ECU2

HALL:

MCAL:

CAN

Description: module to deal with Timer

APIs

- **CANInit(void)**
  Syntax: void CANInit(void)
  Description: Initialize CAN depending on the configration in CAN_config.h
  Sync\Async: Synchronous
  Reentrancy: Non Reentrant
  Parameters (in): None
  Parameters (out): None
  Return value: void
- **CANSetCallBack(void (*CallBack)(void))**
  Syntax: void CANSetCallBack(void (*CallBack)(void))
  Description: Set the CallBack function to call it in the ISR
  Sync\Async: Synchronous
  Reentrancy: Non Reentrant
  Parameters (in): CallBack pointer to the callback function
  Parameters (out): None
  Return value: void

GPIO

Description: module to deal with general purpose input output

typedefs

- **enum PORT_type : a value to specify which port to deal with**
- **enum Direction_type: to determine the direction output or input**
- **struct PIN_config: to contain elements that describe the pin configuration (port, pin number, direction, pullup/pulldown,..)**

APIs

- **InitPORT(PORT_type thePort, Direction_type theDir,..)**
  Syntax: void InitPORT(PORT_type thePort, Direction_type theDir,..)
  Description: Initialize specific port to specific configrations
  Sync\Async: Synchronous
  Reentrancy: Reentrant
  Parameters (in): parameterName Parameter Describtion
  thePort pick the specific port needed to be intialized
  theDir set the direction of the port
  .. any other config. like PULLUP/PullDown can be passed as parameter
  Parameters (out): None
  Return value: void
- **InitPin(Pin_config myPin)**
  Syntax: void InitPin(Pin_config myPin)
  Description: Initialize specific pin to specific configurations
  Sync\Async: Synchronous
  Reentrancy: Reentrant
  Parameters (in): parameterName Parameter Describtion
  myPin struct contain all configrations of the pin as element
  Parameters (out): None
  Return value: void
- **ReadPort(PORT_type myPORT myPort)**
  Syntax: int ReadPort(PORT_type myPort)
  Description: Read port value
  Sync\Async: Synchronous
  Reentrancy: Reentrant
  Parameters (in): myPort The port to be read
  Parameters (out): None
  Return value: int the value on the port
- **_WritePort(PORT_type myPort,int value)**
  Syntax: void WritePort(PORT_type myPort,int value)
  Description: write port value
  Sync\Async: Synchronous
  Reentrancy: Non Reentrant
  Parameters (in): myPort The port to write on
  value the value to be written
  Parameters (out): None
  Return value: void
  Return value: void

- **ReadPIN(PORT_type thePort ,int PinNum)**
  Syntax: int ReadPIN(PORT_type thePort ,int PinNum)
  Description: Read port value
  Sync\Async: Synchronous
  Reentrancy: Reentrant
  Parameters (in): thePort The port of the pin to be read
  PinNum the pin number
  Parameters (out): None
  Return value: int the value on the pin
- **_WritePIN(PORT_type thePort,int PinNum,int value)**
  Syntax: void WritePIN(PORT_type thePort,int PinNum,int value)
  Description: write pin value
  Sync\Async: Synchronous
  Reentrancy: Non Reentrant
  Parameters (in): thePort The port of the pin to write on
  PinNum the pin number
  value the value to be writtenParameters (out): None
  Parameters (out): None
  Return value: void

On Board:

Light

Description: module to deal with light Device

APIs

- **LightInit(Light_type myDevice)**
  Syntax: void LightInit(Light_type myDevice)
  Description: Initialize Light depending on the configration in the struct myDevice
  Sync\Async: Synchronous
  Reentrancy: Reentrant
  Parameters (in): myDevice struct contain the pins that the sensor connected to
  Parameters (out): None
  Return value: void
- **LightWrite(Light_type myDevice, LightValue myVal)**
  Syntax: void LightWrite(Light_type myDevice ,LightVlaue myVal)
  Description: Write to the device
  Sync\Async: Synchronous
  Reentrancy: Reentrant
  Parameters (in): myDevice struct contain the pins that the device connected to
  myVal value to write to the device
  Parameters (out): None
  Return value: void

Buzzer

Description: module to deal with Buzzer Device

APIs

- **BuzzerInit(Buzzer_type myDevice)**
  Syntax: void BuzzerInit(Buzzer_type myDevice)
  Description: Initialize Buzzer depending on the configration in the struct myDevice
  Sync\Async: Synchronous
  Reentrancy: Reentrant
  Parameters (in): myDevice struct contain the pins that the sensor connected to
  Parameters (out): None
  Return value: void
- **BuzzerWrite(Buzzer_type myDevice, BuzzerValue myVal)**
  Syntax: void BuzzerWrite(Buzzer_type myDevice ,BuzzerVlaue myVal)
  Description: Write to the device
  Sync\Async: Synchronous
  Reentrancy: Reentrant
  Parameters (in): myDevice struct contain the pins that the device connected to
  myVal value to write to the device
  Parameters (out): None
  Return value: void