Ministry of Communications and Information Technology

egyptfwd

مستقبلنا رقمي

Sprints

JDACITY

Embedded System Advanced Track

EgFWD-Aug Cohort,2022

Embedded Software Design Graduation Project

Automotive Door Control System Design

Part 1

Fully Static Design

Mohamed Hassaneen Ahmed

September, 2022

Content:

Introduction:

In the automotive industry, ECUs are communicating together to provide safety and driver's comfort. ECUs are microcontrollers connected with input and output devices, input devices to sense the surrounding environments, and output devices to perform actions according to readings that came from input devices.

Project Requirements

1. Provide Fully Static Design
2. Provide a Fully Dynamic design

For two ECUs communicating together to control car lights according to door state, light switch state, and car speed state.

Project Specification:

Static design analysis	For ECU 1: 1. Make the layered architecture 2. Specify ECU components and modules 3. Provide full detailed APIs for each module as well as a detailed description for the used typedefs 4. Prepare your folder structure according to the previous points For ECU 2: 1. Make the layered architecture 2. Specify ECU components and modules 3. Provide full detailed APIs for each module as well as a detailed description for the used typedefs 4. Prepare your folder structure according to the previous points You should deliver a pdf file containing all your work and a video recording where you will discuss your work (maximum 3min long)

The layered architecture:
For ECU1:

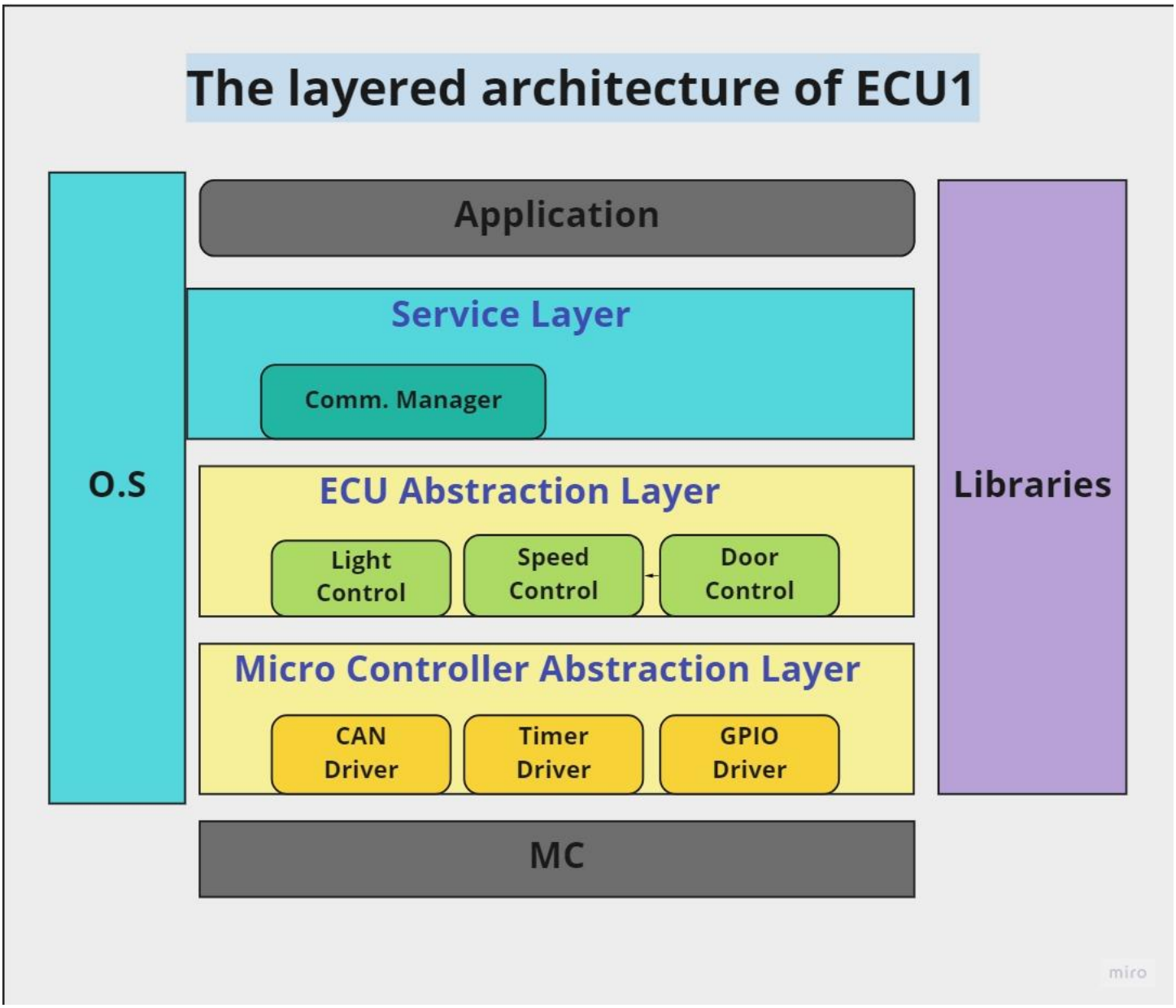

Figure 1- ECU1 LAYERED ARCH.

For ECU1:

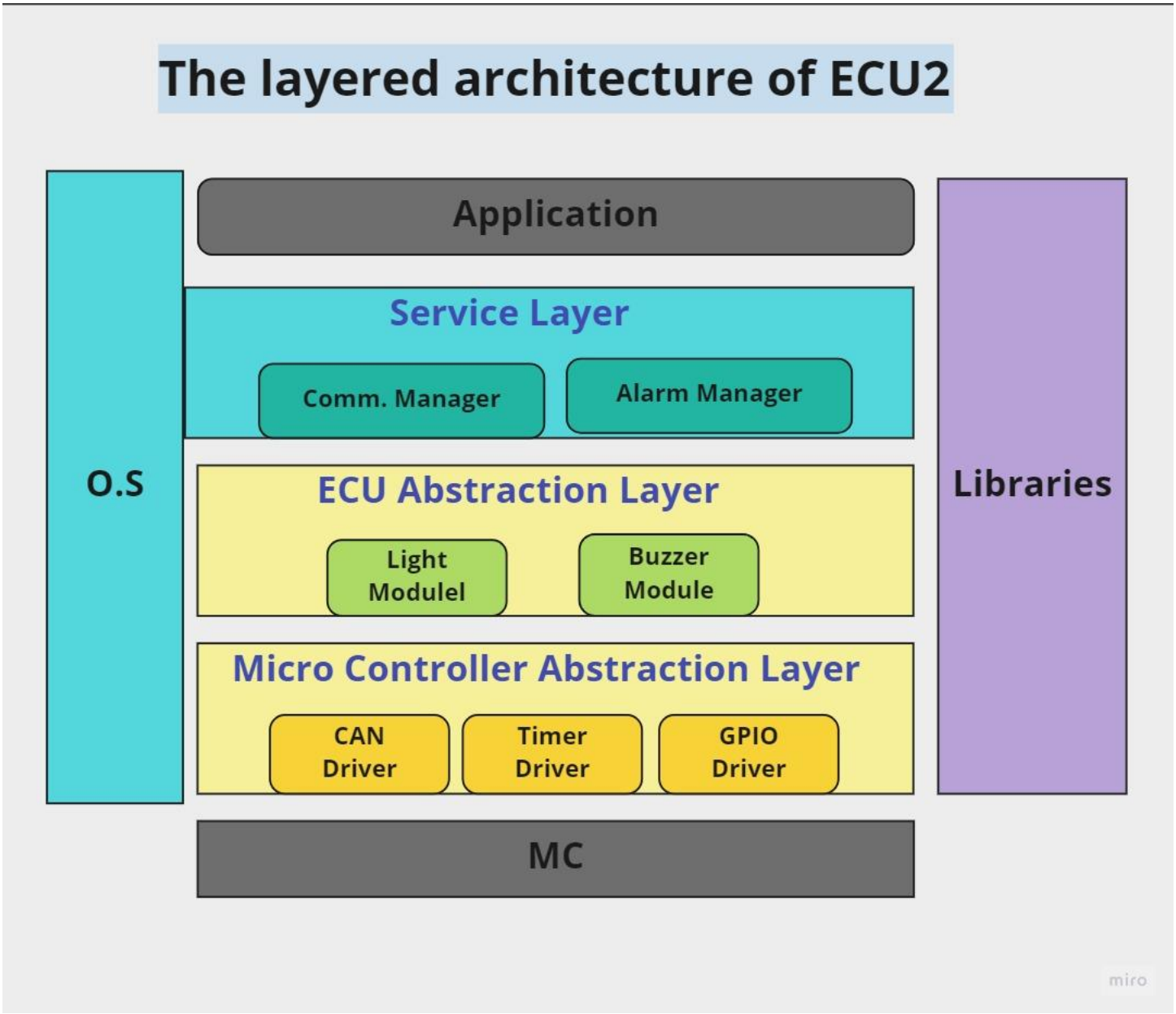

Figure 2-ECU2 LAYERED ARCH.

Specify ECU1 components and modules:

For ECU1:

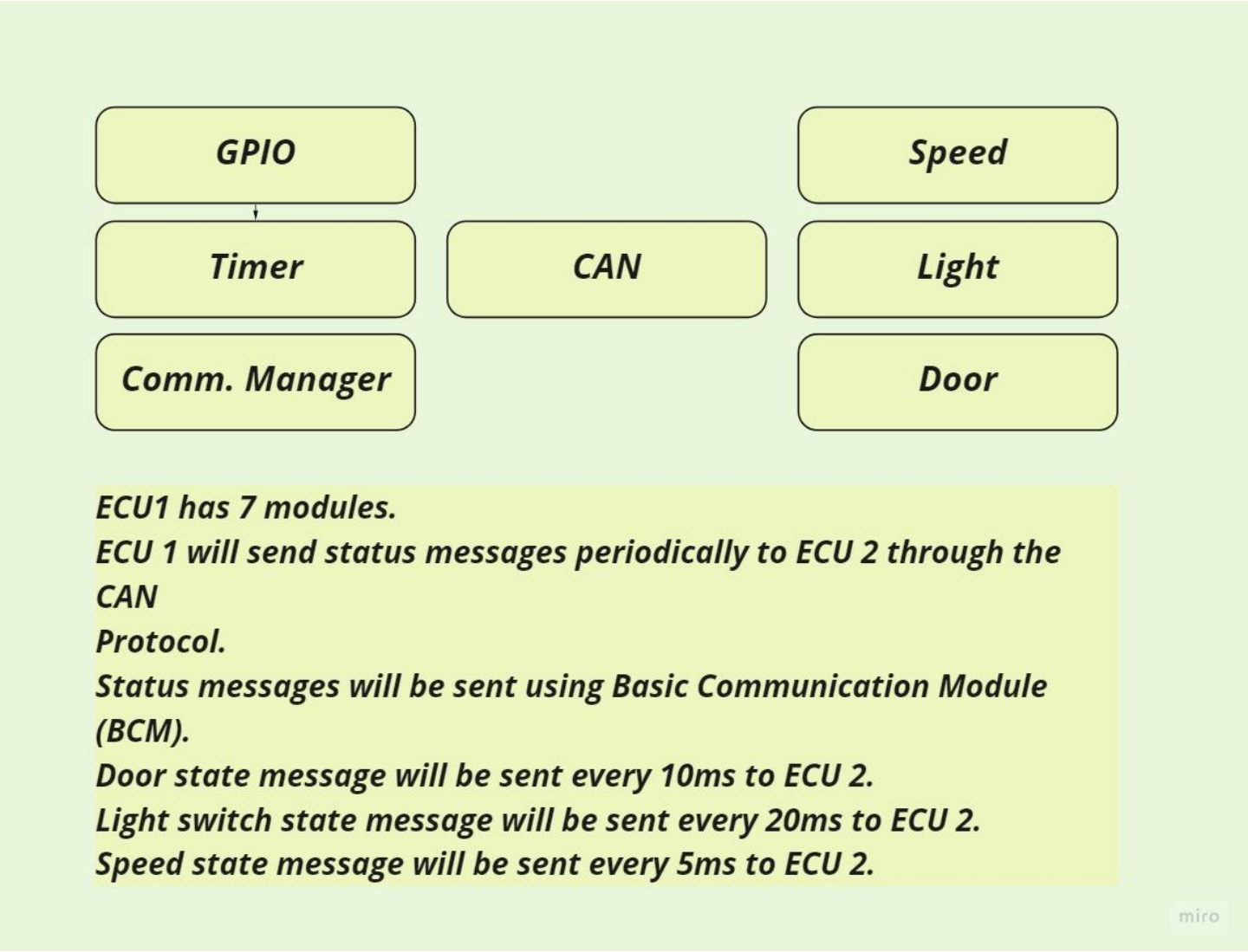

Figure 3- Specify ECU1 components and modules

For ECU2:

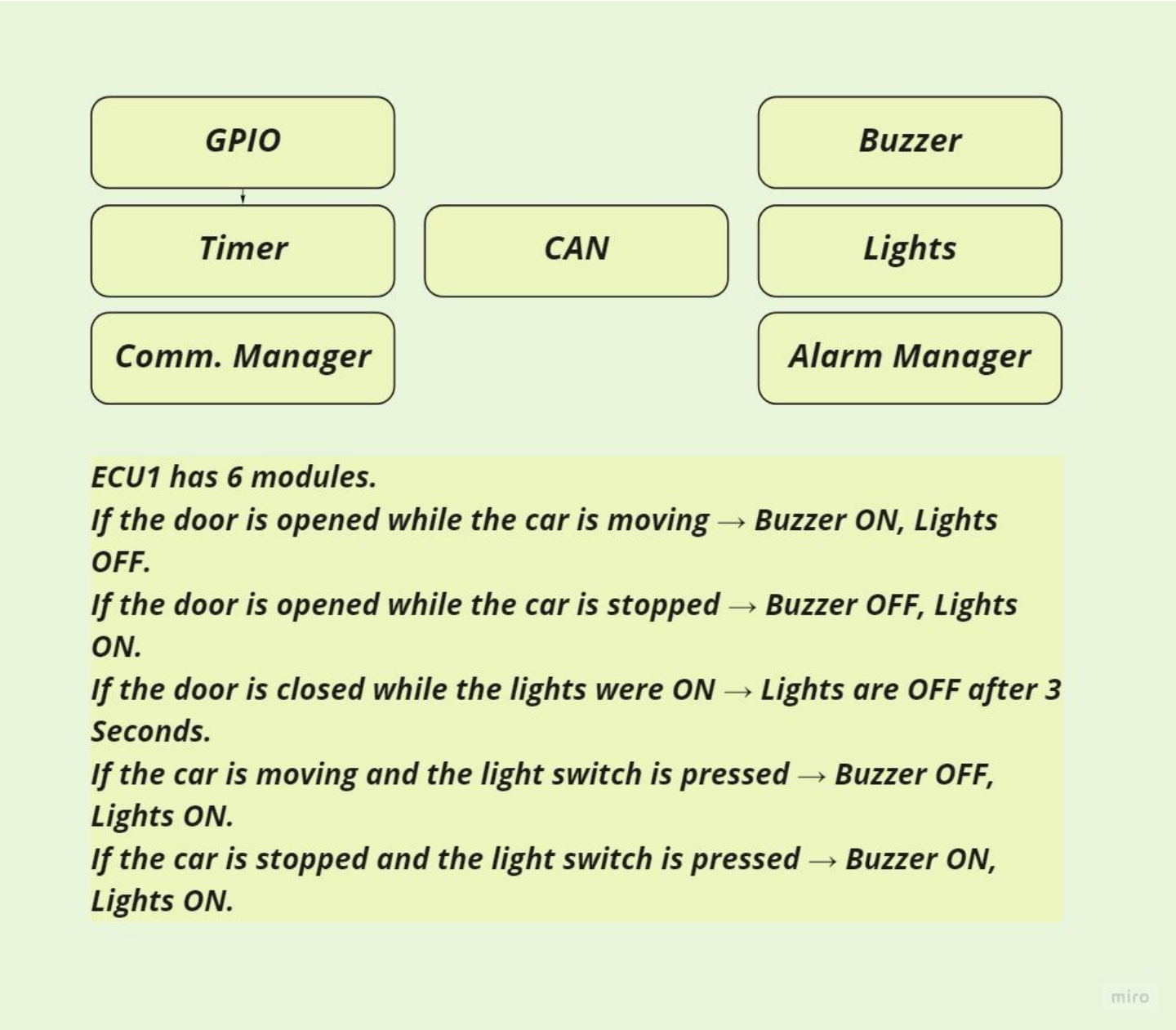

Figure 4- Specify ECU2 components and modules

Modules Design:

For ECU1 & ECU2:

1. **GPIO Module**

a. **Types**

Name	**GPIO_PinLevelType**	
Type	**Enumeration**	
Range	**GPIO_PIN_LOW**	**0**
	GPIO_PIN_HIGH	**1**
Description		

Name	**GPIO_PinDirectionType**	
Type	**Enumeration**	
Range	**GPIO_PIN_INPUT**	**0**
	GPIO_PIN_OUTPUT	**1**
Description		

Name	**GPIO_PinDigEnType**	
Type	**Enumeration**	
Range	**GPIO_PIN_DIGITAL_DIS**	**0**
	GPIO_PIN_DIGITAL_EN	**1**
Description		

Name	**GPIO_PortIDType**	
Type	**Enumeration**	
Range	**GPIO_PORTA**	**0**
	GPIO_PORTB	**1**
	GPIO_PORTC	**2**
	GPIO_PORTD	**3**
	GPIO_PORTE	**4**
	GPIO_PORTF	**5**
Description		

Name	**GPIO_PinIDType**	
Type	**Enumeration**	
Range	**GPIO_PN0**	**0**
	GPIO_PN1	**1**
	GPIO_PN2	**2**

	GPIO_PN3	3
	GPIO_PN4	4
	GPIO_PN5	5
	GPIO_PN6	6
	GPIO_PN7	7
Description		

Name	GPIO_CfgType		
Type	Structure		
Range	Port_id	GPIO_PortIDType	Select port
	Pin_id	GPIO_PinIDType	Select pin
	Pin_dir	GPIO_PinDirectionType	Set direction
	Pin_digEn	GPIO_PinDigEnType	Set digital enable
Description			

b. APIs

Function Name	GPIO_Init	
Arguments	INPUTS	
	GPIO_CfgPtr	GPIO_CfgType
	Description	Ptr to cfg array
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	GPIO Initialization	

Function Name	GPIO_PinWrite	
Arguments	INPUTS	
	Pin_id	GPIO_PortIDType
	Description	Select pin number
	Port_id	GPIO_PinIDType
	Description	Select port number
	Pin_Level	GPIO_PinLevelType
	Description	Set pin status
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	GPIO – Set pin	

Function Name	GPIO_PinRead	
Arguments	INPUTS	
	Pin_id	GPIO_PortIDType
	Description	Select pin number
	Port_id	GPIO_PinIDType
	Description	Select port number
	OUTPUTS	
	Pin_Level	GPIO_PinLevelType
	Description	Read pin status
Return	E_OK	0
	E_NOK	1
Description	GPIO – Read pin	

2. Timer Module:

a. Types

Name	GPT_ChannelIDType	
Type	Enumeration	
Range	GPT_TIMER0	0
	GPT_TIMER1	1
	GPT_TIMER2	2
	GPT_TIMER3	3
	GPT_TIMER4	4
	GPT_TIMER5	5
Description		

Name	GPT_DirectionType	
Type	Enumeration	
Range	GPT_UPCOUNT	0
	GPT_DOWNCOUNT	1
Description		

Name	GPT_SenseType	
Type	Enumeration	
Range	GPT_LEVELSENSE	0
	GPT_EDGESENSE	1
Description		

Name	**GPT_LoadValueType**
Type	**Uint32_t**
Range	**0 – (2^32-1)**
Description	

Name	**GPT_CfgType**		
Type	**Structure**		
Range	**Module_id**	**GPT_ChannelIDType**	**Select module**
	Load_value	**GPT_LoadValueType**	**Count value**
	Direction	**GPT_DirectionType**	**Up or Down**
	Sense	**GPT_SenseType**	**Level or edge**
Description			

b. APIs

Function Name	**Timer_Init**	
Arguments	**INPUTS**	
	GPT_Cfgptr	**GPT_CfgType**
	Description	**Ptr to cfg array**
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**Timer Initialization**	

Function Name	**Timer_Start**	
Arguments	**INPUTS**	
	Module_id	**GPT_ChannelIDType**
	Description	**Select module**
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**Start Timer**	
Function Name	**Timer_Stop**	
Arguments	**INPUTS**	

	Module_id	**GPT_ChannelIDType**
	Description	**Select module**
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**StopTimer**	

3. CAN Module:

a. Types

Name	**CAN_ChannelType**	
Type	**Enumeration**	
Range	**CAN_CH1**	**0**
	CAN_CH2	**1**
Description		

b. APIs

Function Name	**CAN_Init**	
Arguments	**INPUTS**	
	channel	**CAN_ChannelType**
	Description	**Select channel**
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**CAN Initialization**	

Function Name	**CAN_SendMessage**	
Arguments	**INPUTS**	
	messageptr	**CAN _CfgType**
	Description	**Ptr to message structure**
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**

Description	CAN – send message

Function Name	CAN_SendMessage	
Arguments	INPUTS	
	MessageSentPtr	CAN _CfgType
	Description	Ptr to message structure
	OUTPUTS	
	MessageRecieve	CAN _CfgType
	Description	Receive message
Return	E_OK	0
	E_NOK	1
Description	CAN – Recieve message	

Function Name	CAN_ReadyToSendMessage	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	CAN Initialization	

Function Name	CAN_ReadyToReadMessage	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	CAN Initialization	

For ECU1 Only:

1) **Light Module**
   a. **Types**

Name	Light_StatusType	
Type	Enumeration	
Range	LIGHT_OFF	0
	LIGHT_ON	1
Description		

b. APIs

Function Name	Light_Init	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	Light Initialization	

Function Name	Light_ReadStatus	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	Reading light status periodically	

Function Name	Light_GetStatus	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	Light_status	Light_StatusType
	Description	To get light status
Return	E_OK	0
	E_NOK	1
Description	Get light status	

2) Door Module

a. Types

Name	Door_StatusType	
Type	Enumeration	
Range	DOOR_OPEN	0
	DOOR_CLOSED	1

Description		

b. APIs

Function Name	Door_Init	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	Door Initialization	

Function Name	Door_ReadStatus	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	Reading Door status periodically	

Function Name	Door_GetStatus	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	Door_status	Door_StatusType
	Description	To get Door status
Return	E_OK	0
	E_NOK	1
Description	Get Door status	

3) Speed Module

a. Types

Name	Speed_StatusType	
Type	Enumeration	
Range	SPEED_STOP	0

	SPEED_MOVE	**1**
Description		

b. APIs

Function Name	**Speed_Init**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**Speed Initialization**	

Function Name	**Speed_ReadStatus**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**Reading Speed status periodically**	

Function Name	**Speed_GetStatus**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	Speed_status	**Speed_StatusType**
	Description	**To get Speed status**
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**Get Speed status**	

4) Comm. Manager

a. Types

Name	CommMgr_MessageIDType	
Type	Enumeration	
Range	SPEED_ID	0
	DOOR_ID	1
	LIGHT_ID	2
Description		

b. APIs

Function Name	CommMgr_SpeedSendMessage	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	CAN Initialization	

Function Name	CommMgr_DoorSendMessage	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	CAN Initialization	

Function Name	CommMgr_LighrSendMessage	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	CAN Initialization	

For ECU2 Only:

1) **Light Module**
   a. **Types**

Name	**Light_StatusType**	
Type	**Enumeration**	
Range	**LIGHT_OFF**	**0**
	LIGHT_ON	**1**
Description		

   b. **APIs**

Function Name	**Light_Init**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**Light Initialization**	

Function Name	**Light_ReadStatus**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**Reading light status periodically**	

Function Name	**Light_GetStatus**	
Arguments	**INPUTS**	
	NON	

	Description	
	OUTPUTS	
	Light_status	**Light_StatusType**
	Description	**To get light status**
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**Get light status**	

Function Name	**Light_On**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**Light Initialization**	

Function Name	**Light_Off**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**Light Initialization**	

2) Buzzer Module

a. Types

Name	**Buzzer_StatusType**	
Type	**Enumeration**	
Range	**BUZZER_OFF**	**0**
	BUZZER_ON	**1**
Description		

b. APIs

Function Name	Buzzer_Init	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	Buzzer Initialization	

Function Name	Buzzer_ReadStatus	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	E_OK	0
	E_NOK	1
Description	Reading Buzzer status periodically	

Function Name	Buzzer_GetStatus	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	Buzzer_status	Buzzer_StatusType
	Description	To get Buzzer status
Return	E_OK	0
	E_NOK	1
Description	Get Buzzer status	

Function Name	Buzzer_On	
Arguments	INPUTS	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	

Return	**E_OK**	**0**
	E_NOK	**1**
Description	**Buzzer on**	

Function Name	**Buzzer_Off**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**Buzzer off**	

3) Comm. Manager

a. Types

Name	**CommMgr_MessageIDType**	
Type	**Enumeration**	
Range	**SPEED_ID**	**0**
	DOOR_ID	**1**
	LIGHT_ID	**2**
Description		

b. APIs

Function Name	**CommMgr_RecieveMessageHandler**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**CAN Initialization**	

Function Name	**CommMgr_ReadMessage**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	

Return	**E_OK**	**0**
	E_NOK	**1**
Description	**CAN Initialization**	

Function Name	**CommMgr_GetSpeedMessage**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	Speed_Message	**CommMgr_MessageType**
	Description	**Read speed status**
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**CAN Initialization**	

Function Name	**CommMgr_GetDoorMessage**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	Door_Message	**CommMgr_MessageType**
	Description	**Read Door status**
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**CAN Initialization**	

Function Name	**CommMgr_GetLightMessage**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	Light_Message	**CommMgr_MessageType**
	Description	**Read Light status**
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**CAN Initialization**	

4) **Alarm Manager**
   a. **Types**
   b. **APIs**

Function Name	**Alarm_TASK1**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**CAN Initialization**	

Function Name	**Alarm_TASK2**	
Arguments	**INPUTS**	
	NON	
	Description	
	OUTPUTS	
	NON	
	Description	
Return	**E_OK**	**0**
	E_NOK	**1**
Description	**CAN Initialization**	