DIO Drivers

For

AVR Microcontrollers

LOW LEVEL DESGIN Document

1. API specification

1.1 Type definitions

Name:	STD_TYPE_DIO
Type:	Enum
Element:	E_OK, E_NOT_OK
Description:	This help to make sure that work done correctly

Name:	DIO_LEVEL_TYPE
Type:	Enum
Element:	LOGIC_LOW, LOGIC_HIGH
Description:	This help to know the Level of specific pin

Name:	DIO_PIN_DIRECTION_TYPE
Type:	Enum
Element:	PIN_INPUT, PIN_OUTPUT, PIN_INFREE
Description:	This help to know the direction of pin

Name:	DIO_PORT_DIRECTION_TYPE
Type:	Enum
Element:	PORT_INPUT, PORT_OUTPUT,
Description:	This help to know the direction of port

Name:	DIO_PORT_ID
Type:	Enum
Element:	PORTA_ID, // Port A. PORTB_ID, // Port B. PORTC_ID, // Port C. PORTD_ID, // Port D. TOTALPORTS_ID,
Description:	This help to know ID of every port

Name:	DIO_PIN_ID
Type:	Enum
Element:	PIN0_ID, PIN1_ID, PIN2_ID, PIN3_ID, PIN4_ID, PIN5_ID, PIN6_ID, PIN7_ID, TOTALPINS_ID
Description:	This help to know ID of every Pin

Name:	Dio_ConfigType
Type:	Struct
Element:	DIO_PIN_ID PIN. DIO_PORT_ID PORT. DIO_PORT_DIRECTION_TYPE PORT_DIRECTION; DIO_PIN_DIRECTION_TYPE PIN_DIRECTION;
Description:	Structure to represent a DIO configuration.

2. Function definitions

Service name:	MCAL_DIO_INIT ();
Syntax:	void MCAL_DIO_INIT();
Parameters (in):	NONE
Parameters (in/out):	NONE
Parameters (out):	NONE
Return value:	NONE
Description:	The function make the initialization for DIO (
Requirement:	[DIO_001]

Service name:	**MCAL_DIO_INIT_PINS**
Syntax:	void MCAL_DIO_INIT_PINS (Dio_ConfigType * configType , DIO_PIN_DIRECTION_TYPE pinStatus)
Parameters (in):	Dio_ConfigType * configType, DIO_PIN_DIRECTION_TYPE pin Status
Parameters (in/out):	NONE
Parameters (out):	NONE
Return value:	NONE
Description:	Function to Initialize the Dio module.
Requirement:	[DIO_001] [DIO_005]

Service name:	**MCAL_DIO_READ_PORT_DIRECTION**
Syntax:	MCAL_DIO_READ_PORT_DIRECTION(DIO_PORT_ID PortId);
Parameters (in):	DIO_PORT_ID - port id
Parameters (in/out):	NONE
Parameters (out):	NONE
Return value:	DIO_LEVEL_TYPE
Description:	Function to Read and return the value of the required port.
Requirement:	[DIO_003] [DIO_006]

Service name:	**MCAL_DIO_READ_PIN_DIRECTION**
Syntax:	MCAL_DIO_READ_PIN_DIRECTION(DIO_PORT_ID PortId,DIO_PIN_ID PinId);
Parameters (in):	DIO_PORT_ID - port id, DIO_PIN_ID - pin id
Parameters (in/out):	NONE
Parameters (out):	NONE
Return value:	DIO_LEVEL_TYPE
Description:	Function to Read and return the value for the required pin, it should be Logic High or Logic Low.
Requirement:	[DIO_003] [DIO_005]

Service name:	**MCAL_DIO_Std_FLIP_PORT_DIRECTION**
Syntax:	MCAL_DIO_Std_FLIP_PORT_DIRECTION(DIO_PORT_ID PortId);
Parameters (in):	DIO_PORT_ID - port id
Parameters (in/out):	NONE
Parameters (out):	NONE
Return value:	STD_TYPE_DIO
Description:	Function to toggle port
Requirement:	[DIO_003]

Service name:	**MCAL_DIO_Std_FLIP_PIN_DIRECTION**
Syntax:	MCAL_DIO_Std_FLIP_PIN_DIRECTION(DIO_PORT_ID PortId, DIO_PIN_ID PIN);
Parameters (in):	DIO_PORT_ID - port id DIO_PIN_ID - pin id
Parameters (in/out):	NONE
Parameters (out):	NONE
Return value:	STD_TYPE_DIO
Description:	Function to toggle specific pin
Requirement:	[DIO_003]

Service name:	**MCAL_DIO_Std_WRITE_PORT**
Syntax:	MCAL_DIO_Std_WRITE_PORT (Dio_ConfigType * Config_type , uint8 data)
Parameters (in):	Dio_ConfigType - Pointer to post-build configuration data , Data
Parameters (in/out):	NONE
Parameters (out):	NONE
Return value:	STD_TYPE_DIO
Description:	Function to write data on specific port

Requirement:	[DIO_003] [DIO_007]

Service name:	**MCAL_DIO_Std_WRITE_PIN**
Syntax:	MCAL_DIO_Std_WRITE_PIN (Dio_ConfigType * Config_type , DIO_LEVEL_TYPE VOLT)
Parameters (in):	Dio_ConfigType - Pointer to post-build configuration data ,VOLT
Parameters (in/out):	NONE
Parameters (out):	NONE
Return value:	STD_TYPE_DIO
Description:	Function to write data on specific pin
Requirement:	[DIO_003] [DIO_007]