Full detailed APIs For Each Module as well as detailed description for each Typedef : “ECU_2”

1)DIO_Module:

```
/************************************************************************************
*   \Description :  function to set configuration for all Dio of Mcu and
                    it's Alternative Functions
*   \sync\Async  :  Synchronous
*   \Reentrancy  :  Reentrant
*   \Parameters (in) :  Pointer to Struct
*   \Parameters (out):  void
*   \Return value    :  void
************************************************************************************/

        void Dio_Init(const Dio_ConfigType* ConfigPtr );

/************************************************************************************************
*   \Description: function to read any DIO Channel which takes ID of Channel and return level
*   \sync\Async:  Synchronous
*   \Reentrancy:  Reentrant
*   \Parameters(in):   DIO Channel Type "Enum"
*   \Parameters (out): DIO_LevelType
************************************************************************************************/

      Dio_LevelType Dio_ReadChannel(Dio_ChannelType ChannelId);
```

```
/************************************************************************************
*   \Description :  function to write on any Channel"Pin" which takes Pin no. and level
*   \sync\Async  :  Synchronous
*   \Reentrancy  :  Reentrant
*   \Parameters (in): Channel ID "of type Dio_ChannelType" & DIO Level "of typeDio_LevelType"
*   \Parameters (out):  void
*   \Return value :  void
************************************************************************************/
void DIO_WriteChannel(Dio_ChannelType ChannelId , Dio_LevelType Level);
```

2)TIMER_Module:

```
/************************************************************************************
*   \Description     :  function to set Timer Configurations
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  const Gpt_ConfigType*ConfigPtr "Pointer to struct"
*   \Parameters (out):  void
************************************************************************************/
void Gpt_Init(const Gpt_ConfigType*ConfigPtr);
/************************************************************************************
*   \syntax:
*   \Description     :  function to Disable Timer Interrupts in Run Time
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Non-Reentrant
*   \Parameters (in) :  Gpt_ChannelType Channel "Timer ID "
*   \Parameters (out):  void
************************************************************************************/
void Gpt_DisableNotification(Gpt_ChannelType Channel)

/************************************************************************************
```

```
*   \Description     :  function to Enable Timer Interrupts in Run Time
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Non-Reentrant
*   \Parameters (in) :  Gpt_ChannelType Channel "Timer ID "
*   \Parameters (out):  void
*******************************************************************************/
void Gpt_EnableNotification(Gpt_ChannelType Channel);
/*******************************************************************************
*   \Description     :  function to Start Timer Count
*   \sync\Async      :  ASynchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  Gpt_ChannelType Channel "Timer ID " , Gpt_ValueType Value
                         "No.of Counts in Tick"
*   \Parameters (out):  void
*******************************************************************************/
void Gpt_StartTimer(Gpt_ChannelType Channel , Gpt_ValueType Value);
/*******************************************************************************
*   \Description     :  function to Stop Timer Count
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  Gpt_ChannelType Channel "Timer ID "
*   \Parameters (out):  void
*******************************************************************************/
void Gpt_StopTimer(Gpt_ChannelType Channel);




/*******************************************************************************
```

```
*   \Description     :  function to Get Time Elapsed from Beginning of Count
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  Gpt_ChannelType Channel "Timer ID "
*   \Parameters (out):  Gpt_ValueType  "uint 32"
*******************************************************************************
Gpt_ValueType Gpt_GetTimeElapsed(Gpt_ChannelType Channel);
```

3)CAN_Module:

```
/*******************************************************************************
*   \Description     :  function to Initialize CAN Module.
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  CAN_ConfigType* Config_Ptr “Pointer to Cofiguration Struct”
*   \Parameters (out):  void
*******************************************************************************
void CAN_Init(CAN_ConfigType* Config_Ptr);

/*******************************************************************************
*   \Description     :  function to DeIntialize CAN_Module and Stop it.
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void CAN_Deinit(void);


/*******************************************************************************
*   \Description     :  function to Send Data
```

```
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Non-Reentrant
*   \Parameters (in) : Can_DataType Data “Data to be send”
*   \Parameters (out): void
*******************************************************************************
void CAN_SendData(Can_DataType Data);
/*******************************************************************************
*   \Description     :  function to Receive Data from CAN Bus
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Non_Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  Can_DataType
*******************************************************************************
CAN_DataType  CAN_ReceiveData (void);
```

4)Light:

```
/*******************************************************************************
*   \Description:  function to Intialize Light Module “RL & LL Pins Direction”
*   \sync\Async :  Synchronous
*   \Reentrancy :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void  Light_Init (void);




/*******************************************************************************
*   \Description     :  function to  Get light status
```

```
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  Light_Type light
*   \Parameters (out):  Boolean
*******************************************************************************/
Boolean Get_LightStatus(Light_Type light);

/*******************************************************************************
*   \Description     :  function to Set light status
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  Light_Type light
*   \Parameters (out):  void
*******************************************************************************/
void Set_LightStatus(Light_Type light);

/*******************************************************************************
*   \Description     :  function to Start light Timer
*   \sync\Async      :  ASynchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void Light_StartLightTimer(void);

/*******************************************************************************
*   \Description     :  function to Stop light Timer
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
```

```
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void Light_StopLightTimer(void);
```

4)Buzzer:

```
/*******************************************************************************
*   \Description:  function to Intialize Buzzer Module
*   \sync\Async :  Synchronous
*   \Reentrancy :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void Buzzer_Init (void);



/*******************************************************************************
*   \Description     :  function to Set Buzzer status
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  Buzzer_Type light
*   \Parameters (out):  void
*******************************************************************************/
void Set_BuzzerStatus(Buzzer_Type Buzzer);
```

4)Comm._Mng.:

```
/*******************************************************************************
*   \Description     :  function to Manage Interactions between App and Mcal
*   \sync\Async      :  Synchronous
```

```
*   \Reentrancy      :  Non_Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  E_Ok
*                    :  E_Nok
*******************************************************************************/
Mng_DataType CommMgr_Recieve(u8 Bus_Id ) ;
```

5)Car_Control:

```
/*******************************************************************************
*   \Description     :  function to Initialize All Car_modules
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void CarControl_Init(void);

/*******************************************************************************
*   \Description     :  function to process and control Right and Left lights
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void CarControl_lightProcessing (void);
/*******************************************************************************
*   \Description     :  function to process Receiving Data
*   \sync\Async      :  Synchronous
```

```
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void CarControl_CanReciever (void);

/*******************************************************************************
*   \Description     :  function to process and control Buzzer
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void CarControl_BuzzerProcessing (void);
```