode.

Return Interrupt flag want to read raised or not.

#### 6.3.3.10 Ae_ReadFaultsEventsStatus()

```
void Ae_ReadFaultsEventsStatus (
            void  )
```

Check status flag used in polling mode for digital input, high/low voltage detect.

#### 6.3.3.11 Ae_GetVersionInfo()

```
void Ae_GetVersionInfo (
            Std_VersionInfoType * VersionInfoPtr )
```

Returns the version information of this module.

Parameters

| | | |
|---|---|---|
| out | *VersionInfoPtr* | Pointer where to store the version information of this particular module instance.<br>implements Ae_GetVersionInfo_Activity |

***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.