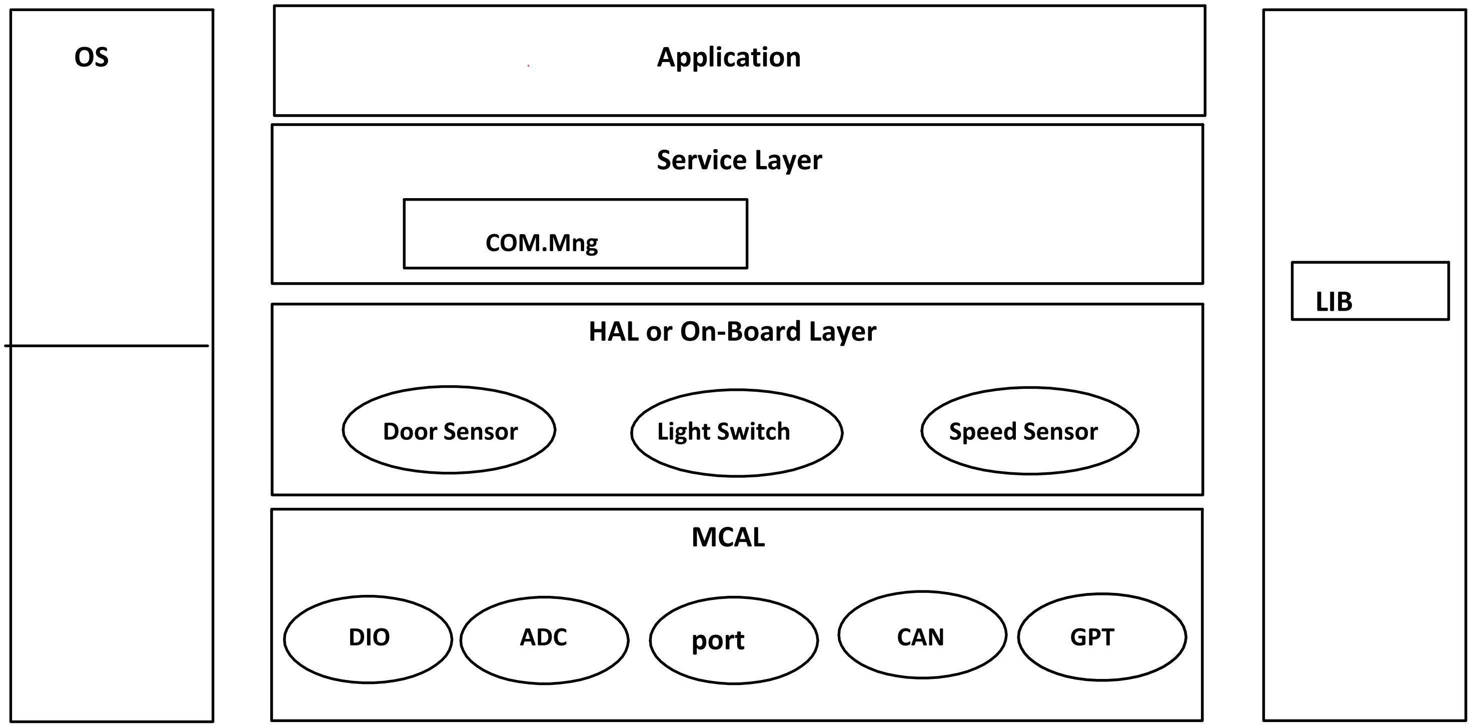
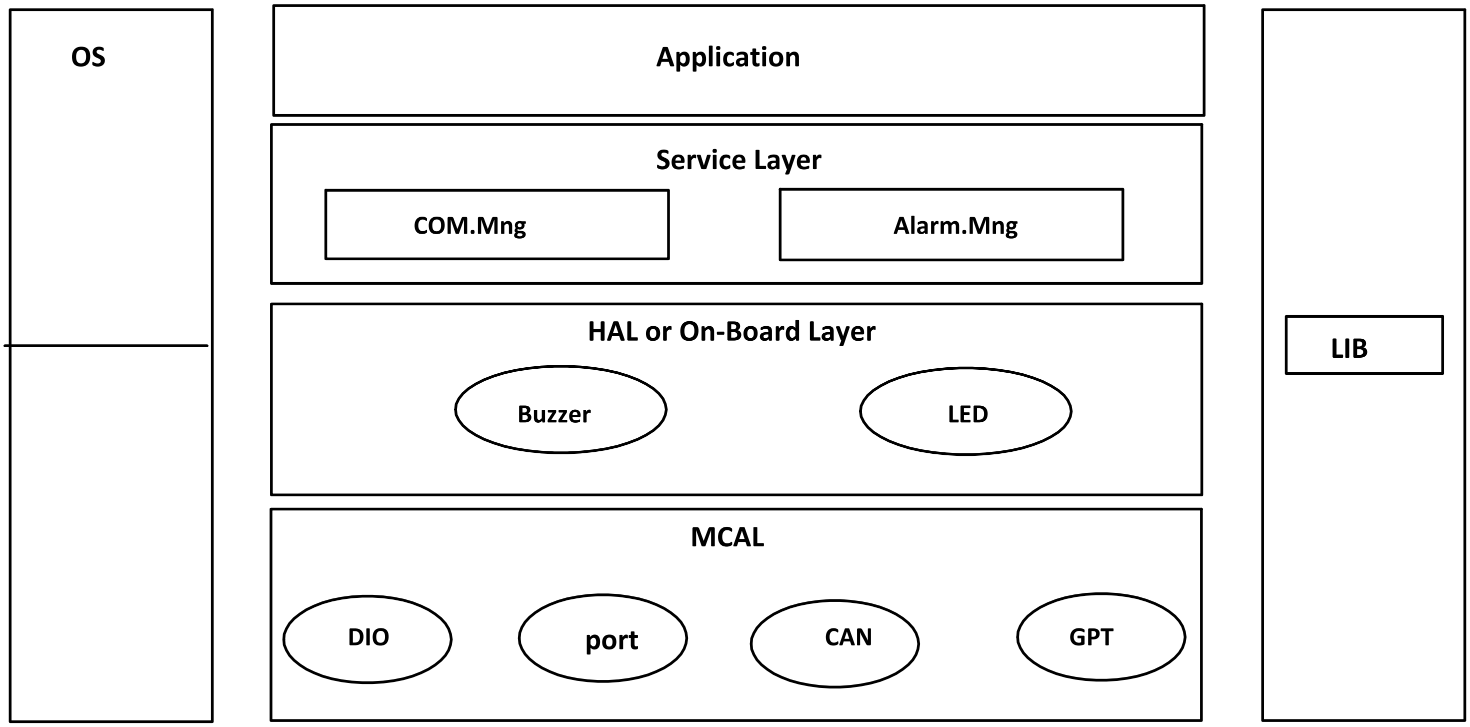
Fully Static Design

Saturday, October 1, 2022 9:03 AM

ECU 1



ECU 2



Module APIs (According to AUTOSAR Docs)

- DIO Driver
  - Function definitions
  - Type definitions

Service name:	Dio_ReadChannel
Syntax:	Dio_LevelType Dio_ReadChannel(Dio_ChannelType ChannelId)
Service ID[hex]:	0x00
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	ChannelId ID of DIO channel
Parameters (out):	None
Return value:	Dio_LevelType STD_HIGH The physical level of the corresponding Pin is STD_HIGH STD_LOW The physical level of the corresponding Pin is STD_LOW
Description:	Returns the value of the specified DIO channel.

Service name:	Dio_WriteChannel
Syntax:	void Dio_WriteChannel(Dio_ChannelType ChannelId, Dio_LevelType Level)
Service ID[hex]:	0x01
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	ChannelId ID of DIO channel Level Value to be written
Parameters (out):	None
Return value:	None
Description:	Service to set a level of a channel.

Name:	Dio_ChannelType	
Type:	uint	
Range:	This is implementation specific but not all values may be valid within the type.	Shall cover all available DIO channels
Description:	Numeric ID of a DIO channel	

Name:	Dio_LevelType		
Type:	uint8		
Range:	STD_LOW STD_HIGH	0x00 0x01	Physical state 0V Physical state 5V or 3.3V
Description:	These are the possible levels a DIO channel can have (input or output)		

- ADC Driver
  - Function definitions
  - Type definitions

Service name:	Adc_Init
Syntax:	void Adc_Init (const Adc_ConfigType *ConfigPtr)
Service ID[hex]:	0x00
Sync/Async:	Synchronous
Reentrancy:	Non re-entrant
Parameters (in):	ConfigPtr Pointer to configuration set
Parameters (out):	None
Return value:	None

Description:	ADC054: Service for initialising the hardware ADC module according to a configuration set referenced by ConfigPtr.

Service name:	Adc_SingleValueReadChannel
Syntax:	Adc_ValueType Adc_SingleValueReadChannel (Adc_ChannelType channel)
Service ID[hex]:	0x04
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	channel Selected numeric channel ID
Parameters (out):	None
Return value:	Adc_ValueType Conversion result
Description:	ADC075: Service for reading the last valid conversion result of the requested channel. ADC113: This service shall return the raw converted value without further

Name:	Adc_ConfigType
Type:	structure
Range:	Implementation specific configuration data structure. See chapter 10 for configurable parameters.
Description:	Type of the data structure including the configuration set required for initialising the ADC unit

- PORT Driver
    - Function definitions
    - Type definitions

Service name:	Port_Init
Syntax:	void Port_Init(const Port_ConfigType* ConfigPtr)
Service ID[hex]:	0x00
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	ConfigPtr Pointer to configuration set
Parameters (out):	None
Return value:	None
Description:	Initializes the Port Driver module.

Service name:	Port_SetPinMode
Syntax:	void Port_SetPinMode(Port_PinType Pin, Port_PinModeType Mode)
Service ID[hex]:	0x04
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	Pin Port Pin ID number Mode New Port Pin mode to be set on port pin
Parameters (out):	None
Return value:	None
Description:	Sets the port pin mode.

Service name:	Port_SetPinDirection
Syntax:	void Port_SetPinDirection(Port_PinType Pin, Port_PinDirectionType Direction)
Service ID[hex]:	0x01
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	Pin Port Pin ID number Direction Port Pin Direction
Parameters (out):	None

Return value:	None
Description:	Sets the port pin direction

Name:	Port_ConfigType
Type:	structure
Range:	Hardware Dependent Structure The contents of the initialization data structure are specific to the MCU
Description:	Type of the external data structure containing the initialization data for this module

- CAN Driver
    - Function definitions
    - Type definitions

Service name:	Can_Init
Syntax:	void Can_Init(const Can_ConfigType* Config)
Service ID[hex]:	0x00
Sync/Async:	Synchronous
Reentrancy:	Non re-entrant
Parameters (in):	Config Pointer to driver configuration.
Parameters (out):	None
Return value:	None
Description:	This function initializes the module.

Service name:	Can_SetBaudrate
Syntax:	Std_ReturnType Can_SetBaudrate(uint8 Controller, uint16 BaudRateConfigID)
Service ID[hex]:	0x0f
Sync/Async:	Synchronous
Reentrancy:	Reentrant for different Controllers. Non reentrant for the same Controller.
Parameters (in):	Controller CAN controller, whose baud rate shall be set BaudRateConfigID references a baud rate configuration by ID (see CanControllerBaudRateConfigID)
Parameters (out):	None
Return value:	None
Description:	This function initializes the module.

Service name:	Can_Write
Syntax:	Std_ReturnType Can_Write(Can_HwHandleType Hth, const Can_PduType* PduInfo)
Service ID[hex]:	0x06
Sync/Async:	Synchronous
Reentrancy:	Reentrant (thread-safe)
Parameters (in):	Hth information which HW-transmit handle shall be used for transmit. Implicitly this is also the information about the controller to use because the Hth numbers are unique inside one hardware unit.
Parameters (out):	PduInfo Pointer to SDU user memory, Data Length and Identifier
Return value:	Std_ReturnType E_OK: Write command has been accepted E_NOT_OK: development error occurred CAN_BUSY: No TX hardware buffer available or pre-emptive call of Can_Write that can 't be implemented re-entrant
Description:	This function is called by CanIf to pass a CAN message to CanDrv for transmission.

Name:	Can_ConfigType
Type:	structure
Range:	Implementation specific

Description:	This is the type of the external data structure containing the overall initialization data for the CAN driver and SFR settings affecting all controllers. Furthermore it contains pointers to controller configuration structures. The contents of the initialization data structure are CAN hardware specific.

Name:	Can_PduType
Type:	structure
Range:	Implementation specific
Description:	This type unites PduId (swPduHandle), SduLength (length), SduData (sdu), and CanId (id) for any CAN L-SDU

Name:	Can_HwHandleType
Type:	Uint8 , uint16
Range:	Standard -- 0..0x0FF Extended -- 0..0xFFFF
Description:	Represents the hardware object handles of a CAN hardware unit. For CAN hardware units with more than 255 HW objects use extended range.

- GPT Driver
  - Function definitions
  - Type definitions

Service name:	Gpt_Init
Syntax:	void Gpt_Init(const Gpt_ConfigType* ConfigPtr)
Service ID[hex]:	0x01
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	ConfigPtr Pointer to a selected configuration structure
Parameters (out):	PduInfo Pointer to SDU user memory, Data Length and Identifier
Return value:	None
Description:	Initializes the GPT driver

Service name:	Gpt_StartTimer
Syntax:	void Gpt_StartTimer(Gpt_ChannelType Channel, Gpt_ValueType Value)
Service ID[hex]:	0x05
Sync/Async:	Synchronous
Reentrancy:	Reentrant (but not for the same timer channel)
Parameters (in):	Channel Numeric identifier of the GPT channel. Value Target time in number of ticks.
Parameters (out):	PduInfo Pointer to SDU user memory, Data Length and Identifier
Return value:	None
Description:	Starts a timer channel

Service name:	Gpt_StopTimer
Syntax:	void Gpt_StopTimer(Gpt_ChannelType Channel)
Service ID[hex]:	0x06
Sync/Async:	Synchronous
Reentrancy:	Reentrant (but not for the same timer channel)
Parameters (in):	Channel Numeric identifier of the GPT channel.
Parameters (out):	None
Return value:	None
Description:	Stops a timer channel

- DOOR Driver

    - Function definitions
    - Type definitions

Service name:	DoorSensor_Init
Syntax:	void DoorSensor_Init( )
Sync/Async:	Synchronous
Reentrancy:	Non-Reentrant
Parameters (in):	
Parameters (out):	None
Return value:	None
Description:	Initialize the door

Service name:	DoorSensor_GetStatus
Syntax:	void DoorSensor_GetStatus()
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	
Parameters (out):	None
Return value:	E_OK 1 E_NOK 0
Description:	Reads the door sensor status periodically

- Speed Driver
    - Function definitions
    - Type definitions

Service name:	SpeedSensor_Init
Syntax:	void SpeedSensor_Init()
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	None
Parameters (out):	None
Return value:	None
Description:	Initialize the speed sensor

Service name:	SpeedSensor_ReadStatus
Syntax:	void SpeedSensor_ReadStatus()
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	None
Parameters (out):	None
Return value:	E_OK 1 E_NOK 0
Description:	Reads the speed sensor status periodically

- Light Driver
    - Function definitions
    - Type definitions

Service name:	Light_Init
Syntax:	void Light_Init()
Sync/Async:	Synchronous

Reentrancy:	Reentrant
Parameters (in):	None
Parameters (out):	None
Return value:	None
Description:	Reads the speed sensor status periodically

Service name:	Light_ReadStatus
Syntax:	Light_ReadStatus()
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	None
Parameters (out):	None
Return value:	E_OK 1 E_NOK 0
Description:	Reads the light status periodically

- Comm.Mng
  - Function definitions
  - Type definitions

Service name:	ComM_Init
Syntax:	void ComM_Init(const ComM_ConfigType* ConfigPtr)
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	ConfigPtr Pointer to post-build configuration data
Parameters (out):	None
Return value:	None
Description:	Initializes Communication Manager

Service name:	ComM_GetState
Syntax:	Std_ReturnType ComM_GetState(NetworkHandleType Channel, ComM_StateType* State)
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	Channel
Parameters (out):	State
Return value:	Std_ReturnType
Description:	Return current state

Service name:	ComM_SendSpeedMsg
Syntax:	Void ComM_SendSpeedMsg(void)
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	Channel
Parameters (out):	None
Return value:	Nine
Description:	Send Speed Message

Service name:	ComM_SendDoorMsg
Syntax:	Void ComM_SendDoorMsg(void)
Sync/Async:	Synchronous

Reentrancy:	Non Reentrant
Parameters (in):	Channel
Parameters (out):	None
Return value:	Nine
Description:	Send Door Message

Service name:	ComM_SendLightMsg
Syntax:	Void ComM_SendLightMsg(void)
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	Channel
Parameters (out):	None
Return value:	None
Description:	Send Light Message

- Buzzer Driver
  - Function definitions
  - Type definitions

Service name:	Buzzer_Init
Syntax:	Void Buzzer_Init(void)
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	Channel
Parameters (out):	None
Return value:	None
Description:	Buzzer Initialization

Service name:	Buzzer_On
Syntax:	Void Buzzer_On(void)
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	Channel
Parameters (out):	None
Return value:	None
Description:	Buzzer Initialization

Service name:	Buzzer_Off
Syntax:	Void Buzzer_Off(void)
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	Channel
Parameters (out):	None
Return value:	None
Description:	Buzzer Initialization