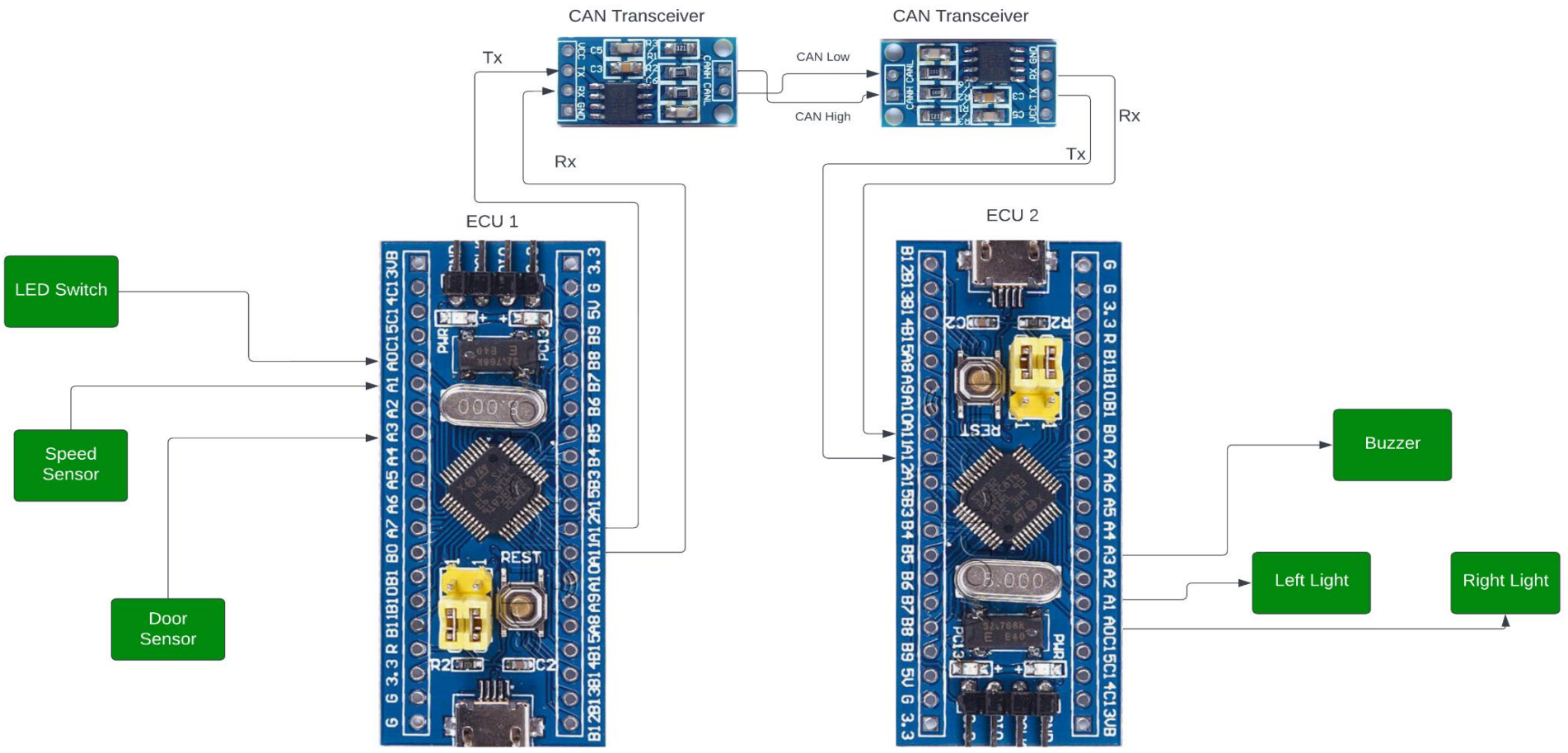
Automotive door control system Static Design

Name: Ahmed Abdelnaser Shaker Aboraya

➢ Case study:

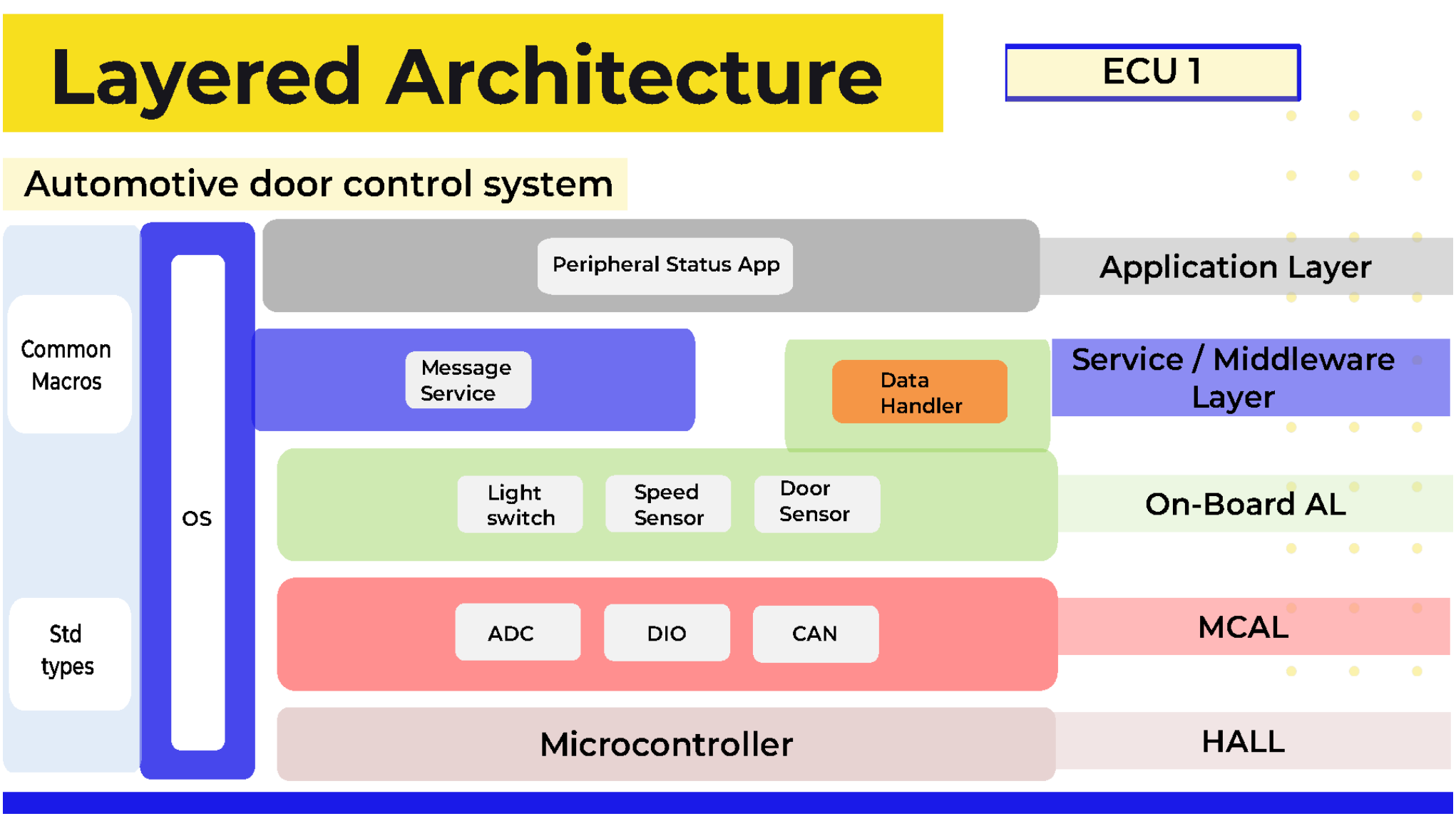
ECU 1 will send status messages to Door state, Light switch state and Speed state periodically to ECU 2 through the CAN protocol so that ECU 2 take Action on car Light and car Alarm based on these status.

➢ Block Diagram :

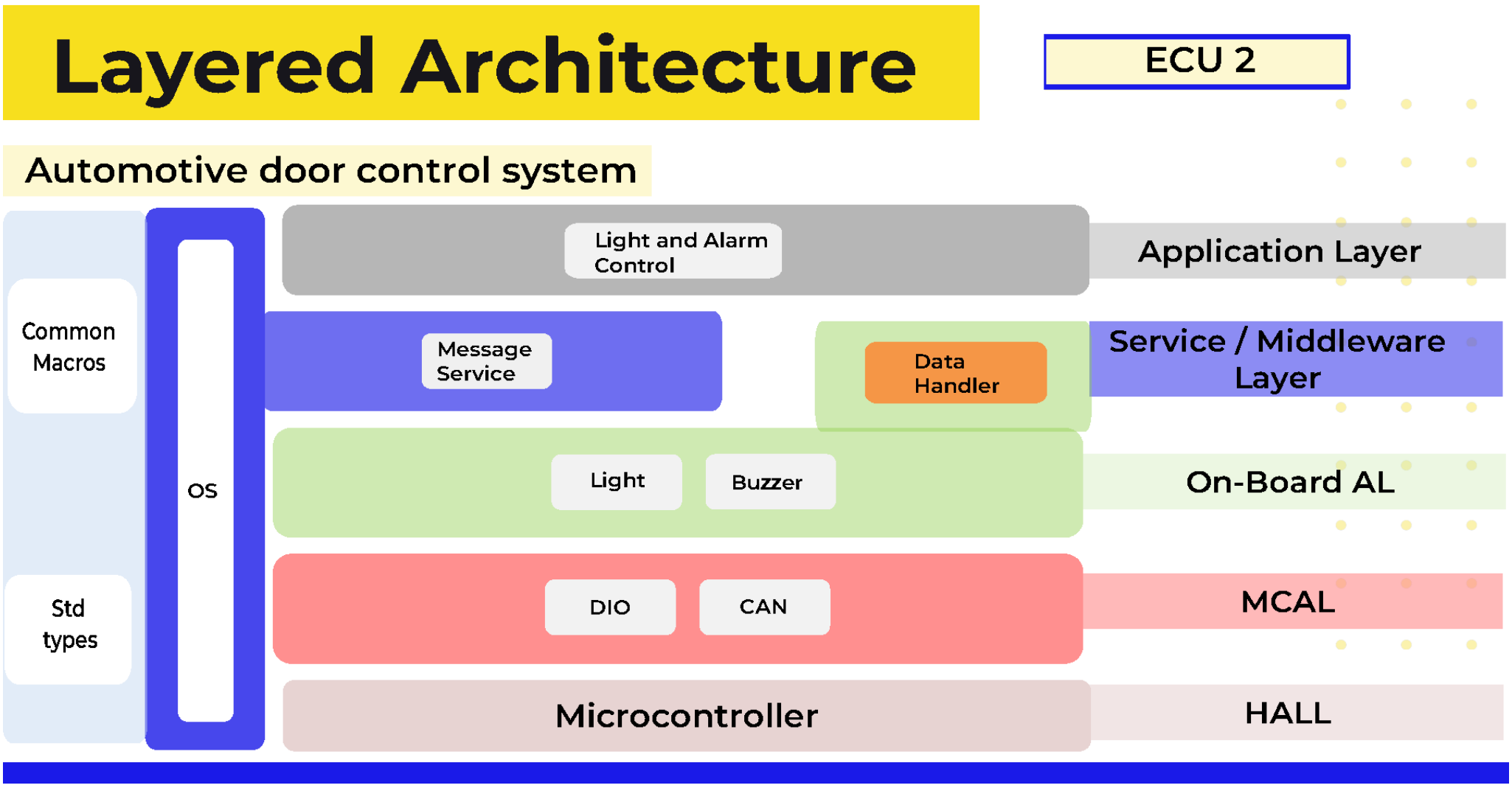


➢ Layered Architecture

1-ECU1:



2- ECU2:



- **ECU components and modules**
  - **1- ECU 1:**
    - ADC
    - DIO
    - CAN
    - Light switch
    - Door Sensor
    - Speed Sensor
    - RTOS
    -
    - Message Service
    - Data Handler
  - **2- ECU 2:**
    - CAN
    - DIO
    - Right and left Light
    - Buzzer
    - RTOS
    - Message Service
    - Data Handler

- **APIs and Data Types**
  - **1- ECU 1:**

DIO Module

Function name	DIO_init		
Arguments	Inputs	ConfigPtr	DIO_PinConfig_t*
		Description: DIO selected Configuration	
Return	void	-	
Description: Initializes DIO according to the parameters of ConfigPtr			

Function name	DIO_ReadChannel		
Arguments	Inputs	ChannelId	DIO_Channel_t
		Description: select DIO channel (DIOx,PINx)	
	Output	PinValue	DIO_Level_t
		Reading value of the input pin	
Return	LOW	0	
	HIGH	1	
Description: Read specific DIOx Pin value			

Function name	DIO_WriteChannel		
Arguments	Inputs	ChannelId	DIO_Channel_t
		Description: select DIO channel (DIOx,PINx)	
		Level	DIO_Level_t
		Description: PIN value	
	Output	-	
Return	void	-	
Description: Write value on specific Pin			

Function name	DIO_ReadPort		
Arguments	Inputs	DIOx	DIO_TypeDef*
		Description: select GPIO peripheral	
	Output	PortValue	DIO_PortLevel_t
		The Reading value of the input pin	
Return	uint8_t	0->255	
Description: Read PORT Value			

Function name	DIO_WritePort		
Arguments	Inputs	DIOx	DIO_TypeDef*
		Description: select GPIO peripheral	
		Level	DIO_PortLevel_t
		Description: PIN value	
Return	void	-	
Description: Write value on PORT			

Name:	DIO_Pin_t		
Type:	Enumeration		
Range:	PIN0	0	Select pin 0
	PIN1	1	Select pin 1
	------→		
	PIN7	7	Select pin 7
Description:	Define DIO Pins to help with selecting pins		

Name:	DIOA, DIOB, DIOC,....
Type:	DIO_TypeDef*
	Pointer to Struct containing port registers
Description:	Define DIO PORTs to help with selecting port and modify it

Name:	DIO_Direction_t		
Type:	Enumeration		
Range:	INPUT	0	Input direction
	OUTPUT	1	output direction
Description:	define Port / Pin Direction to help with configuring DIO		

Name:	DIO_Level_t		
Type:	Enumeration		
Range:	LOW	0	Low pulse
	HIGH	1	High pulse
Description:	define Port / Pin pulse Level to help with configuring DIO		

Name:	DIO_PinConfig_t
Type:	Structure
Elements:	Contain configuration parameters
Description:	DIO Configuration to Set DIO Pins

ADC Module

Function name	ADC_init		
Arguments	Inputs	ConfigPtr	ADC_Config_t*
		Description: ADC selected Configuration	
	Output	-	
Return	void	-	
Description: Initializes ADC according to the parameters of ConfigPtr			

Function name	ADC_ReadChannel		
Arguments	Inputs	ChannelId	ADC_Channel_t
		Description: select ADC channel to read from it	
	Output	PinValue	DIO_Level_t
		value converted from the input pin	
Return	LOW	0	
	HIGH	1	
Description: Read digital value converted from ADC Channels analog values			

Name:	ADC_Config_t
Type:	Structure
Elements:	Contain configuration parameters
Description:	ADC Configuration to Set ADC Channels

Name:	ADC_Channel_t		
Type:	Enumeration		
Range:	ADC_Channel0	0	Select Channel 0
	ADC_Channel1	1	Select Channel 1
	------→		
	ADC_ChannelN	n	Select Channel n
Description:	Define ADC Channels to help select Channels and configure it		

CAN Module

Function name	CAN_init		
Arguments	Inputs	ConfigPtr	CAN_Config_t*
		Description: CAN selected Configuration	
Return	void	-	
Description: Initializes CAN according to the parameters of ConfigPtr			

Function name	CAN_Send		
Arguments	Inputs	Frame_Buffer	CAN_Frame_t*
		Description: Data Frame contain status messages	
	Output	-	
Return	E_OK	0	
	E_NOK	1	
Description: send a CAN Frame contain status messages Data to ECU 2			

Name:	CAN_Config_t
Type:	Structure
Elements:	Contain configuration parameters
Description:	CAN Configuration to Set CAN Modue

HAL

Function name	Door_GetStatus		
Arguments	Inputs	-	
	Output	PinValue	DIO_Level_t
		Reading value of the input pin	
Return	LOW	0	
	HIGH	1	
Description: get Digital status of Door Sensor			

Function name	**LightSw_GetStatus**		
Arguments	**Inputs**	**-**	
	Output	**PinValue**	**DIO_Level_t**
		Reading value of the input pin	
Return	**LOW**	**0**	
	HIGH	**1**	
Description: get Digital status of Light Switch			

Function name	**Car_GetStatus**		
Arguments	**Inputs**	**-**	
	Output	**PinValue**	**DIO_Level_t**
		Reading value of the input pin	
Return	**LOW**	**0**	
	HIGH	**1**	
Description: get Digital status of Speed Sensor			

Function name	**CAN_MessageCreate**		
Arguments	**Inputs**	**MessageID**	**DIO_Channel_t**
		Description: ID of CAN Frame Message	
		StateVal	**DIO_Level_t**
		Description: Store the status value to be sent	
	Output		
Return	**Frame_Buffer**	**CAN_Frame_t***	
	Description: Data Frame contain status message		
Description: Create A CAN Frame using Channel ID and Status Value			

2- ECU 2:

DIO Module

>> Look at DIO Module at ECU 1, it is similar to this Module at ECU 2

CAN Module

Function name	**CAN_init**		
Arguments	**Inputs**	**ConfigPtr**	**CAN_Config_t***
		Description: CAN selected Configuration	
	Output	-	
Return	**void**	-	
Description: Initializes CAN according to the parameters of ConfigPtr			

Function name	**CAN_Receive**		
Arguments	**Inputs**	-	
	Output	-	
Return	**Frame_Buffer**	CAN_Frame_t*	
	Description: Data Frame contain status messages		
Description: Receive a CAN Frame contain status messages Data From ECU 1			

Name:	CAN_Config_t
Type:	**Structure**
Elements:	**Contain configuration parameters**
Description:	CAN Configuration to Set CAN Modue

HAL

Function name	**Light_Control**		
Arguments	**Inputs**	**Light_Status**	**Light_Status_t**
		Description: Light Status Value to set - Light_Status_t : Struct {Light_ON, Light_OFF}	
	Output	-	
Return	**void**	-	
Description: Set State of Light based on the input value of Light_Status			

Function name	Buzzer_Control		
Arguments	Inputs	Buzzer_Status	Buzzer_Status_t
		Description: Buzzer Status Value to set - Buzzer_Status_t: Struct {Buzzer_ON, Buzzer_OFF}	
	Output	-	
Return	void	-	
Description: Set State of Buzzer based on the input value of Buzzer_Status			

Functionname	CAN_MessageHandle		
Arguments	Inputs	Frame_Buffer	CAN_Frame_t*
		Description: ID of CAN Frame Message	
	Output		
Return	Status_Message	message_t	
	Description: >> message_t: struct {MessageID, bool Data}		
Description: extract status message from CAN Frame			

➢ Detailed Block Diagram with APIs and Data Types: