

Embedded Systems Advanced Nanodegree Program  
Embedded Software Design  
Project: Automotive door control system design  
Static design analysis

**Name:** Elsayed Ayman Elsayed Ali Habib

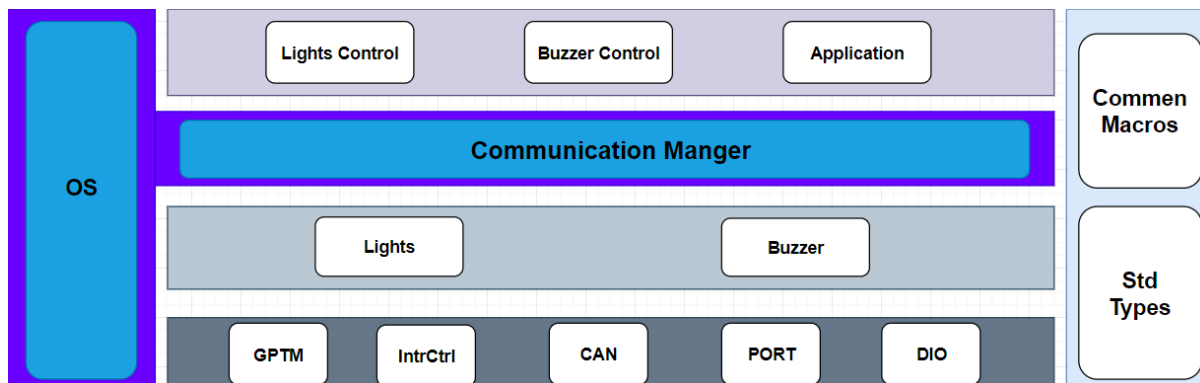
**Email:** [sayedaymanhabib@gmail.com](mailto:sayedaymanhabib@gmail.com)

# Contents

Second ECU.....	3
Layered Architecture of ECU2.....	3
ECU2 Modules:.....	3
DIO Driver.....	4
DIO typedefs: .....	4
DIO Structures:.....	5
DIO Function: .....	6
Port Driver.....	7
Port typedefs:.....	7
Port Structures:.....	8
Port Function: .....	8
Timer Driver .....	9
Timer typedefs: .....	9
Timer Structure:.....	10
Timer Function: .....	11
Interrupt Driver.....	13
Interrupt Typedefs .....	13
Interrupt Structure:.....	14
Interrupt Function:.....	14
CAN Driver.....	15
CAN Typedefs.....	15
CAN Structure: .....	15
CAN Functions:.....	16
ECU2 Components .....	18
Buzzer.....	18
Buzzer Function.....	18
Lights .....	19
Lights Function .....	19

## Second ECU.

### Layered Architecture of ECU2



### ECU2 Modules:

- 1- DIO Driver.
- 2- PORT Driver.
- 3- CAN Driver.
- 4- Interrupt Driver.
- 5- Timer Driver.

## DIO Driver

### DIO typedefs:

Name:	Dio_ChannelType
Type:	uint
Range:	0 --> <Number of Channels>
Description:	Numeric ID of a DIO channel.

Name:	Dio_PortType
Type:	Unit
Range:	0 --> <Number of Ports>
Description:	Numeric ID of a DIO port.

Name:	Dio_LevelType
Type:	uint8
Range:	STD_LOW -> 0x00 ( equal 0V ) STD_HIGH -> 0x01 ( equal 3.3V OR 5V )
Description:	possible levels of DIO channel for input or output pins

Name:	Dio_PortLevelType
Type:	uint
Range:	0 --> <Number of possible Initial levels of DIO Port>
Description:	If the MCU owns ports of different port pins

## DIO Structures:

Name:	Dio_ChannelGroupType	
Type:	Structure	
Element:	mask	This element mask which defines the positions of the channel group.
	offset	This element shall be the position of the Channel Group on the port, counted from the LSB.
	port	This shall be the port on which the Channel group is defined.
Description	Type for the definition of a channel group, which consists of several adjoining channels within a port.	

Name:	Dio_ConfigChannel	
Type:	Structure	
Range:	Port_Num	Member contains the ID of the Port that this channel belongs to
	Ch_Num	Member contains the ID of the Channel
Description:	Type for the definition of a Configuration of the Channels, that contain the ID of the port and the ID of the Channel	

Name:	Dio_ConfigType	
Type:	Structure	
Range:	0 --> <Number of Configure Channel>	
Description:	This structure contains Array of all parameters of Dio_ConfigChannel Structure in DIO driver.	

## DIO Function:

Function name:	Dio_Init		
Arguments:	Input	*ConfigPtr	Dio_ConfigType
		Pointer to the container	
	Output	None	None
	Input/Output	None	None
Return:	None	None	
Description:	Initializes the DIO module.		

Function name:	Dio_WriteChannel		
Arguments:	Input	ChannelId	Dio_ChannelType
		Which Pin in DIO channel	
		Level	Dio_LevelType
		Value will write in corresponding Pin	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Write Level Value in pin that chosen		

Function Name:	Dio_ReadChannel		
Arguments:	Input	ChannelId	Dio_ChannelType
		Which Pin in DIO channel	
	Output	None	
	Input/Output	None	
Return:	STD_HIGH	1	
	STD_LOW	0	
Description:	Return the Value Level of Specific Pin that chosen in Channel ID		

## Port Driver

### Port typedefs:

Name:	Port_PinType		
Type:	uint		
Range:	0 --> <number of port pins>		
Description:	Data type of the name of a port pin.		

Name:	Port_PinDirectionType		
Type:	Enumeration		
Range:	PORT_PIN_IN	0	Set port pin as an input
	PORT_PIN_OUT	1	Set port pin as ana output
Description:	Type for defining the direction of a Port Pin		

Name:	Port_PinModeType		
Type:	uint		
Range:	0 --> <number of the port pin modes>		
Description:	Different port pin modes.		

Name:	Pin_InitialValue		
Type:	uint8		
Range:	PORT_PIN_LEVEL_LOW --> 0x00		
	PORT_PIN_LEVEL_HIGH --> 0x01		
Description:	Pin Initial Level Value		

Name:	Pin_InternalResistor		
Type:	Enumeration		
Range:	OFF	0	None
	PULL_UP	1	Activation of internal pull-up
	PULL_DOWN	2	Activation of internal pull-down
Description:	Type definition for Input Pin Internal Resistor		

## Port Structures:

Name:	Port_ConfigChannel	
Type:	Structure	
Elements:	port_num	Member contains ID Port (A, B, C, D, ...)
	pin_num	Member contains ID Pin (1,2,3,4, ...)
	direction	Member contains Pin Direction
	resistor_state	Member contains Value of internal resistor
	init_Val	Member contains Initial Level Value
	mode	Member contains the mode of the port pin
Description:	Type for the definition of a Configuration of the Channels, that contain the ID of the port and the ID of the pin, the direction, the initial value, the value of input internal resistor pins and the mode of the chosen pin.	

Name:	Port_ConfigType	
Type:	Structure	
Elements:	0 --> <Number of Port Configure Channel>	
Description:	This structure contains Array of all parameters of Port_ConfigChannel Structure in PORT driver.	

## Port Function:

Function name:	Port_Init		
Arguments:	Input	*ConfigPtr	Port_ConfigType
		Pointer to the container	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Initializes the PORT module.		



## Timer Driver

### Timer typedefs:

Name:	Gpt_ChannelType
Type:	uint
Range:	0 --> <number of Timer in Module>
Description:	Numeric ID of a Timer channel.

Name:	Gpt_ValueType
Type:	unit
Range:	0 -> <number of ticks that can be counted in the selected timer>
Description:	Type for reading and setting the timer values (in number of ticks)

Name:	Gpt_ModeType		
Type:	Enumeration		
Range:	GPT_MODE_NORMAL	0	That a Normal Mode of the Timer Module
	GPT_MODE_SLEEP	1	This mode used to reduce power in Timer Module
Description:	Allows the selection of different power modes.		

Name:	Gpt_StateType		
Type:	Enumeration		
Range:	INITIALIZED	0	Initial state if the timer
	RUNNING	1	Timer in Running State
	STOPPED	2	Timer stopped from count ticks
	EXPIRED	3	Timer Finish count the Ticks
Description:	Allows the State of Timer		

Name:	Gpt_RunningMode		
Type:	Enumeration		
Range:	GPT_CONTINUOUS	0	Reach to Elapsed Value and repeat this operation
	GPT_ONE_SHOTT	1	Reach to Elapsed Value only one time
	GPT_CAPTURE	2	This mode use the timer act as a counter not Timer
Description:	Allows the Count Mode of the timer		

Name:	Gpt_InterruptOption		
Type:	Enumeration		
Range:	INTERRUPT_OFF	0	Disable Interrupt
	INTERRUPT_ON	1	Enable Interrupt
Description:	Enable OR Disable the Interrupt of the ID of Timer Channel		

## Timer Structure:

Name:	Gpt_ConfigChannel	
Type:	Structure	
Elements:	GptChannelId	ID of timer Channel
	Gpt_ChannelMode	Mode of the Channel of the timer
	Gpt_ChannelState	State of the Channel of the Timer
	GptChannelTickFrequency	Contain the Tick Value of the Channel of the Timer
	GptChannelTickValueMax	Contain maximum Tick Value of the Channel of the Timer
	Gpt_EnableChannelNotification	Enable/Disable Interrupt
	ChannelMode	Mode of the timer ( Timer / Counter )
	(*GptNotifcation)(void)	-
Description:	Type for the definition of a Configuration of the Channels, that contain all required data to initializing the GPT timer	

Name:	Gpt_ConfigType
Type:	Structure
Elements:	0 --> <Number of Timer Configure Channel>
Description:	This structure contains Array of all parameters of Gpt_ConfigChannel Structure in Timer driver.

## Timer Function:

Function Name:	Gpt_init		
Arguments:	Input	*ConfigPtr	Gpt_ConfigType
		Pointer to a selected configuration structure	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Initializes the timer module.		

Function Name:	Gpt_DisableNotification		
Arguments:	Input	Channel	Gpt_ChannelType
		Numeric identifier of the GPT channel.	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Enable the Interrupt of the ID of Timer Channel		

Function Name:	Gpt_EnableNotification		
Arguments:	Input	Channel	Gpt_ChannelType
		Numeric identifier of the GPT channel.	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Disable the Interrupt of the ID of Timer Channel		

Function Name:	Gpt_StartTimer		
Arguments:	Input	Channel	(Gpt_ChannelType
		Numeric identifier of the GPT channel.	
		Value	Gpt_ValueType
		Target time in number of ticks.	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Start the Channel of the timer		

Function name:	Gpt_StopTimer		
Arguments:	Input	Channel	Gpt_ChannelType
		Numeric identifier of the GPT channel.	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Stop the Channel of the timer		

Function name:	Gpt_GetTimerElapsed		
Arguments:	Input	Channel	Gpt_ChannelType
		Numeric identifier of the GPT channel.	
	Output	None	
	Input/Output	None	
Return:	Gpt_ValueType	0 -> <number of ticks that can be counted in the selected timer>	
Description:	Returns the time already elapsed.		

Function name:	Gpt_TimerRemaining		
Arguments:	Input	Channel	Gpt_ChannelType
		Numeric identifier of the GPT channel.	
	Output	None	
	Input/Output	None	
Return:	Gpt_ValueType	0 -> <number of ticks that can be counted in the selected timer>	
Description:	Returns the time remaining until the target time is reached.		

## Interrupt Driver

### Interrupt Typedefs

Name:	InterruptPeripheralGates
Type:	uint
Range:	0 --> <number of Exception Types >
Description:	Numeric ID of an Interrupt Peripheral Gates.

Name:	InterruptGroupPriority
Type:	uint8
Range:	0 --> 7
Description:	Set Group priority of the interrupt

Name:	InterruptSub_GroupPriority
Type:	uint8
Range:	0 --> 7
Description:	Set the Subgroup of the interrupt

Name:	IntCtrl_InterruptType		
Type:	Enumeration		
Range:	Interrupt Peripheral Gates	0 --> <138>	Interrupt of all Peripherals
Description:	Enable the interrupt of the peripheral		

## Interrupt Structure:

Name:	IntCtrl_ConfigChannel	
Type:	Structure	
Elements:	IntCtrl_Interrupt_Number	Numeric ID of an Interrupt Peripheral
	Group Priority	Set Group priority of the interrupt
	subgroup Priority	Set the Subgroup of the interrupt
Description:	Type for the definition of a Configuration of the Channels, that contain all required data to initializing the Interrupt Driver	

Name:	IntCtrl_ConfigType	
Type:	Structure	
Elements:	0 --> <Number of Interrupt Configure Channel>	
Description:	This structure contains Array of all parameters of IntCtrl_ConfigChannel Structure in Interrupt driver.	

## Interrupt Function:

Function name:	IntCtrl_init		
Arguments:	Input	*ConfigPtr	IntCtrl_ConfigType
		Pointer to a selected configuration structure	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Initializes the Interrupt module.		

## CAN Driver

### CAN Typedefs

Name:	Can_IdType		
Type:	uint		
Range:	0 --> <0x7FF>	Set 11 bits (Standard)	
	0 --> <0x1FFFFFFF>	Set 29 bits (Extended)	
Description:	Represents the Identifier of an L-PDU. For extended IDs the most significant bit is set.		

Name:	Can_StateTransitionType		
Type:	Enumeration		
Range:	CAN_T_START	0	CAN controller request state be STARTED.
	CAN_T_STOP	1	CAN controller request state be STOPPED.
	CAN_T_SLEEP	2	CAN controller request state be STLEEP.
	CAN_T_WAKEUP	3	CAN controller request state be WAKEUP.
Description:	State of the Can controller		

Name:	Can_ReturnType		
Type:	Enumeration		
Range:	CAN_OK	0	Return success
	CAN_NOT_OK	1	Return not success cause of error occurred
	CAN_BUSY	2	transmitter or receiver wasn't available
Description:	Return values of CAN driver API.		

### CAN Structure:

Name:	Can_ControllerBaudrateConfigType	
Type:	Structure	
Elements:	CanControllerBaudRate	Specifies the baud rate of the controller in kbps from 0 to 2000
	CanControllerSeg1	Specifies phase segment 1 in time
	CanControllerSeg2	Specifies phase segment 2 in time
	CanControllerSeg3	Specifies phase segment 3 in time
Description:	This is the type of the external data structure containing the bit timing related initialization data for one CAN controller. The contents of the initialization data structure are CAN hardware specific.	

Name:	Can_PduType		
Type:	Structure		
Elements:	swPduHandle		-
	length		-
	id		-
	sdu		-
Description:	This type is used to provide ID, DLC and SDU from CAN interface to CAN driver.		

Name:	can_ConfigType1		
Type:	Structure		
Elements:	0 --> <Number of Baud rate Configure Channel>		
Description:	This structure contains Array of all parameters of Can_ControllerBaudrateConfigType Structure in Timer driver.		

Name:	can_ConfigType2		
Type:	Structure		
Elements:	0 --> <Number of PduType Configure Channel>		
Description:	This structure contains Array of all parameters of Can_PduType Structure in Timer driver.		

## CAN Functions:

Function name:	CanBaudRate_init		
Arguments:	Input	*ConfigPtr1	can_ConfigType1
		Pointer to a selected Baud rate configuration structure	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	brief This function initializes the module.		



Function name:	CanPduType_init		
Arguments:	Input	*ConfigPtr2	can_ConfigType2
		Pointer to a selected Pdu Type configuration structure	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	brief This function initializes the module.		

Function name:	Can_SetBaudrate		
Arguments:	Input	Controller	uint8
		ID controller (CAN0 ,CAN1 ,...)	
		CanControllerBaudRate	uint16
		sets Baudrate	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	brief This function sets Baudrate the module.		

Function name:	CAN_SendData		
Arguments:	Input	Data	uint16
		Data required to be sent	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Send Data from the CAN controller.		

Function name:	CAN_ReceiveData		
Arguments:	Input	None	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Receive Data from the CAN controller.		

## ECU2 Components

1. Buzzer
2. Lights (Right, left)

### Buzzer

#### Buzzer Function

Function name:	Buzzer_init	
Arguments:	Input	None
	Output	None
	Input/Output	None
Return:	None	
Description:	Initializes the Buzzer Module	

Function name:	Buzzer_OFF		
Arguments:	Input	PinId	Dio_PinIdType
		ID of DIO PIN	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Turn off a Buzzer in specific DIO_Pin		

Function name:	Buzzer_ON		
Arguments:	Input	PinId	Dio_PinIdType
		ID of DIO PIN	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Turn on a Buzzer in specific DIO_Pin		

## Lights

### Lights Function

Function name:	RightLight_init	
Arguments:	Input	None
	Output	None
	Input/Output	None
Return:	None	
Description:	Initializes the Right Light Module	

Function name:	RightLight_OFF		
Arguments:	Input	PinId	Dio_PinIdType
		ID of DIO PIN	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Turn off a Right Light in specific DIO_Pin		

Function name:	RightLight_ON		
Arguments:	Input	PinId	Dio_PinIdType
		ID of DIO PIN	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Turn on a Right Light in specific DIO_Pin		

Function name:	LeftLight_init	
Arguments:	Input	None
	Output	None
	Input/Output	None
Return:	None	
Description:	Initializes the Left Light Module	

Function name:	LeftLight_OFF		
Arguments:	Input	PinId	Dio_PinIdType
		ID of DIO PIN	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Turn off a Left Light in specific DIO_Pin		

Function name:	LeftLight_ON		
Arguments:	Input	PinId	Dio_PinIdType
		ID of DIO PIN	
	Output	None	
	Input/Output	None	
Return:	None		
Description:	Turn on a Left Light in specific DIO_Pin		