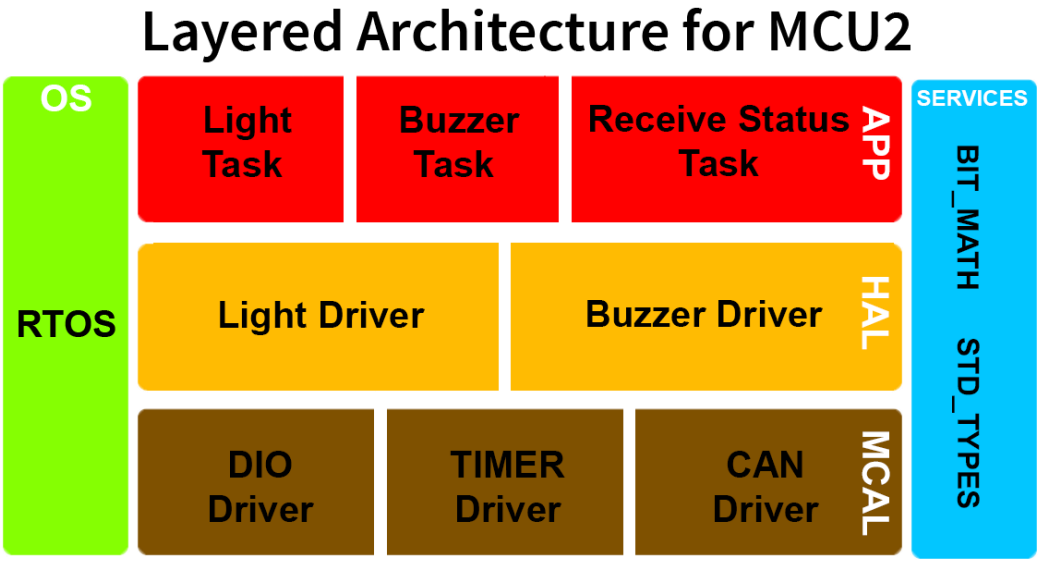


ECU1 Components

1. MCAL

- **DIO Driver:**
  Driver interface with ECU DIO peripheral which interduce DIO_Set, DIO_Reset, DIO_Toggle, DIO_Read APIs to upper layers.

- **TIMER Driver:**
  Driver interface with ECU TIMER peripheral which interduce TIMER_Init, TIMERDelay_ms, Interrupts set call back APIs to upper layers.

- **CAN Driver:**
  Driver interface with ECU CAN peripheral which interduce CAN_Init, CAN_Read, CAN_Write APIs to upper layers

2. HAL

- **Light Driver:**
  Module responsible for interface with right light and left light providing API LightOn and LightOff to control he car lights.

- **Buzzer Driver**:
- Module responsible for interface with Buzzer providing API BuzzerOn and BuzzerOff to control he car Buzzer.

3. SERVICES LAYER

- **BIT_MATH:**
  A header file contains functions like macros which perform bit manipulation SET_BIT, RESET_BIT, TOGGLE_BIT, GETVAL_BIT.

- **STD_TYPES:**
  A header file contains standard types u8, u16, u32, u64, s8, s16, s32, s64, f32, f64.

4. APPLICATION LAYER

The system has 3 tasks and 1 queue

- **Light task:**
  The task is responsible for controlling the car lights on or off.

- **Buzzer task**:
  The task is responsible for controlling the car Buzzer on or off.

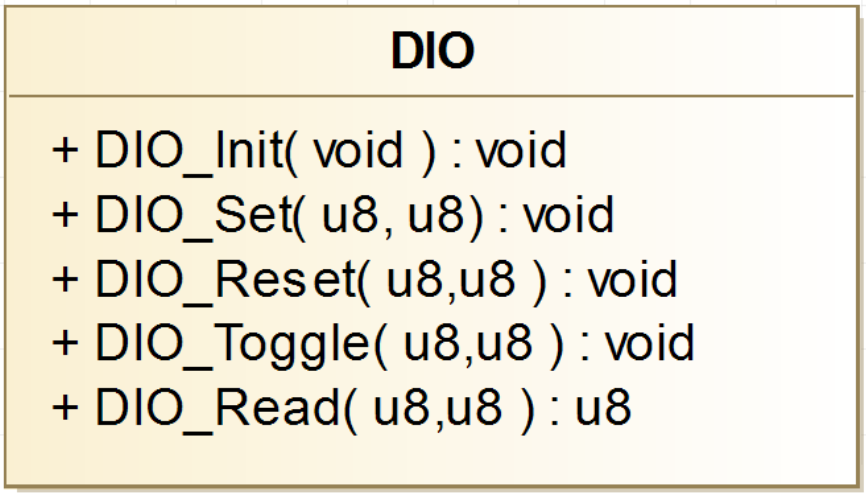
- **Receive status task**:
  The task is responsible for receive the system status sent from MCU1 through CAN bus.

- **Status queue**:
  Intercommunication to sync status messages form Receive status task to the tasks**.**

ECU2 APIs Discerption

1. MCAL

- **DIO Driver:**



DIO_Init

Description: This API use to initialize DIO ports directions

Arguments: void

Return: void

DIO_Set

Description: This API use to set the value of a DIO bin

Arguments: DIO_PORT,BIN_NUM

Return: void

DIO_Reset

Description: This API use to Reset the value of a DIO bin

Arguments: DIO_PORT,BIN_NUM

Return: void

DIO_Toggle

Description: This API use to Toggle the value of a DIO bin

Arguments: DIO_PORT,BIN_NUM

Return: void

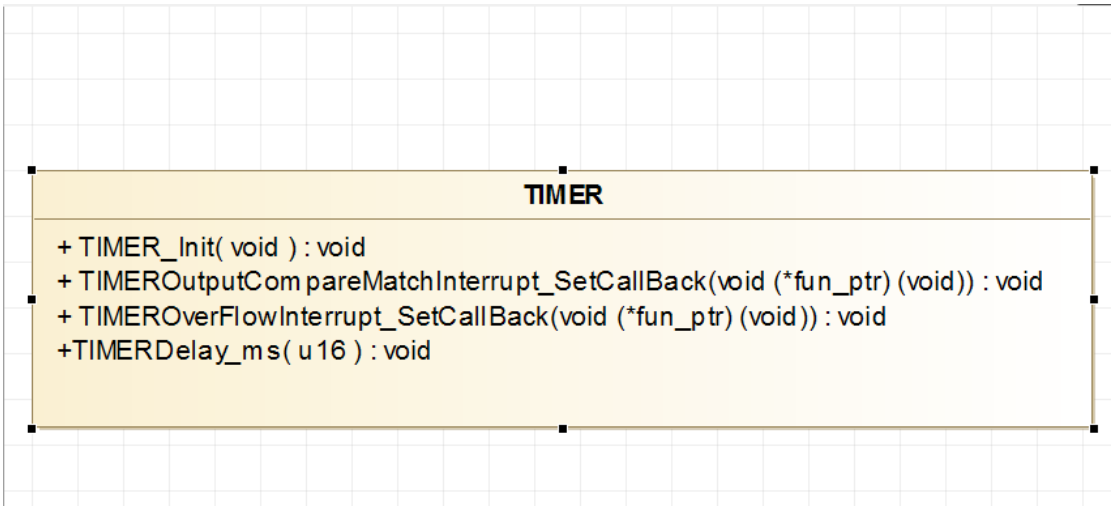
DIO_Read

Description: This API use to Read the value of a DIO bin

Arguments: DIO_PORT,BIN_NUM

Return: Boolean 0 or 1

- **TIMER Driver:**



TIMER_Init

Description: This API use to initialize TIMER tick time
Arguments: void
Return: void

TIMEROutpoutCompareMatchInterrupt_SetCallBack

Description: This API use to assign the address of timer compare match interrupt handler in his right place in vector table
Arguments: pointer to timer compare match interrupt handler
Return: void

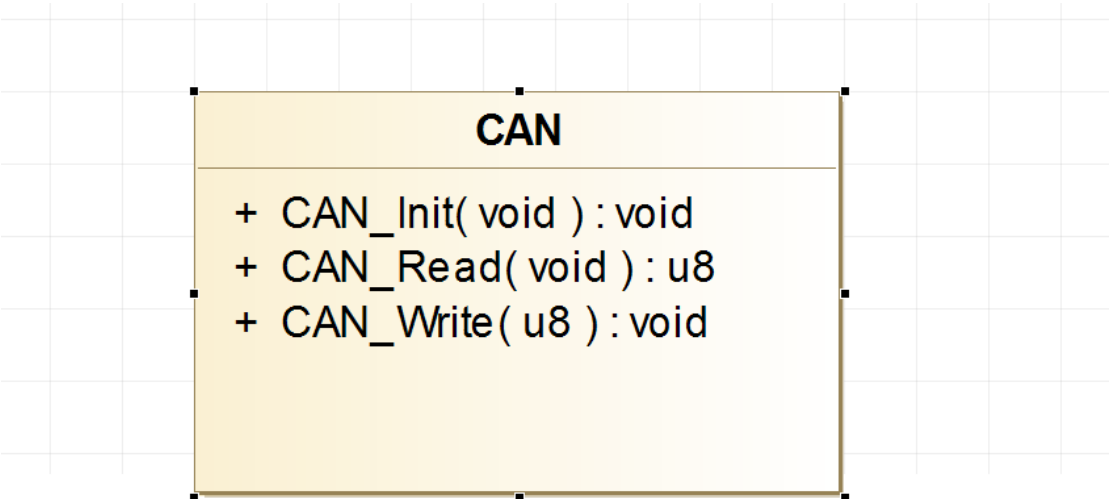
TIMEROverFlowInterrupt_SetCallBack

Description: This API use to assign the address of timer over flow interrupt handler in his right place in vector table
Arguments: pointer to timer over flow interrupt handler
Return: void

TIMERDelay_ms

Description: This API use to a delay using timer in milliseconds
Arguments: u16 which represent the desired delay in milliseconds
Return: void

- **CAN Driver:**



CAN_Init

Description: This API use to initialize CAN

Arguments: void

Return: void

CAN_Read

Description: This API use to read data sent through CAN bus

Arguments: void

Return: u8 data

CAN_Write

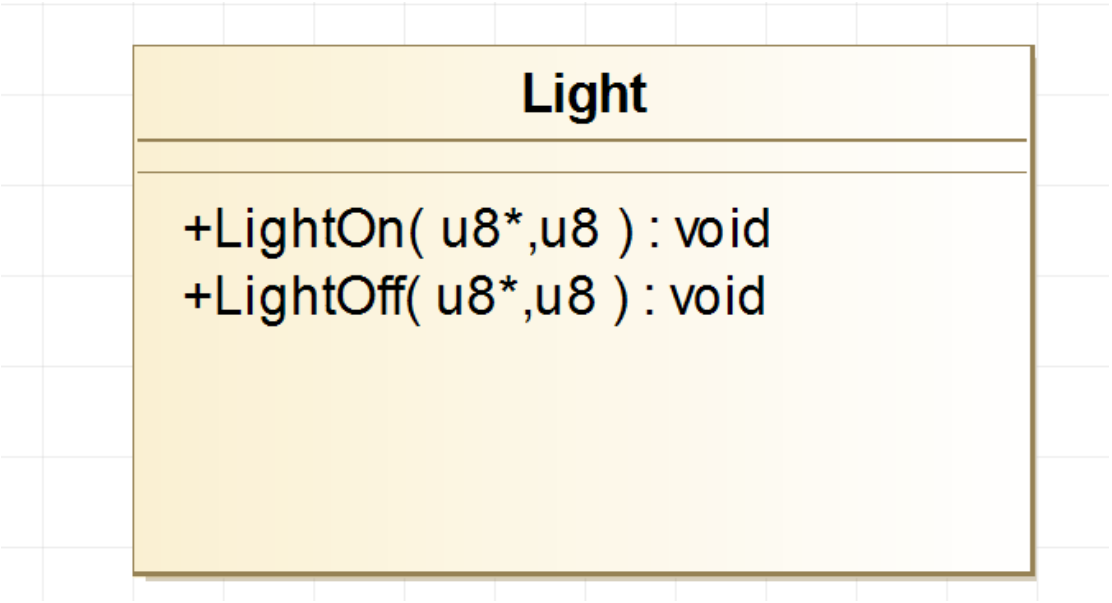
Description: This API use to write data through CAN bus

Arguments: u8 data

Return: void

2. HAL

- **Light Driver:**



LightOn

Description: This API use to turn on the right and the left lights of the car.

Arguments: first argument reg_name which is the DIO port that the car lights connected to it.

which represent a pointer to the address of the DIO register.

Second argument bit_num which represent the bit number in the same register**.**

Return: void

LightOff

Description: This API use to turn off the right and the left lights of the car.

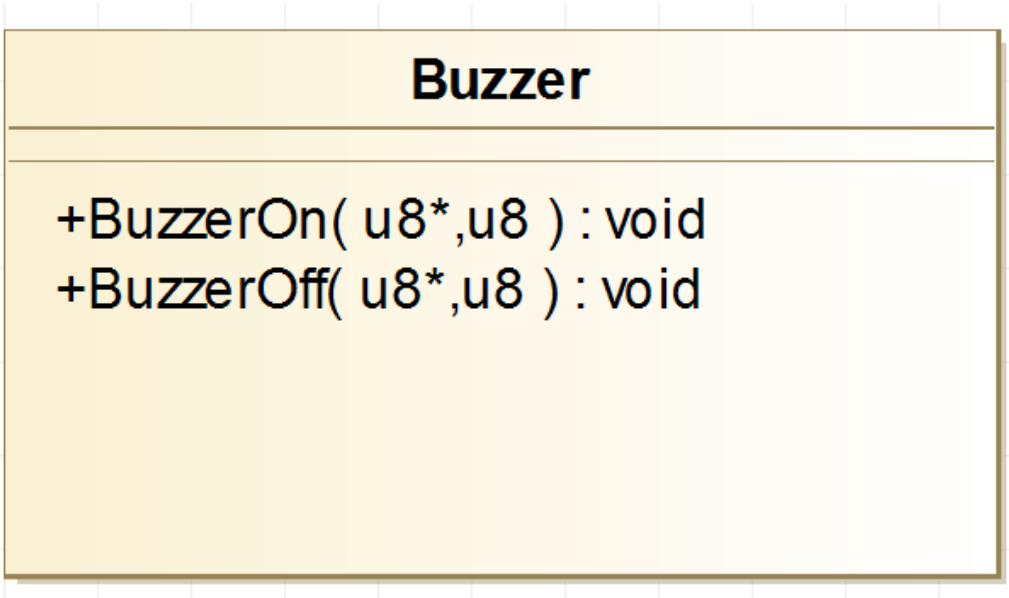
Arguments: first argument reg_name which is the DIO port that the car lights connected to it.

which represent a pointer to the address of the DIO register.

Second argument bit_num which represent the bit number in the same register**.**

Return: void

- **Buzzer Driver:**



BuzzerOn

Description: This API use to turn on the Buzzer of the car.

Arguments: first argument reg_name which is the DIO port that the car Buzzer connected to it.

which represent a pointer to the address of the DIO register.

Second argument bit_num which represent the bit number in the same register**.**

Return: void

BuzzerOff

Description: This API use to turn off the Buzzer of the car.

Arguments: first argument reg_name which is the DIO port that the car Buzzer connected to it.

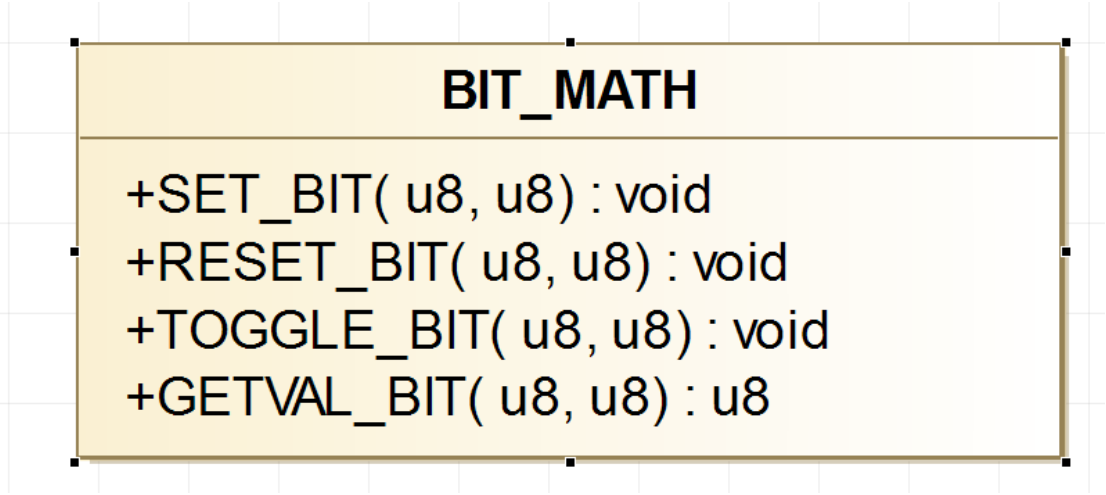
which represent a pointer to the address of the DIO register.

Second argument bit_num which represent the bit number in the same register**.**

Return: void

3. SERVICES Layer

- **BIT_MATH:**



SET_BIT

Description: This is a function like macro which be replaced with reg_name|=(1<<bit_num) to set the value of a DIO bin

Arguments: reg_name which represent a pointer to the address of the register
bit_num which represent the bit number in the same register

Return: void

RESET_BIT

Description: This is a function like macro which be replaced with reg_name&=~(1<<bit_num) to reset the value of a DIO bin

Arguments: reg_name which represent a pointer to the address of the register

bit_num which represent the bit number in the same register

Return: void

TOGGLE_BIT

Description: This is a function like macro which be replaced with reg_name^=(1<<bit_num) to toggle the value of a DIO bin

Arguments: reg_name which represent a pointer to the address of the register
bit_num which represent the bit number in the same register

Return: void

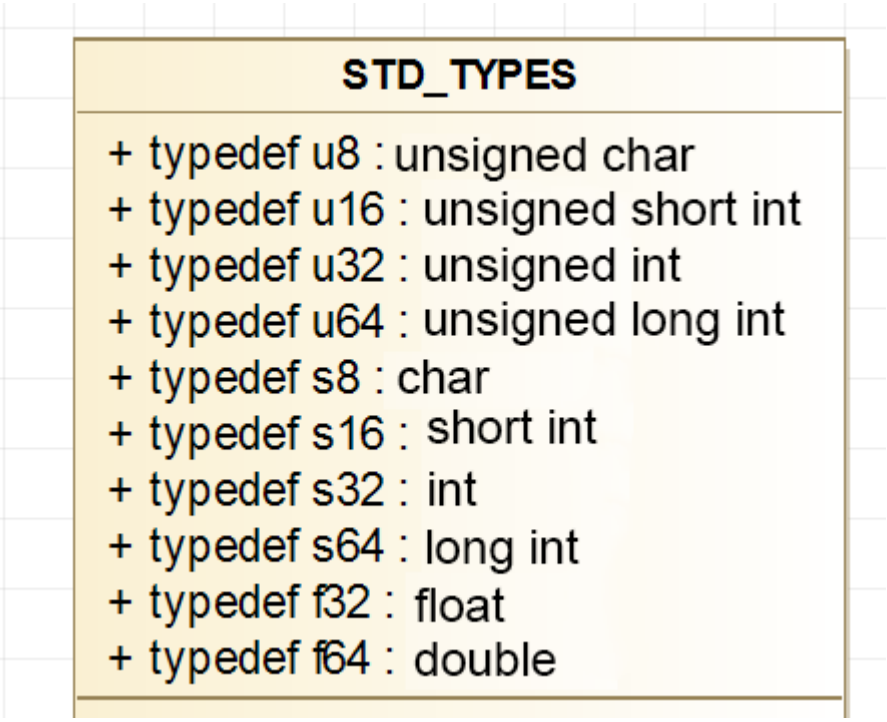
GITVAL_BIT

Description: This is a function like macro which be replaced with (reg_name>>bit_num)&1 to get the value of a DIO bin

Arguments: reg_name which represent a pointer to the address of the register
bit_num which represent the bit number in the same register

Return: u8 which represent the bin value

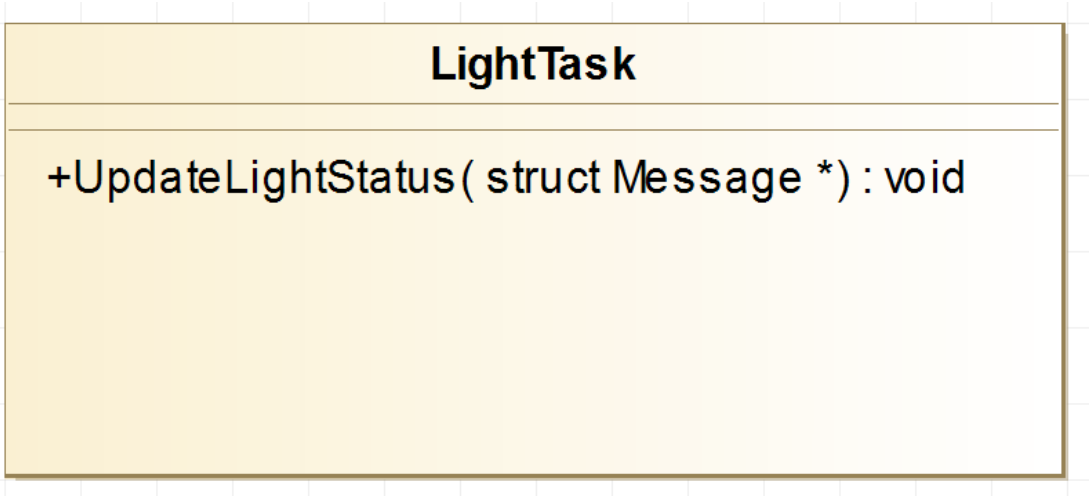
- **STD_TYPES:**



Description: A header file contains standard types u8, u16, u32, u64, s8, s16, s32, s64, f32, f64.

4. APPLICATION Layer

- **LightTask:**



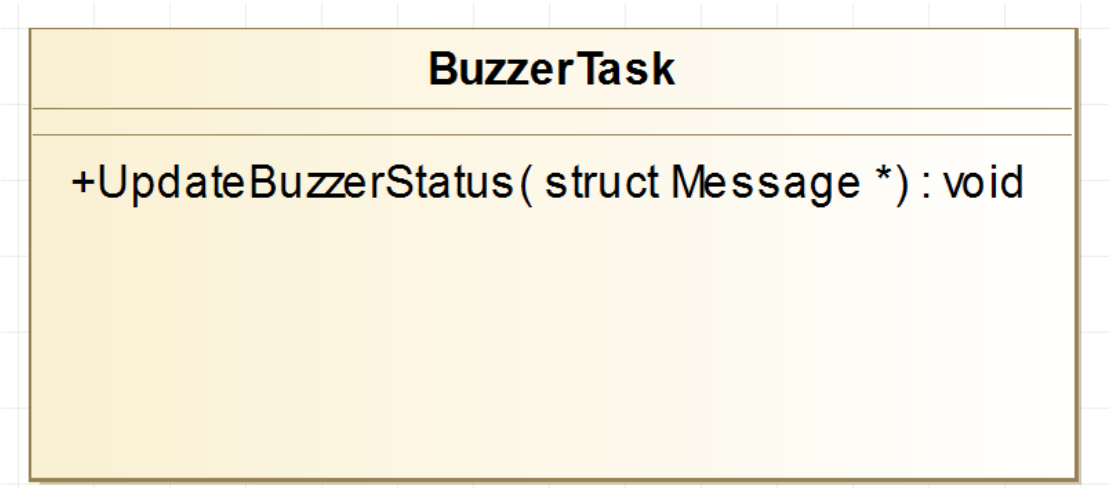
UpdateLightStatus

Description: This is a function use to control the behavior of the right and the left lights based on sensors readings sent form MCU1

Arguments: pointer to struct Message

Return: void

- **BuzzerTask:**



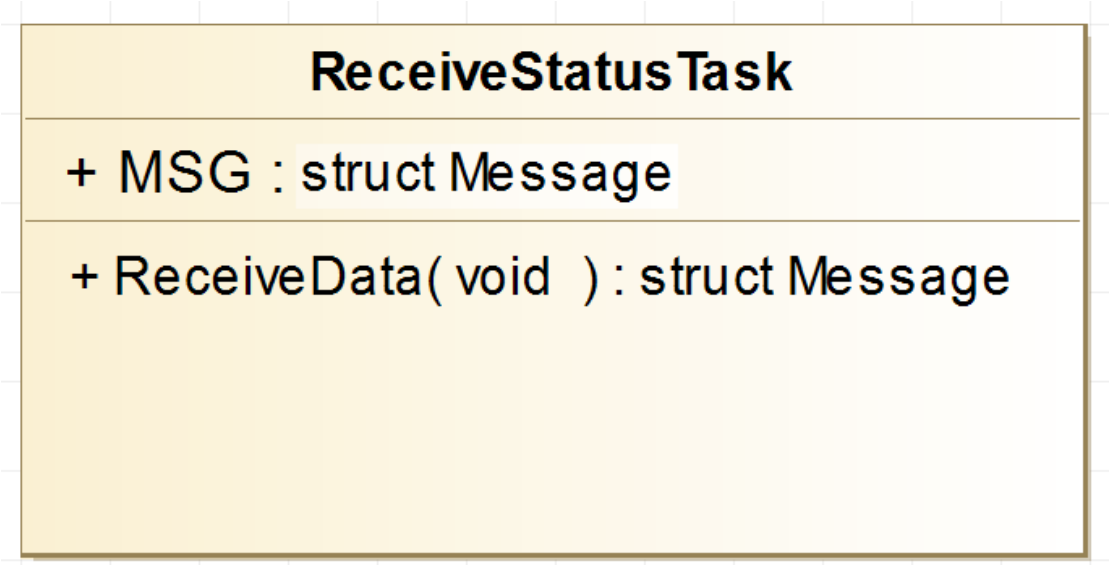
UpdateBuzzerStatus

Description: This is a function use to control the behavior of the Buzzer based on sensors readings sent form MCU1

Arguments: pointer to struct Message

Return: void

- **ReceiveStatusTask:**



ReceiveData

Description: This is a function use to receive the reads of car sensors sent from MCU1 through CAN bus

Arguments: void

Return: struct Message

MSG

Description: a struct variable contains the read of a car sensor sent from MCU1

ECU2 Class diagram