Full detailed APIs For Each Module as well as detailed description for each Typedef : “ECU_1”

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)ICU_Module:

```
/************************************************************************************
*   \Description     :  function to set ICU Configurations
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  const Icu_ConfigType*ConfigPtr "Pointer to struct"
*   \Parameters (out):  void
************************************************************************************/
void Icu_Init(const Icu_ConfigType*ConfigPtr);


/************************************************************************************
*   \Description     :  function to Start Icu Count
*   \sync\Async      :  ASynchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  Icu_ChannelType Channel "Timer ID "
*   \Parameters (out):  void
************************************************************************************/
void Icu_StartTimer(Icu_ChannelType Channel );
```

```
/************************************************************************************
*   \Description     :  function to Stop Timer Count
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  Icu_ChannelType Channel "Timer ID "
*   \Parameters (out):  void
************************************************************************************/
void Icu_StopTimer(Icu_ChannelType Channel);

/************************************************************************************
*   \Description     :  function to Get Input Capture value.
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  Icu_ChannelType Channel "Timer ID "
*   \Parameters (out):  Icu_ValueType  "uint 32"
************************************************************************************/
Icu_ValueType Icu_getInputCaptureValue (Icu_ChannelType Channel);

/************************************************************************************
*   \syntax:
*   \Description     :  function to Set Required Edge Detection
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  Icu_EdgeType EdgeType
*   \Parameters (out):  void
************************************************************************************/
void Icu_SetEdgeDetection (Icu_EdgeType EdgeType);
```

```
/************************************************************************************
*   \Description     :  function to clear Timer Value.
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  Icu_ChannelType Channel "Timer ID "
*   \Parameters (out):  void
************************************************************************************
void Icu_ClearTimerValue(Icu_ChannelType Channel);
/************************************************************************************
*   \Description     :  function to set call back.
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) : ( void (*a_ptr)void );  //pointer to func from app.
*   \Parameters (out):  void
************************************************************************************/
void  Icu_SetCallBack ( void (*app_ptr)void );
```

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  “uint 32”
************************************************************************/
Can_DataType  Can_ReceiveData (void);
```

4)Door Sensor :

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

/************************************************************************************
*   \Description     :  function to  Get light status
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  Light_Type light
*   \Parameters (out):  Boolean
************************************************************************************/
Boolean Get_DoorSensorValue(void);
```

5)Light_Switch:

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

```
*   \Description     :  function to Get light Switch Value
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
Boolean Get_LightSwitchValue(void);
```

6)Speed_Sensor:

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


/*******************************************************************************
*   \Description     :  function to Get Speed Sensor Output
*   \sync\Async      :  ASynchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  SpeedSensor_Type
*******************************************************************************/
SpeedSensor_Type Get_SpeedSensorOutput(void);

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

```
*   \Description     :  function to Get Speed Sensor processing
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void SpeedSensor_StartSensorProcessing(void);

/*******************************************************************************
*   \Description     :  function to Stop Speed Sensor Processing
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void SpeedSensor_StopSensorProcessing(void);
```

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
*******************************************************************************/
ErrorStatus CommMgr_Send(u8 Bus_Id , Mng_DataType Data);
```

5)Car_Status:

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

/************************************************************************
*   \Description     :  function to Process LightSwitch Modules
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
************************************************************************/
void CarStatus_LightSwitchProcessing(void);

/************************************************************************
*   \Description     :  function to process Speed Sensor
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
************************************************************************/
void CarStatus_SpeedProcessing (void);



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

```
*   \Description     :  function to process DoorSensor
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void CarStatus_DoorProcessing (void);

/*******************************************************************************
*   \Description     :  function to Send Car Status
*   \sync\Async      :  Synchronous
*   \Reentrancy      :  Reentrant
*   \Parameters (in) :  void
*   \Parameters (out):  void
*******************************************************************************/
void CarStatus_CanTransmitter (void);
```