System Static Design

Contents

1 FUNCTIONS API

1.1 ECU 2

1.1.1 MCAL Layer Drivers API

1.1.1.1 CAN Driver

1.1.1.1.1 CAN Driver types

Name	CAN_Container_Type		
Type	Struct		
Elements	CAN32_Module_Type	Can_Module	Contain the CAN module ID that I have specified in the software
	Uint32_t	SystemFreq	It contains the system frequency
	CAN_bit_tq	PhaseSeg1Tq	This argument describes Phase Segment 1
	CAN_bit_tq	PhaseSeg2Tq	This argument describes Phase Segment 2
	CAN_bit_tq	PropagationSegTq	This argument describes Phase Segment 2
	CAN_bit_tq	PropagationSegTq	This argument describes Propagation Delay
	CAN_bit_tq	SyncJumpWidth	Syncro Jump Width
Description	It contains all the needed configurations of the CAN module to initialize it		

Name	CAN_bit_tq
Type	Uint32_t
range	0:
Description	This type describes Phase Segment

Name	CAN32_Module_Type	
Type	Enumerate	
range	0	Module 1
	1	Module 0
Description	This type describes the module representation number in the ECU	

Name	Error_Type		
Type	Enumerate		
range	0	No Error	The transmitting operation is successfully done
	1	Frame Format checking	Check the format of a CAN message is not right
	2	CRC	Calculated and received CRC are not the same
	3	Bit Monitoring	The bus digital level is not the same as the sender
	4	ACK	A frame is not acknowledged by at least one other node
	5	Bit stuffing	Sender didn’t transmit a complementary bit at least after transmitting five homogeneous signal edges
Description	It contains the error type that happened If existed		

1.1.1.1.2 CAN Driver API

Function Name: CAN_Init			
Arguments	Inputs	CAN_Container_Type	Can_Container
		It contains all the needed configurations of the CAN module to initialize it	
	Output	None	None

	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to initialize the can module in the board		

Function Name: CAN_Deinit			
Arguments	Inputs	CAN32_Module_Type	Can_Module
		It contains all the needed configurations of the DIO module to initialize it	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to DE initialize the can module in the board		

Function Name: CAN_Send			
Arguments	Inputs	CAN32_Module_Type	Can_Module
		Contain the CAN module ID that I have specified in the software	
		Uint8_t	Data
		It contains the system frequency	
		Error_Type	&Error_var
		It contains the error type that happened If existed	

	Output	Error_Type	Error_var
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function Sends the data, return whether the data was sent or not and return the error type using a pass by reference method.		

Function Name: CAN_rcv			
Arguments	Inputs	CAN32_Module_Type	Can_Module
		Contain the CAN module ID that I have specified in the software	
		Uint8_t	&Data
		It uses pass be reference method, it returns the received data into that variable	
		Error_Type	&Error_var
		It will return the error type into that variable	
	Output	Error_Type	Error_var
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function receives the data, return whether the data was sent or not and return the error type using a pass by reference method.		

1.1.1.2 DIO Driver

1.1.1.2.1 DIO Driver types

Name	DIO_PinContainer_Type		
Type	Struct		
Elements	Pin_Type	PinNumber	Contains the pin number in the ECU
	Port_Type	PortNumber	It contains the port number in the port
	PinDirection_Type	PinDirection	It contains the pin direction whether it was input or output
	Current_Type	Current	It contains the amount of current the pin will produce
Description	It contains all the needed configurations of the DIO Pin to initialize it		

Name	Pin_Type
Type	Uint8_t
range	0:8
Description	This type describes the pin number

Name	Port_Type
Type	Uint8_t
range	0:6
Description	This type describes the port number

Name	PinDirection_Type	
Type	Enumerate	
range	0	Input
	1	Output
Description	This type describes the direction of the pin	

Name	Current_Type

Type	Enumerate		
range	0	2 micro amperes	the output current will be 2 micro amperes
	1	4 micro amperes	the output current will be 4 micro amperes
	2	8 micro amperes	the output current will be 8 micro amperes
Description	This type describes the output current of the pin		

1.1.1.2.2 DIO Driver API

Function Name: DIO_PinInit			
Arguments	Inputs	DIO_PinContainer_Type	DIO_PinContainer
		It contains all the needed configurations of the DIO Pin to initialize it	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to initialize the DIO Pin in the board		

Function Name: DIO_PinDeInit			
	Inputs	Pin_Type	PinNumber
		Contains the pin number in the ECU	
		Port_Type	PortNumber
		It contains the port number in the ECU	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	

	E_NOK	1
Description:	This Function helps us to reset all the bits in the registers that affect that specific pin	

Function Name: DIO_PinRead			
	Inputs	Pin_Type	PinNumber
		Contains the pin number in the ECU	
		Port_Type	PortNumber
		It contains the port number in the ECU	
	Output	None	None
	Input/Output	None	None
Return	Uint8	The bit value	
Description:	This Function helps us to read a specific pin		

Function Name: DIO_PinWrite			
	Inputs	Pin_Type	PinNumber
		Contains the pin number in the ECU	
		Port_Type	PortNumber
		It contains the port number in the ECU	
		Uint8_t	PinValue
		It contains the value of the pin whether it is 1 or 0	
	Output	None	None

	Input/Output	None	None
Return	None	None	
Description:	This Function helps us to write over a specific pin		

Function Name: DIO_ChangeDirection			
Arguments	Inputs	DIO_PinContainer_Type	DIO_PinContainer
		It contains all the needed configurations of the DIO Pin to initialize it	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to just change the specified DIO Pin direction		

1.1.1.3 GPT Driver

1.1.1.3.1 GPT Driver types

Name	GPT_Container_Type		
Type	Struct		
Elements	GPTMode_Type	GPTMode	the chosen mode of the timer
	GPTNumber_Type	GPTNumber	Contains the timer ID specified in the software
Description	It contains all the needed configurations of the GPT module to initialize it		

Name	GPTMode_Type
Type	Enumerate

range	0	Periodic	The timer channel mode will be periodic, which means it will work again automatically when its overflow
	1	One Shot	The timer channel mode will be periodic, which means it won’t work again automatically when its overflow
	2	Edge Counter	The timer channel mode will count the specified edges, whether it’s rising, falling or both
	3	PWM	The timer channel will work as a pulse width modulator
	4	RTC	The timer channel will work as a real-time clock
Description	the chosen mode of the timer		

Name	GPTNumber_Type		
Type	Enumerate		
range	0	Timer0A	the output current will be 2 micro amperes
	1	Timer0B	the output current will be 4 micro amperes
	2	Timer1A	the output current will be 8 micro amperes
	3	Timer1B	the output current will be 8 micro amperes
Description	Contains the timer ID specified in the software		

1.1.1.3.2 GPT Driver API

Function Name: GPT_Init			
	Inputs	GPT_Container_Type	TimerContainer
		It contains all the needed configurations of the GPT module to initialize it	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to initialize the timer module in the ECU		

Function Name: GPT_DeInit			
	Inputs	GPTNumber_Type	GPTNumber
		Contains the timer ID specified in the software	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to De initialize the timer module in the ECU		

Function Name: GPT_SetValue			
	Inputs	GPTNumber_Type	GPTNumber
		Contains the timer ID specified in the software	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to set the timer’s value		

Function Name: GPT_GetValue			
	Inputs	GPTNumber_Type	GPTNumber
		Contains the timer ID specified in the software	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to read the timer value.		

Function Name: GPT_GetTimerElapsed			
	Inputs	GPTNumber_Type	GPTNumber
		Contains the timer ID specified in the software	
	Output	Uint32_t	data
		Will return the elapsed time in that variable using the method pass by reference	
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to get the elapsed time of the timer		

1.1.1.4 PORT Driver

1.1.1.4.1 PORT Driver types

Name	Port_Container_Type		
Type	Struct		
Elements	DIO_PinContainer_Type	DIO_PinContainer	it contains all the pin’s details of direction, number and current value
	Port_PinModeType	Pin_Mode	Contains the pin alternative mode in the ECU
Description	It contains all the needed pre-configurations of all the pins in the ECU		

1.1.1.4.2 PORT Driver API

Function Name: Port_Init			
Arguments	Inputs	Port_Container_Type	Port_Container

		It contains all the needed pre-configurations of all the pins in the ECU	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to initialize all the pins once the system started		

Typedef struct Port_Container_Type	
DIO_PinContainer_Type	DIO_PinContainer
it contains all the pin's details of direction, number and current value	
Port_PinModeType	Pin_Mode
Contains the pin alternative mode in the ECU	

Function Name: Port_DeInit			
Arguments	Inputs	DIO_PinContainer_Type	Port_Container
		it contains all the pin's details of direction, number and current value	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to De initialize the specified pin		

Function Name: Port_ChangeMode			
Arguments	Inputs	Port_Container_Type	Port_Container
		It contains all the needed pre-configurations of all the pins in the ECU	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to initialize all the pins once the system started		

1.1.2 HAL Layer

1.1.2.1 Buzzer Driver

Function Name: Buzzer_Write			
	Inputs	Uint8_t	Value
		The value will be either STD_HIGH or STD_LOW	
		Uint16	Time
		This value will contain the amount of time the buzzer will go on, if zero will go forever.	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to write a specific value on the buzzer		

1.1.2.2 RightLight Driver

Function Name: RightLight_Write			
	Inputs	Uint8_t	Value
		The value will be either STD_HIGH or STD_LOW	
		Uint16	Time
		This value will contain the amount of time the buzzer will go on, if zero will go forever.	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to write a specific value on the right light		

1.1.2.3 LefLight Driver

Function Name: LefLight_Write			
	Inputs	Uint8_t	Value
		The value will be either STD_HIGH or STD_LOW	
		Uint16	Time
		This value will contain the amount of time the buzzer will go on, if zero will go forever.	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to write a specific value on the LefLight		

1.1.2.4 CAN Handler

Function Name: CAN_Send			
	Inputs	Uint8_t	Data
		The data that will be send by the CAN module	
		Uint16	Error
		Pass by reference variable that will be changed if there are any error with the specified error ID	
		Uint8_t	CAN_ID
		Contain the specified CAN for the sending, whether it is the on-board CAN or the internal CAN	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to send data over the CAN bus		

Function Name: CAN_rcv			
	Inputs	Uint8_t	Data
		The variable that will be send by reference to receive the data over it	
		Uint16	Error
		Pass by reference variable that will be changed if there are any error with the specified error ID	
		Uint8_t	CAN_ID
		Contain the specified CAN for the sending, whether it is the on-board CAN or the internal CAN	
	Output	None	None

	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to send data over the CAN bus		

1.1.3 BASIC COMMUNICATION MANAGER

1.1.3.1 CAN Manager

Function Name: Comm_Mgr_send			
	Inputs	Uint8_t	Data
		The data that will be send by the CAN module	
		Uint8_t	Data_ID
		The data ID that will represent the module’s pre-configurations that will be used	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to send data over a bus		

Function Name: Comm_Mgr_rcv			
	Inputs	Uint8_t	Data
		The data variable will follow the send by reference method, and will return the received data in it	
		Uint8_t	Data_ID
		The data ID that will represent the module’s pre-configurations that will be used	
	Output	None	None
	Input/Output	None	None

Return	E_OK	0
	E_NOK	1
Description:	This Function helps us to receive data over a bus	

1.1.4 Application layer

1.1.4.1 Buzzer Task

Function Name: Buzzer_Task			
	Input	None	
	Output	None	None
	Input/Output	None	None
Return	None		
Description:	This Function control the buzzer and change its value with periodicity 5ms		

1.1.4.2 RightLight Task

Function Name: RightLight_Task			
	Input	None	
	Output	None	None
	Input/Output	None	None
Return	None		
Description:	This Function control the right light and change its value with periodicity 5ms		

1.1.4.3 LeftLight Task

Function Name: Buzzer_Task			
	Input	None	
	Output	None	None
	Input/Output	None	None

Return	None
Description:	This Function control the left light and change its value with periodicity 5ms

1.2 ECU 1

1.2.1 MCAL Layer Drivers API

1.2.1.1 CAN Driver

1.2.1.1.1 CAN Driver types

Name	CAN_Container_Type		
Type	Struct		
Elements	CAN32_Module_Type	Can_Module	Contain the CAN module ID that I have specified in the software
	Uint32_t	SystemFreq	It contains the system frequency
	CAN_bit_tq	PhaseSeg1Tq	This argument describes Phase Segment 1
	CAN_bit_tq	PhaseSeg2Tq	This argument describes Phase Segment 2
	CAN_bit_tq	PropagationSegTq	This argument describes Phase Segment 2
	CAN_bit_tq	PropagationSegTq	This argument describes Propagation Delay
	CAN_bit_tq	SyncJumpWidth	Syncro Jump Width
Description	It contains all the needed configurations of the CAN module to initialize it		

Name	CAN_bit_tq
Type	Uint32_t
range	0:
Description	This type describes Phase Segment

Name	CAN32_Module_Type	
Type	Enumerate	
range	0	Module 1
	1	Module 0
Description	This type describes the module representation number in the ECU	

Name	Error_Type		
Type	Enumerate		
range	0	No Error	The transmitting operation is successfully done
	1	Frame Format checking	Check the format of a CAN message is not right
	2	CRC	Calculated and received CRC are not the same
	3	Bit Monitoring	The bus digital level is not the same as the sender
	4	ACK	A frame is not acknowledged by at least one other node
	5	Bit stuffing	Sender didn't transmit a complementary bit at least after transmitting five homogeneous signal edges
Description	It contains the error type that happened If existed		

1.2.1.1.2 CAN Driver API

Function Name: CAN_Init			
Arguments	Inputs	CAN_Container_Type	Can_Container
		It contains all the needed configurations of the CAN module to initialize it	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to initialize the can module in the board		

Function Name: CAN_Deinit			
Arguments	Inputs	CAN32_Module_Type	Can_Module
		It contains all the needed configurations of the DIO module to initialize it	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to DE initialize the can module in the board		

Function Name: CAN_Send			
Arguments	Inputs	CAN32_Module_Type	Can_Module
		Contain the CAN module ID that I have specified in the software	
		Uint8_t	Data
		It contains the system frequency	
		Error_Type	&Error_var
		It contains the error type that happened If existed	
	Output	Error_Type	Error_var
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function Sends the data, return whether the data was sent or not and return the error type using a pass by reference method.		

Function Name: CAN_rcv			
Arguments	Inputs	CAN32_Module_Type	Can_Module
		Contain the CAN module ID that I have specified in the software	
		Uint8_t	&Data
		It uses pass be reference method, it returns the received data into that variable	
		Error_Type	&Error_var

		It will return the error type into that variable	
	Output	Error_Type	Error_var
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function receives the data, return whether the data was sent or not and return the error type using a pass by reference method.		

1.2.1.2 DIO Driver

1.2.1.2.1 DIO Driver types

Name	DIO_PinContainer_Type		
Type	Struct		
Elements	Pin_Type	PinNumber	Contains the pin number in the ECU
	Port_Type	PortNumber	It contains the port number in the port
	PinDirection_Type	PinDirection	It contains the pin direction whether it was input or output
	Current_Type	Current	It contains the amount of current the pin will produce
Description	It contains all the needed configurations of the DIO Pin to initialize it		

Name	Pin_Type
Type	Uint8_t
range	0:8
Description	This type describes the pin number

Name	Port_Type
Type	Uint8_t

range	0:6
Description	This type describes the port number

Name	PinDirection_Type	
Type	Enumerate	
range	0	Input
	1	Output
Description	This type describes the direction of the pin	

Name	Current_Type		
Type	Enumerate		
range	0	2 micro amperes	the output current will be 2 micro amperes
	1	4 micro amperes	the output current will be 4 micro amperes
	2	8 micro amperes	the output current will be 8 micro amperes
Description	This type describes the output current of the pin		

1.2.1.2.2 DIO Driver API

Function Name: DIO_PinInit			
Arguments	Inputs	DIO_PinContainer_Type	DIO_PinContainer
		It contains all the needed configurations of the DIO Pin to initialize it	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to initialize the DIO Pin in the board		

Function Name: DIO_PinDeInit			
	Inputs	Pin_Type	PinNumber
		Contains the pin number in the ECU	
		Port_Type	PortNumber
		It contains the port number in the ECU	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to reset all the bits in the registers that affect that specific pin		

Function Name: DIO_PinRead			
	Inputs	Pin_Type	PinNumber
		Contains the pin number in the ECU	
		Port_Type	PortNumber
		It contains the port number in the ECU	
	Output	None	None
	Input/Output	None	None
Return	Uint8	The bit value	
Description:	This Function helps us to read a specific pin		

Function Name: DIO_PinWrite			
	Inputs	Pin_Type	PinNumber
		Contains the pin number in the ECU	
		Port_Type	PortNumber
		It contains the port number in the ECU	
		Uint8_t	PinValue
		It contains the value of the pin whether it is 1 or 0	
	Output	None	None
	Input/Output	None	None
Return	None	None	
Description:	This Function helps us to write over a specific pin		

Function Name: DIO_ChangeDirection			
Arguments	Inputs	DIO_PinContainer_Type	DIO_PinContainer
		It contains all the needed configurations of the DIO Pin to initialize it	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to just change the specified DIO Pin direction		

1.2.1.3 GPT Driver

1.2.1.3.1 GPT Driver types

Name	GPT_Container_Type		
Type	Struct		
Elements	GPTMode_Type	GPTMode	the chosen mode of the timer
	GPTNumber_Type	GPTNumber	Contains the timer ID specified in the software
Description	It contains all the needed configurations of the GPT module to initialize it		

Name	GPTMode_Type		
Type	Enumerate		
range	0	Periodic	The timer channel mode will be periodic, which means it will work again automatically when its overflow
	1	One Shot	The timer channel mode will be periodic, which means it won’t work again automatically when its overflow
	2	Edge Counter	The timer channel mode will count the specified edges, whether it’s rising, falling or both
	3	PWM	The timer channel will work as a pulse width modulator
	4	RTC	The timer channel will work as a real-time clock
Description	the chosen mode of the timer		

Name	GPTNumber_Type		
Type	Enumerate		
range	0	Timer0A	the output current will be 2 micro amperes
	1	Timer0B	the output current will be 4 micro amperes
	2	Timer1A	the output current will be 8 micro amperes
	3	Timer1B	the output current will be 8 micro amperes
Description	Contains the timer ID specified in the software		

1.2.1.3.2 GPT Driver API

Function Name: GPT_Init			
	Inputs	GPT_Container_Type	TimerContainer
		It contains all the needed configurations of the GPT module to initialize it	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to initialize the timer module in the ECU		

Function Name: GPT_DeInit			
	Inputs	GPTNumber_Type	GPTNumber
		Contains the timer ID specified in the software	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	

Description:	This Function helps us to De initialize the timer module in the ECU

Function Name: GPT_SetValue			
	Inputs	GPTNumber_Type	GPTNumber
		Contains the timer ID specified in the software	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to set the timer’s value		

Function Name: GPT_GetValue			
	Inputs	GPTNumber_Type	GPTNumber
		Contains the timer ID specified in the software	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to read the timer value.		

Function Name: GPT_GetTimerElapsed			
	Inputs	GPTNumber_Type	GPTNumber
		Contains the timer ID specified in the software	
	Output	Uint32_t	data
		Will return the elapsed time in that variable using the method pass by reference	
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to get the elapsed time of the timer		

1.2.1.4 PORT Driver

1.2.1.4.1 PORT Driver types

Name	Port_Container_Type		
Type	Struct		
Elements	DIO_PinContainer_Type	DIO_PinContainer	it contains all the pin's details of direction, number and current value
	Port_PinModeType	Pin_Mode	Contains the pin alternative mode in the ECU
Description	It contains all the needed pre-configurations of all the pins in the ECU		

1.2.1.4.2 PORT Driver API

Function Name: Port_Init			
Arguments	Inputs	Port_Container_Type	Port_Container
		It contains all the needed pre-configurations of all the pins in the ECU	
	Output	None	None

	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to initialize all the pins once the system started		

Typedef struct Port_Container_Type	
DIO_PinContainer_Type	DIO_PinContainer
it contains all the pin’s details of direction, number and current value	
Port_PinModeType	Pin_Mode
Contains the pin alternative mode in the ECU	

Function Name: Port_DeInit			
Arguments	Inputs	DIO_PinContainer_Type	Port_Container
		it contains all the pin’s details of direction, number and current value	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to De initialize the specified pin		

Function Name: Port_ChangeMode			
Arguments	Inputs	Port_Container_Type	Port_Container
		It contains all the needed pre-configurations of all the pins in the ECU	
	Output	None	None

	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to initialize all the pins once the system started		

1.2.2 HAL Layer

1.2.2.1 DoorSensor Driver

Function Name: Door_Read			
		Uint8_t	&Data
		The variable follows pass-by-reference method. The function will read the door sensor and save its value in that variable	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to read the door state whether its open or closed		

1.2.2.2 Light switch Driver

Function Name: LightSwitch_Read			
	Inputs	Uint8_t	&Value
		The variable follows pass-by-reference method. The value will represent the button state whether its pressed or not	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	

Description:	This Function helps us to read the door state

1.2.2.3 Speed Sensor Driver

Function Name: SpeedSensor_Read			
	Inputs	Uint8_t	&Value
		The variable follows pass-by-reference method. The value will represent the car speed whether it is moving or not	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to know whether the car is moving or not		

1.2.2.4 CAN Handler

Function Name: CAN_Send			
	Inputs	Uint8_t	Data
		The data that will be send by the CAN module	
		Uint16	Error
		Pass by reference variable that will be changed if there are any error with the specified error ID	
		Uint8_t	CAN_ID
		Contain the specified CAN for the sending, whether it is the on-board CAN or the internal CAN	
	Output	None	None
	Input/Output	None	None

Return	E_OK	0
	E_NOK	1
Description:	This Function helps us to send data over the CAN bus	

Function Name: CAN_rcv			
	Inputs	Uint8_t	&Data
		The variable follows pass-by-reference method. The variable that will be contain the received data	
		Uint16	&Error
		Pass by reference variable that will be changed if there are any error with the specified error ID	
		Uint8_t	CAN_ID
		Contain the specified CAN for the sending, whether it is the on-board CAN or the internal CAN	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	This Function helps us to read the received data over the CAN bus		

1.2.3 Application layer

1.2.3.1 Door Sensor Task

Function Name: Door_Task			
	Input	None	
	Output	None	None
	Input/Output	None	None
Return	None		

Description:	This Function reads the door state and send it over the can bus every 10ms

1.2.3.2 Speed Sensor Task

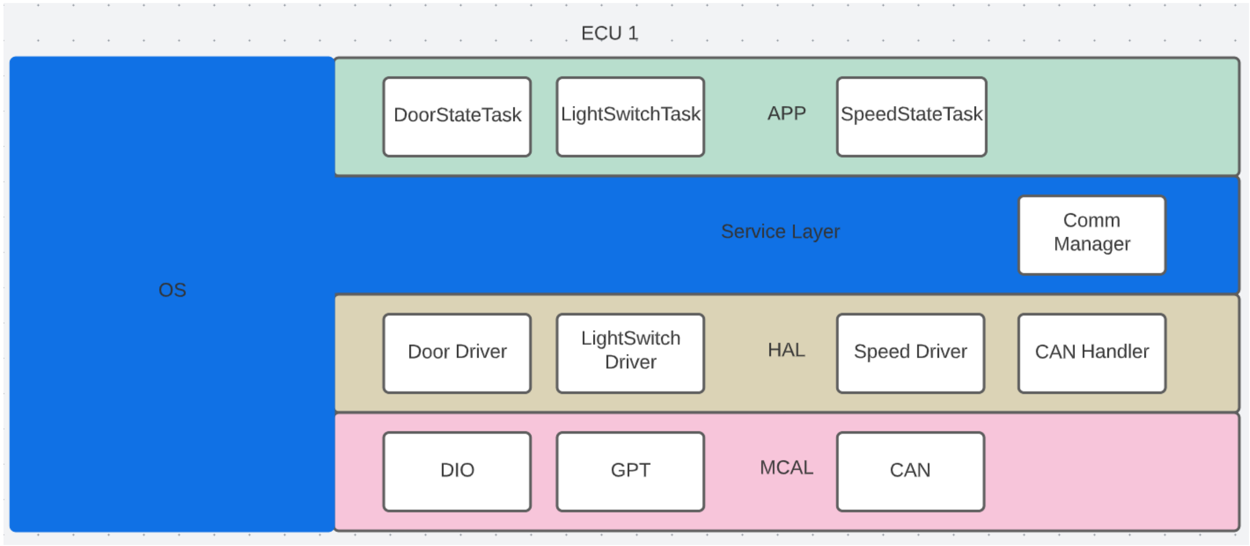
Function Name: SpeedSensor_Task			
	Input	None	
	Output	None	None
	Input/Output	None	None
Return	None		
Description:	This Function reads whether the car is moving or not and send it over the CAN bus every 5ms		

1.2.3.3 LightSwitch Task

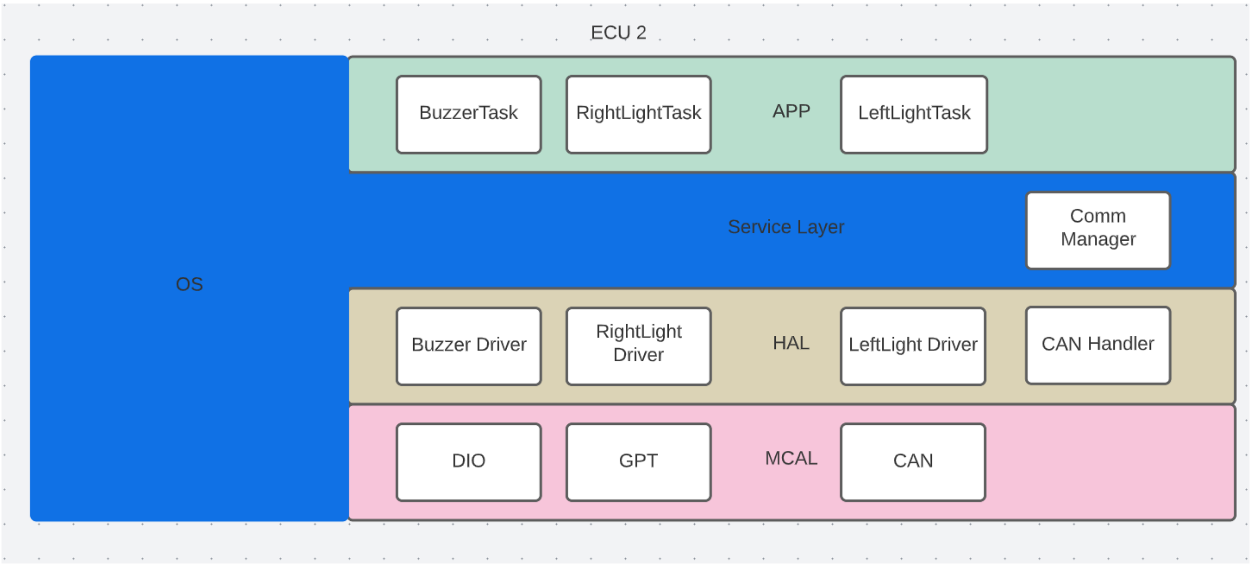
Function Name: Buzzer_Task			
	Input	None	
	Output	None	None
	Input/Output	None	None
Return	None		
Description:	This Function reads the switch and send it over a CAN bus every 20ms		

2 Layered architectures

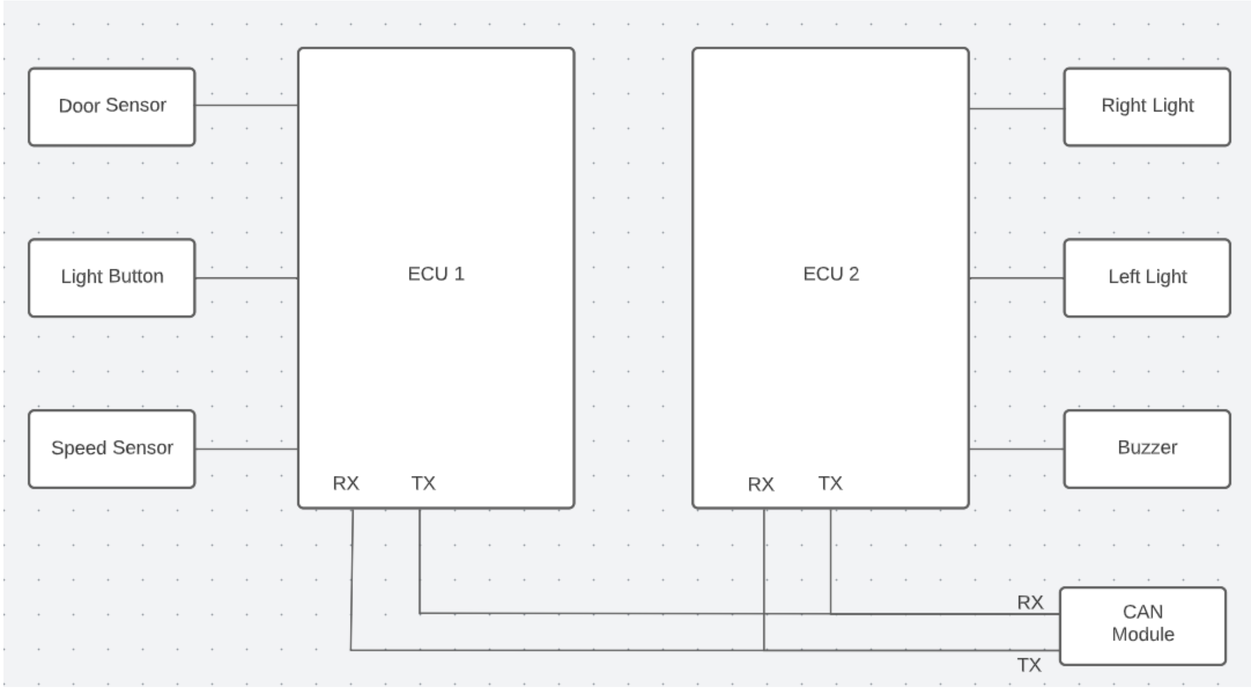
2.1 ECU 1



2.2 ECU 2



3 System block


Door Sensor
Light Button
Speed Sensor
ECU 1
RX
TX
ECU 2
RX
TX
Right Light
Left Light
Buzzer
RX
CAN
Module
TX