User Manual

for MPC5634M DIO Driver

Document Number: UM14DIOASR3.0R2.0.0
Rev. 1.1

freescale™

Contents

Section Number **Title** **Page**

Chapter 1
Revision History

Chapter 2
Introduction

Chapter 3
Driver

Chapter 4
Tresos Configuration Plug-in

Section Number Title Page

Chapter 1
Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0	07-Feb-2011	Giovanni Di Martino	Document generation
1.1	19-Dec-2011	Robin Gupta	Updated for Monaco RTM 2.0.0

Chapter 2
Introduction

This User Manual describes Freescale Semiconductor AUTOSAR Digital Input Output (DIO) for MPC5634M.

AUTOSAR DIO driver configuration parameters and deviations from the specification are described in DIO Driver chapter of this document. AUTOSAR DIO driver requirements and APIs are described in the AUTOSAR DIO driver software specification document.

2.1 Supported Derivatives

The software described in this document is intented to be used with the following microcontroller devices of Freescale Semiconductor.

Table 2-1. MPC5634M Derivatives

Freescale Semiconductor	mpc5634m_bga208, mpc5634m_qfp144, mpc5634m_qfp176

All of the above microcontroller devices are collectively named as MPC5634M.

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 type: Bold is used for important terms, notes and warnings.

Italic font: Italic typeface is 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.

2.4 Acronyms and Definitions

Table 2-2. Acronyms and Definitions

Term	Definition
API	Application Programming Interface
ASM	Assembler
AUTOSAR	Automotive Open System Architecture
BSMI	Basic Software Make file Interface
C/CPP	C and C++ Source Code
DEM	Diagnostic Event Manager
DET	Development Error Tracer
DIO	Digital Input Output
EcuM	ECU state Manager
N/A	Not Applicable
OS	Operating System
VLE	Variable Length Encoding

2.5 Reference List

Table 2-3. Reference List

#	Title	Version
1	AUTOSAR 3.0DIO Driver Software Specification Document.	V2.2.0 R3.0 Rev 0001
2	MPC5634M Reference Manual	Rev. 6, 4 October 2011

Chapter 3
Driver

3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 3.0DIO Driver Software Specification document (See Table Reference List).

3.2 Driver Design Summary

The DIO Driver provides services for reading and writing to/from:

- DIO Channels (Pins)
- DIO Ports
- DIO Channel Groups

The behaviour of those services is synchronous. This module works on pins and ports which are configured by the PORT driver for this purpose. For this reason, there is no configuration and initialization of this port structure in the DIO Driver.

3.3 Deviation from Requirements

The driver deviates from the AUTOSAR DIO Driver software specification in some places. Table dentifies the AUTOSAR requirements that are not fully implemented, implemented differently, or out of scope for the DIO driver. Table Table 3-1 provides Status column description.

Table 3-1. Deviations Status Column Description

Term	Definition
N/A	Not available
N/T	Not testable
N/S	Out of scope
N/I	Not implemented
N/F	Not fully implemented

Below table identifies the AUTOSAR requirements that are not fully implemented, implemented differently, or out of scope for the driver.

Table 3-2. Driver Deviations Table

Requirement	Status	Description	Notes
DIO067	N/I	The Dio module shall report production errors to the Diagnostic Event Manager.	Production errors are not available in Dio
DIO104	N/I	When reading a port which is smaller than the Dio_PortType using the Dio_ReadPort function (see [DIO103]), the function shall set the bits corresponding to undefined port pins to 0.	For performance reasons the code does not ensure this. Reading a partial port register has an undefined behavior.

3.4 Runtime Errors

This driver doesn’t generate any runtime error.

3.5 Software specification

The following sections contains driver software specifications.

3.5.1 Define Reference

Constants supported by the driver are as per AUTOSAR DIO Driver software specification Version 3.0 .

3.5.1.1 Define DIO_E_PARAM_INVALID_CHANNEL_ID

Invalid channel name requested.

Details: Errors and exceptions that will be detected by the DIO driver.

Table 3-3. Define DIO_E_PARAM_INVALID_CHANNEL_ID Description

Name	DIO_E_PARAM_INVALID_CHANNEL_ID
Initializer	(uint8)0x0AU

3.5.1.2 Define DIO_E_PARAM_INVALID_GROUP_ID

Invalid ChannelGroup id passed.

Details: Errors and exceptions that will be detected by the DIO driver.

Table 3-4. Define DIO_E_PARAM_INVALID_GROUP_ID Description

Name	DIO_E_PARAM_INVALID_GROUP_ID
Initializer	(uint8)0x1FU

3.5.1.3 Define DIO_E_PARAM_INVALID_POINTER

API service called with NULL pointer.

Details: Errors and exceptions that will be detected by the DIO driver.

Table 3-5. Define DIO_E_PARAM_INVALID_POINTER Description

Name	DIO_E_PARAM_INVALID_POINTER
Initializer	(uint8)0x20U

3.5.1.4 Define DIO_E_PARAM_INVALID_PORT_ID

Invalid port name requested.

Details: Errors and exceptions that will be detected by the DIO driver.

Table 3-6. Define DIO_E_PARAM_INVALID_PORT_ID Description

Name	DIO_E_PARAM_INVALID_PORT_ID
Initializer	(uint8)0x14U

3.5.1.5 Define DIO_GETVERSIONINFO_ID

API service ID for DIO Get Version Info function.

Details: Parameters used when raising an error/exception.

Table 3-7. Define DIO_GETVERSIONINFO_ID Description

Name	DIO_GETVERSIONINFO_ID
Initializer	(uint8)0x12U

3.5.1.6 Define DIO_MASKEDWRITEPORT_ID

API service ID for DIO Masked Write Port function.

Details: Parameters used when raising an error/exception.

Table 3-8. Define DIO_MASKEDWRITEPORT_ID Description

Name	DIO_MASKEDWRITEPORT_ID
Initializer	(uint8)0x06U

3.5.1.7 Define DIO_READCHANNEL_ID

API service ID for DIO Read Channel function.

Details: Parameters used when raising an error/exception.

Table 3-9. Define DIO_READCHANNEL_ID Description

Name	DIO_READCHANNEL_ID
Initializer	(uint8)0x00U

3.5.1.8 Define DIO_READCHANNELGROUP_ID

API service ID for DIO Read Channel Group function.

Details: Parameters used when raising an error/exception.

Table 3-10. Define DIO_READCHANNELGROUP_ID Description

Name	DIO_READCHANNELGROUP_ID
Initializer	(uint8)0x04U

3.5.1.9 Define DIO_READPORT_ID

API service ID for DIO Read Port function.

Details: Parameters used when raising an error/exception.

Table 3-11. Define DIO_READPORT_ID Description

Name	DIO_READPORT_ID
Initializer	(uint8)0x02U

3.5.1.10 Define DIO_WRITECHANNEL_ID

API service ID for DIO Write Channel function.

Details: Parameters used when raising an error/exception.

Table 3-12. Define DIO_WRITECHANNEL_ID Description

Name	DIO_WRITECHANNEL_ID
Initializer	(uint8)0x01U

3.5.1.11 Define DIO_WRITECHANNELGROUP_ID

API service ID for DIO Write Channel Group function.

Details: Parameters used when raising an error/exception.

Table 3-13. Define DIO_WRITECHANNELGROUP_ID Description

Name	DIO_WRITECHANNELGROUP_ID
Initializer	(uint8)0x05U

3.5.1.12 Define DIO_WRITEPORT_ID

API service ID for DIO Write Port function.

Details: Parameters used when raising an error/exception.

Table 3-14. Define DIO_WRITEPORT_ID Description

Name	DIO_WRITEPORT_ID
Initializer	(uint8)0x03U

3.5.2 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR DIO Driver software specification Version 3.0 .

3.5.3 Function Reference

Functions of all functions supported by the driver are as per AUTOSAR DIO Driver software specification Version 3.0 .

3.5.3.1 Function Dio_GetVersionInfo

Service to get the version information of this module.

Details:

The `Dio_GetVersionInfo()` function shall return the version information of this module. The version information includes:

- Module Id,
- Vendor Id,
- Vendor specific version numbers.

Pre: This function can be used only if has been set in the plugin. .

Satisfied Requirements: DIO139, DIO123, DIO124, DIO126.

Prototype: `void Dio_GetVersionInfo(Std_VersionInfoType *versioninfo);`

Table 3-15. Dio_GetVersionInfo Arguments

Type	Name	Direction	Description
Std_VersionInfoType *	versioninfo	**input, output**	Pointer to where to store the version information of this module.

3.5.3.2 Function Dio_ReadChannel

Returns the value of the specified DIO channel.

Details:

The `Dio_ReadChannel()` function will return the value of the specified DIO channel.

Satisfied Requirements: DIO027, DIO058, DIO083, DIO084, DIO005, DIO085, DIO118, DIO011, DIO012, DIO133.

Prototype: `Dio_LevelType Dio_ReadChannel(const Dio_ChannelType ChannelId);`

Table 3-16. Dio_ReadChannel Arguments

Type	Name	Direction	Description
const Dio_ChannelType	ChannelId	**input**	Specifies the required channel id.

Return : Returns the level of the corresponding pin as STD_HIGH or STD_LOW.

Table 3-17. Dio_ReadChannel Return Values

Name	Description
STD_HIGH	The logical level of the corresponding pin is 1.
STD_LOW	The logical level of the corresponding Pin is 0.

3.5.3.3 Function Dio_ReadChannelGroup

This service reads a subset of the adjoining bits of a port.

Details:

The `Dio_ReadChannelGroup()` function will read a subset of the adjoining bits of a port (channel group). The `Dio_WriteChannelGroup()` function will do the masking of the channel and will do the shifting so that the values written by the function are aligned to the LSB.

Satisfied Requirements: DIO037, DIO058, DIO092, DIO093, DIO056, DIO084, DIO118, DIO005, DIO012,DIO014, DIO137.

Prototype: `Dio_PortLevelType Dio_ReadChannelGroup(const Dio_ChannelGroupType *ChannelGroupIdPtr);`

Table 3-18. Dio_ReadChannelGroup Arguments

Type	Name	Direction	Description
const Dio_ChannelGroupType *	ChannelGroupIdPtr	**input**	Pointer to the channel group.

3.5.3.4 Function Dio_ReadPort

Returns the level of all channels of specified port.

Details:

The `Dio_ReadPort()` function will return the level of all channels of specified port.

Satisfied Requirements: DIO031, DIO058, DIO104, DIO118, DIO005, DIO012, DIO013 DIO135.

Prototype: `Dio_PortLevelType Dio_ReadPort(const Dio_PortType PortId);`

Table 3-19. Dio_ReadPort Arguments

Type	Name	Direction	Description
const Dio_PortType	PortId	**input**	Specifies the required port id.

3.5.3.5 Function Dio_WriteChannel

Service to set a level of a channel.

Details:

If the specified channel is configured as an output channel, the `Dio_WriteChannel()` function shall set the specified Level for the specified channel. If the specified channel is configured as an input channel, the `Dio_WriteChannel()` function shall have no influence on the physical output and on the result of the next Read-Service.

Satisfied Requirements: DIO028, DIO029, DIO057, DIO079, DIO119, DIO005, DIO006, DIO064, DIO070, DIO134.

Prototype: `void Dio_WriteChannel(const Dio_ChannelType ChannelId, const Dio_LevelType Level);`

Table 3-20. Dio_WriteChannel Arguments

Type	Name	Direction	Description
const Dio_ChannelType	ChannelId	**input**	Specifies the required channel id.
const `Dio_LevelType`	Level	**input**	Specifies the channel desired level.

3.5.3.6 Function Dio_WriteChannelGroup

Service to set a subset of the adjoining bits of a port to a specified level.

Details:

The `Dio_WriteChannelGroup()` function will set a subset of the adjoining bits of a port (channel group) to a specified level without changing the remaining channels of the port and channels which are configured as input. The `Dio_WriteChannelGroup()` function will do the masking of the channel and will do the shifting so that the values written by the function are aligned to the LSB.

Satisfied Requirements: DIO039, DIO040, DIO057, DIO090, DIO091, DIO056, DIO119, DIO005, DIO008, DIO064, DIO070, DIO138.

Prototype: `void Dio_WriteChannelGroup(const Dio_ChannelGroupType *ChannelGroupIdPtr, const Dio_PortLevelType Level);`

Table 3-21. Dio_WriteChannelGroup Arguments

Type	Name	Direction	Description
const Dio_ChannelGroupType *	ChannelGroupIdPtr	**input**	Pointer to the channel group.
const Dio_PortLevelType	Level	**input**	Desired level for the channel group.

3.5.3.7 Function Dio_WritePort

Service to set a value of the port.

Details:

The `Dio_WritePort()` function will set the specified value for the specified port.

Satisfied Requirements: DIO034, DIO035, DIO057, DIO105, DIO108, DIO119, DIO005, DIO004, DIO005, DIO007, DIO064, DIO070, DIO136.

Prototype: `void Dio_WritePort(const Dio_PortType PortId, const Dio_PortLevelType Level);`

Table 3-22. Dio_WritePort Arguments

Type	Name	Direction	Description
const Dio_PortType	PortId	**input**	Specifies the required port id.
const Dio_PortLevelType	Level	**input**	Specifies the required level for the port pin.

3.5.4 Structs Reference

Data structures supported by the driver are as per AUTOSAR DIO Driver software specification Version 3.0 .

3.5.5 Types Reference

Types supported by the driver are as per AUTOSAR DIO Driver software specification Version 3.0 .

3.5.6 Variables Reference

Variables supported by the driver are as per AUTOSAR DIO Driver software specification Version 3.0 .

3.6 Symbolic Names DISCLAIMER

All containers having the symbolic name tag set as true in the Autosar schema will generate defines like:

#define <Container_Short_Name> <Container_ID>

For this reason it is forbidden to duplicate the name of such containers across the MCAL configuration, 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 DIO Driver. The most of the parameters are described below.

4.1 Configuration elements of Dio

Included forms :

- IMPLEMENTATION_CONFIG_VARIANT
- CommonPublishedInformation
- DioGeneral
- DioPort

Table 4-1. Revision table

Revision	Date
2.0.0	2011-12-06T17:00:00

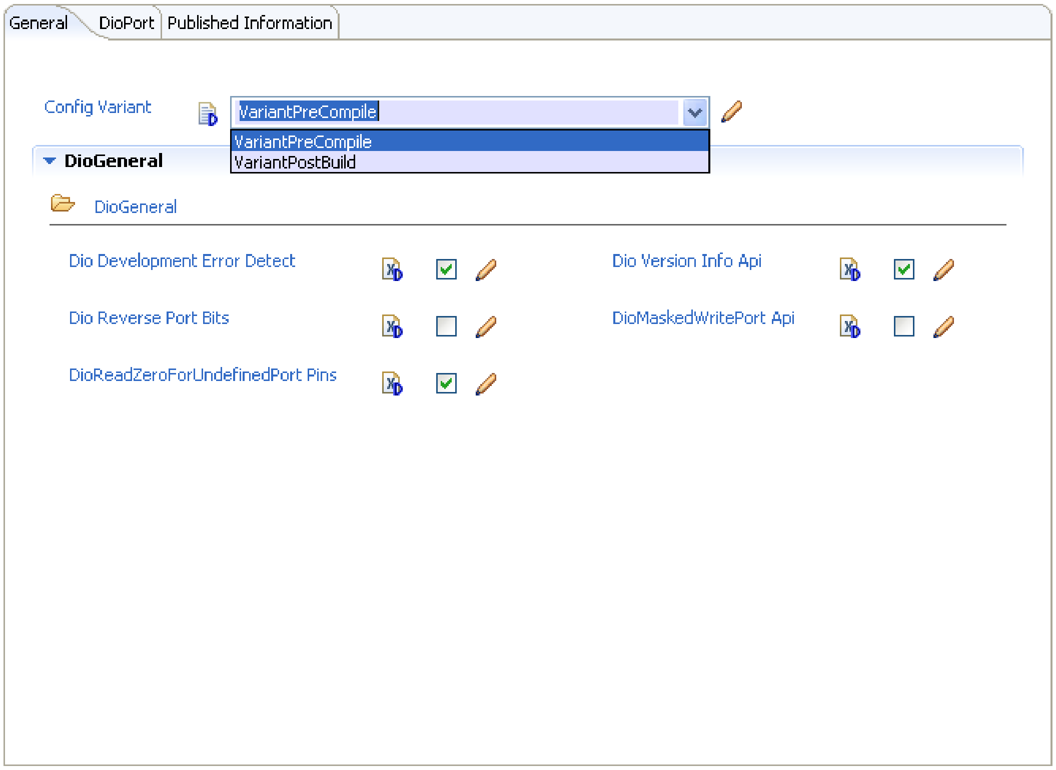
4.2 Form IMPLEMENTATION_CONFIG_VARIANT

VariantPreCompile: Only precompile time configuration parameters.

VariantPostBuild : Mix of precompile and postbuild time configuration parameters.

If Config Variant = VariantPreCompile, the files Dio_Cfg.h and Dio_Cfg.c should be used.

If Config Variant = VariantPostBuild, the files Dio_Cfg.h and Dio_PBcfg.c should be used.



Config Variant: VariantPreCompile

DioGeneral

Dio Development Error Detect ☑

Dio Version Info Api ☑

Dio Reverse Port Bits ☐

DioMaskedWritePort Api ☐

DioReadZeroForUndefinedPort Pins ☑

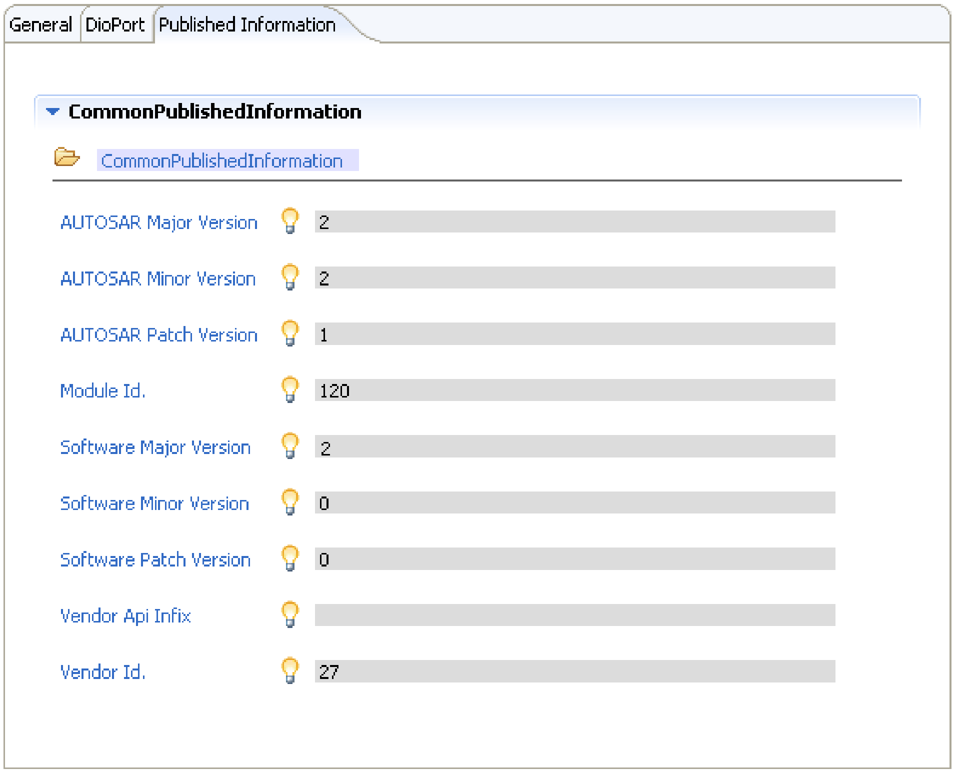
Figure 4-1. Tresos Plugin snapshot for IMPLEMENTATION_CONFIG_VARIANT form.

Table 4-2. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description

Property	Value
Label	Config Variant
Default	VariantPreCompile
Range	VariantPreCompile VariantPostBuild

4.3 Form CommonPublishedInformation

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



AUTOSAR Major Version	2
AUTOSAR Minor Version	2
AUTOSAR Patch Version	1
Module Id.	120
Software Major Version	2
Software Minor Version	0
Software Patch Version	0
Vendor Api Infix	
Vendor Id.	27

Figure 4-2. Tresos Plugin snapshot for CommonPublishedInformation form.

4.3.1 ArMajorVersion (CommonPublishedInformation)

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

Table 4-3. Attribute ArMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Major Version
Origin	Custom
Symbolic Name	false
Default	2
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=2 <=2

4.3.2 ArMinorVersion (CommonPublishedInformation)

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

Table 4-4. Attribute ArMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Minor Version
Origin	Custom
Symbolic Name	false
Default	2
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=2 <=2

4.3.3 ArPatchVersion (CommonPublishedInformation)

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

Table 4-5. Attribute ArPatchVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Patch Version
Origin	Custom
Symbolic Name	false
Default	1
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=1 <=1

4.3.4 ModuleId (CommonPublishedInformation)

Module ID of this module from Module List.

Table 4-6. Attribute ModuleId (CommonPublishedInformation) detailed description

Property	Value
Label	Module Id.
Origin	Custom
Symbolic Name	false
Default	120
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=120 <=120

4.3.5 SwMajorVersion (CommonPublishedInformation)

Major version number of the vendor specific implementation of the module.

Table 4-7. Attribute SwMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Major Version
Origin	Custom
Symbolic Name	false
Default	2
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=2 <=2

4.3.6 SwMinorVersion (CommonPublishedInformation)

Minor version number of the vendor specific implementation of the module.

Table 4-8. Attribute SwMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Minor Version

Table continues on the next page...

Table 4-8. Attribute SwMinorVersion (CommonPublishedInformation) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=0

4.3.7 SwPatchVersion (CommonPublishedInformation)

Patch version number of the vendor specific implementation of the module.

Table 4-9. Attribute SwPatchVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Patch Version
Origin	Custom
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=0

4.3.8 VendorApiInfix (CommonPublishedInformation)

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" 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.

Table 4-10. Attribute VendorApiInfix (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Api Infix
Origin	AUTOSAR_ECUC V1.0.0
Symbolic Name	false
Default	
Enable	false

4.3.9 VendorId (CommonPublishedInformation)

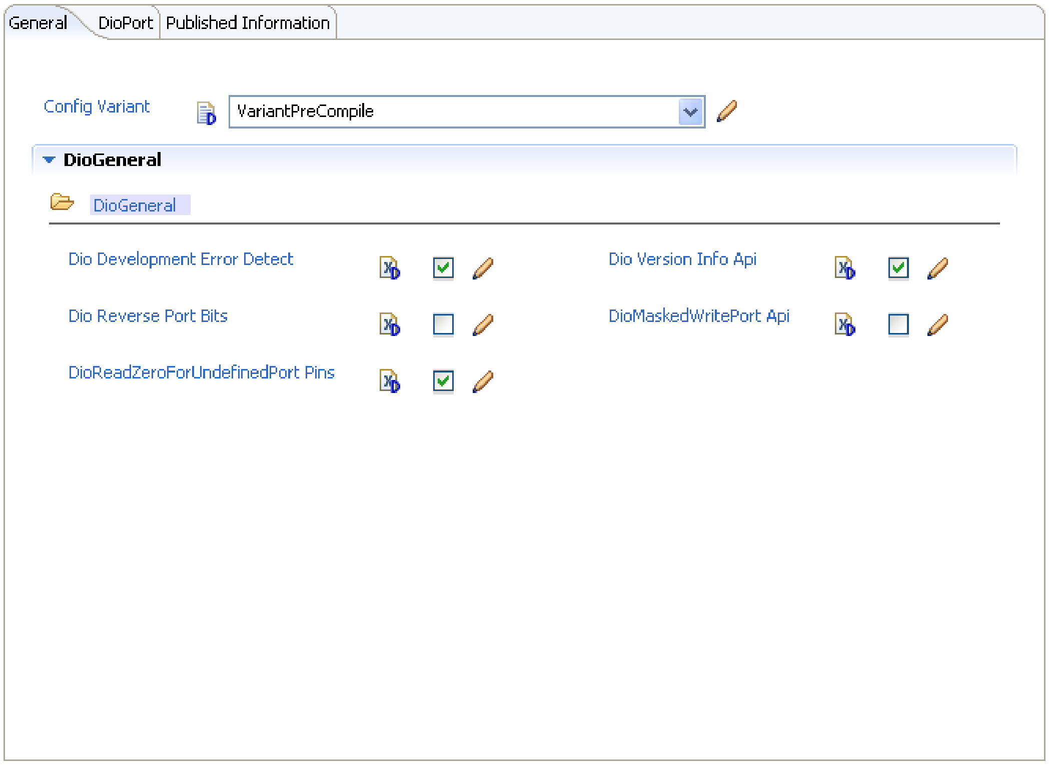
Vendor ID of the dedicated implementation of this module.

Table 4-11. Attribute VendorId (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Id.
Origin	Custom
Symbolic Name	false
Default	27
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=27 <=27

4.4 Form DioGeneral

General DIO module configuration parameters.



Config Variant: VariantPreCompile

DioGeneral

DioGeneral

Dio Development Error Detect ☑

Dio Version Info Api ☑

Dio Reverse Port Bits □

DioMaskedWritePort Api □

DioReadZeroForUndefinedPort Pins ☑

Figure 4-3. Tresos Plugin snapshot for DioGeneral form.

4.4.1 DioDevErrorDetect (DioGeneral)

Switches the Development Error Detection and Notification ON or OFF.

Table 4-12. Attribute DioDevErrorDetect (DioGeneral) detailed description

Property	Value
Label	Dio Development Error Detect
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.4.2 DioVersionInfoApi (DioGeneral)

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

Table 4-13. Attribute DioVersionInfoApi (DioGeneral) detailed description

Property	Value
Label	Dio Version Info Api
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.4.3 DioReversePortBits (DioGeneral)

If this box is checked, the bits written to defined ports will be reversed.

Writing 3 to PORTA with checkbox disabled will set pins 14 and 15

Writing 3 to PORTA with checkbox enabled will set pins 0 and 1

Table 4-14. Attribute DioReversePortBits (DioGeneral) detailed description

Property	Value
Label	Dio Reverse Port Bits
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1
Enable	false

4.4.4 DioMaskedWritePortApi (DioGeneral)

Defines whether the driver function Dio_MaskedWritePort() will be included at compile time or excluded.

True - Dio_MaskedWritePort() API enabled.

False - Dio_MaskedWritePort() API disabled.

Table 4-15. Attribute DioMaskedWritePortApi (DioGeneral) detailed description

Property	Value
Label	DioMaskedWritePort Api
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1
Enable	false

4.4.5 DioReadZeroForUndefinedPortPins (DioGeneral)

Defines whether the Dio_ReadPort() function includes the capability to read the undefined port pins as 0.

True - Enables the Dio_ReadPort() functionality to read the undefined port pins as 0

False - Disables the Dio_ReadPort() functionality to read the undefined port pins as 0

(Supports the normal functionality with Dio_ReadPort())

Table 4-16. Attribute DioReadZeroForUndefinedPortPins (DioGeneral) detailed description

Property	Value
Label	DioReadZeroForUndefinedPort Pins
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1
Enable	false

4.5 Form DioPort

Configuration of individual DIO ports, consisting of channels and possible channel groups.

The single DIO channel levels inside a DIO port represent a bit in the DIO port value. A channel group is a formal logical combination of several adjoining DIO channels within a DIO port. The configuration process for Dio module shall provide symbolic names for each configured DIO channel, port and group.

Included forms :

- DioChannel
- DioChannelGroup

Name: CanFilterMask_0

General

CanFilterMaskValue: 2047

Figure 4-4. Tresos Plugin snapshot for DioPort form.

4.5.1 DioPortId (DioPort)

Numeric identifier of the DIO port. Symbolic names will be generated for each port pin id for the pins which being used for configuration.

NOTE: Use the following values to configure different ports.

- PORT0=0 - pins from 0 to 15
- PORT1=1 - pins from 16 to 31
- PORT2=2 - pins from 32 to 47
- PORT3=3 - pins from 48 to 63
- PORT4=4 - pins from 64 to 79
- PORT5=5 - pins from 80 to 95
- PORT6=6 - pins from 96 to 111
- PORT7=7 - pins from 112 to 127
- PORT8=8 - pins from 128 to 143
- PORT9=9 - pins from 144 to 159
- PORT10=10 - pins from 160 to 175
- PORT11=11 - pins from 176 to 191
- PORT12=12 - pins from 192 to 207

- PORT13=13 - pins from 208 to 223
- PORT14=14 - pins from 224 to 239

The formula for calculating PORT number for PINx (PCR[x]) is:

PORTx=integer(PINx/16) e.g. for PCR[100] => PORT100=integer(100/16)=6

Table 4-17. Attribute DioPortId (DioPort) detailed description

Property	Value
Label	DioPort Id
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	true
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <=14 >=0

4.5.2 Form DioChannel

Configuration of an individual DIO channel. Symbolic names will be generated for each channel.

A general purpose digital IO pin represents a DIO channel, which will be having value either STD_HIGH or STD_LOW.

Is included by form : DioPort

Name CanFilterMask_0

CanFilterMaskValue 2047

Figure 4-5. Tresos Plugin snapshot for DioChannel form.

4.5.2.1 DioChannelId (DioChannel)

Channel Id of the DIO channel. This value will be assigned to the symbolic names.

There is 16 pin in each port, logical value of each pin is the bit value of port. Channel id should be given with port id. Eg, if port id is given as DioPort_0, then channel should be DioChannel_01.

The formula for calculating channel number for PINx (PCR[x]) is:

CHANNELx=PINx % 16 e.g. for PCR[100] => CHANNEL100= 100 % 16 = 4

Table 4-18. Attribute DioChannelId (DioChannel) detailed description

Property	Value
Label	DioChannel Id
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	true
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <=15 >=0

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor 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. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.

AUTOSAR

Document Number: UM14DIOASR3.0R2.0.0
Rev. 1.1

freescale™