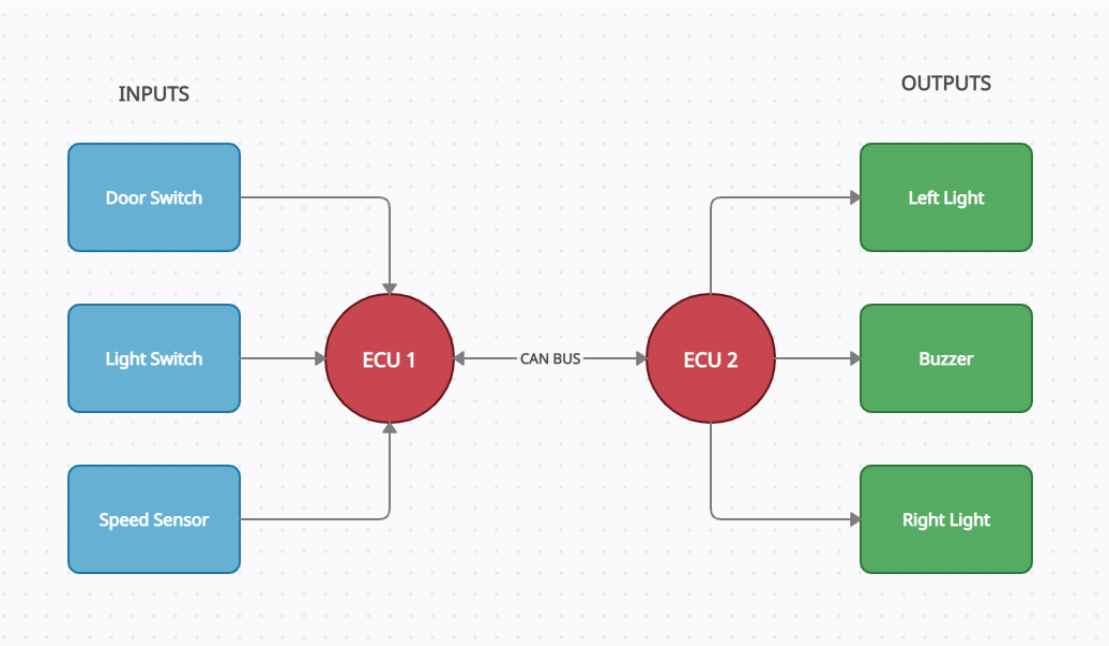
egFWD Embedded Systems Advanced Track

AUTOMOTIVE DOOR CONTROL SYSTEM DESIGN

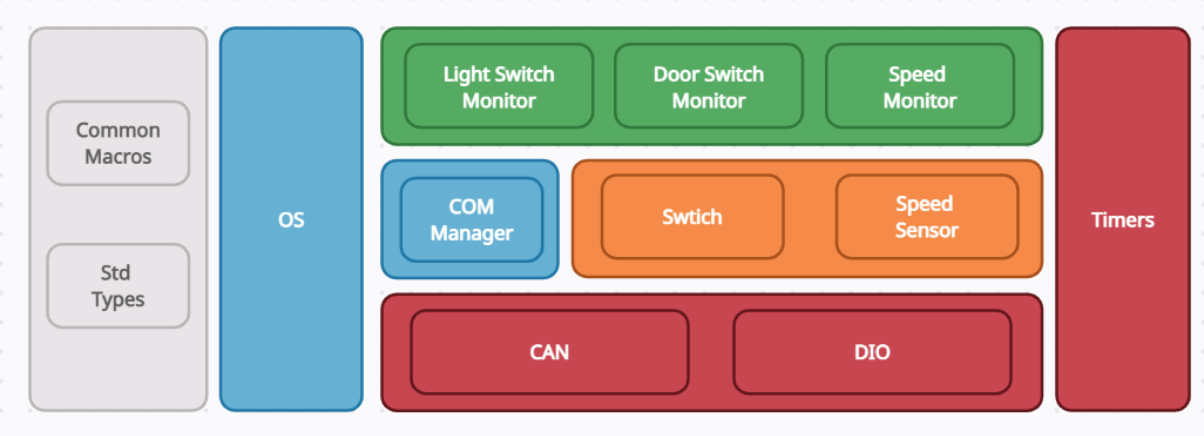
Static Design

Eng. Omar Alaa
06-Dec-2022

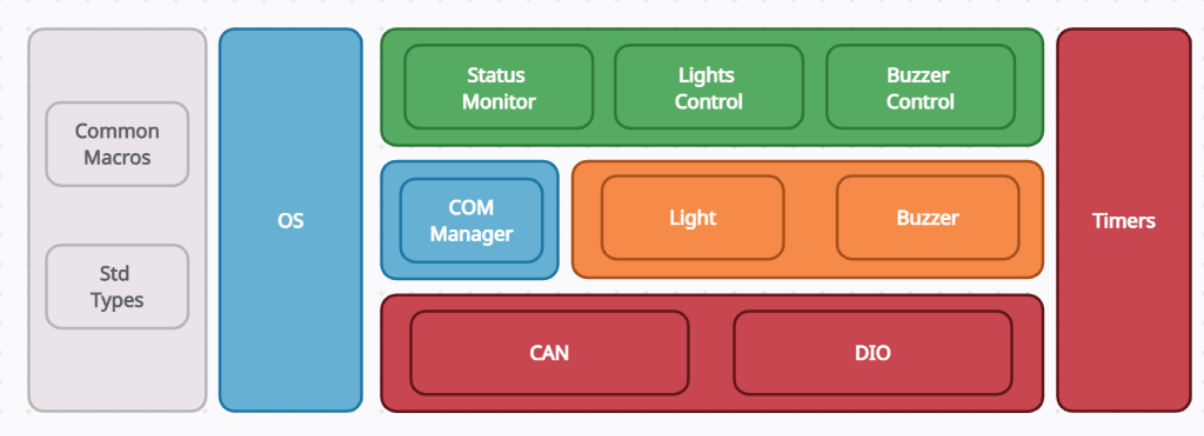
Block diagram



ECU 1 layers



ECU 2 layers



Types & APIs

- **Common:**
  - **Return_t: Enum**
    - Enumeration of all return conditions

- **DIO:**
  - **Types:**
    - **DioCfg_t: Struct**
      - Configuration structure
    - **Port_t: Enum**
      - Enumeration of all available ports
    - **Pin_t: Enum**
      - Enumeration of all available pins
    - **PinState_t: Enum**
      - Enumeration of pin states (PINHIGH/PINLOW)
  - **APIs:**
    - **Void DioInit(DioCfg_t *Config)**
      - Description: Initialize DIO module
      - Parameters: DioCfg_t *Config
      - Reentrancy: Non-Reentrant
      - Synchronization: Synchronous
      - Return: void
    - **PinState_t Dio_ReadChannel(Port_t Port, Pin_t Pin)**
      - Description: Gets current pin state
      - Parameters: Port_t Port, Pin_t Pin
      - Reentrancy: Reentrant
      - Synchronization: Synchronous
      - Return: PinState_t
    - **void Dio_WriteChannel(Port_t Port, Pin_t Pin)**
      - Description: Sets pin to desired state
      - Parameters: Port_t Port, Pin_t Pin
      - Reentrancy: Reentrant
      - Synchronization: Synchronous
      - Return: void

- **Timers:**
  - **Types:**
    - **TimerCfg_t: Struct**
      - Configuration structure
    - **Timer_t: Enum**
      - Enumeration of available timer modules
  - **APIs:**
    - **Void TimerInit(TimerCfg_t *Config)**
      - Description: Initialize timer modules
      - Parameters: TimerCfg_t *Config
      - Reentrancy: Non-Reentrant
      - Synchronization: Synchronous
      - Return: void
    - **Void SetCallback(Timer_t Timer, void (*Callback)())**
      - Description: Sets callback function for timer interrupt handlers
      - Parameters: Timer_t Timer, void (*Callback)()
      - Reentrancy: Non-Reentrant
      - Synchronization: Synchronous
      - Return: void
    - **Void ReloadTimer(Timer_t Timer, uint8 millisecs)**
      - Description: Sets count value for timer
      - Parameters: Timer_t Timer, uint8 millisecs
      - Reentrancy: Reentrant
      - Synchronization: Synchronous
      - Return: void
    - **Void StartTimer(Timer_t Timer)**
      - Description: Starts the timer count
      - Parameters: Timer_t TImer
      - Reentrancy: Reentrant
      - Synchronization: Synchronous
      - Return: void
    - **Void StopTimer(Timer_t Timer)**
      - Description: Stops the timer count
      - Parameters: Timer_t Timer
      - Reentrancy: Reentrant
      - Synchronization: Synchronous
      - Return: void

- **CAN:**
  - **Types:**
    - **CanCfg_t: Struct**
      - Configuration Structure
    - **MsgID_t: Enum**
      - Enumeration of message IDs on the CAN bus
  - **APIs:**
    - **Void CanInit(CANCfg_t *Config)**
      - Description: Initialize CAN module
      - Parameters: CanCfg_t *Config
      - Reentrancy: Non-Reentrant
      - Synchronization: Synchronous
      - Return: void
    - **Return_t CanWrite(MsgID_t MsgID, uint8 *Buffer, uint8 BufferSize)**
      - Description: writes a msg on CAN bus
      - Parameters: MsgID_t MsgID, uint8 *buffer, uint8 BufferSize
      - Reentrancy: Non-Reentrant
      - Synchronization: Asynchronous
      - Return: Return_t
    - **Return_t CanRead(MsgID_t MsgID, uint8 *Buffer, uint8 BufferSize)**
      - Description: reads a msg from CAN bus
      - Parameters: MsgID_t MsgID, uint8 *buffer, uint8 BufferSize
      - Reentrancy: Non-Reentrant
      - Synchronization: Asynchronous
      - Return: Return_t
    - **void CAN_SetCallback(void(*Callback)())**
      - Description: Sets callback function for CAN interrupt handlers
      - Parameters: void (*Callback)()
      - Reentrancy: Non-Reentrant
      - Synchronization: Synchronous
      - Return: void

- **Speed Sensor:**
  - **Types:**
    - **Speed_t: uint8**
  - **APIs:**
    - **Speed_t GetSpeed()**
      - Description: Gets current vehicle speed
      - Parameters: void
      - Reentrancy: Reentrant
      - Synchronization: Asynchronous
      - Return: Speed_t

- **Switch:**
  - **Types:**
    - **Switch_t: Enum**
      - Enumeration of switch instances
  - **APIs:**
    - **State_t GetSwtichState(Switch_t Switch)**
      - Description: Gets current switch state
      - Parameters: Switch_t Switch
      - Reentrancy: Reentrant
      - Synchronization: synchronous
      - Return: State_t

- **Light:**
  - **Types:**
    - **Light_t: Enum**
      - Enumeration of Light instances
  - **APIs:**
    - **Void LightON(Light_t Light)**
      - Description: Turn on the desired light
      - Parameters: Light_t Light
      - Reentrancy: Reentrant
      - Synchronization: synchronous
      - Return: void
    - **Void LightOFF(Light_t Light)**
      - Description: Turn off the desired light
      - Parameters: Light_t Light
      - Reentrancy: Reentrant
      - Synchronization: synchronous
      - Return: void

- **Buzzer:**
  - **APIs:**
    - **Void BuzzerON()**
      - Description: Turn on the buzzer
      - Parameters: void
      - Reentrancy: Reentrant
      - Synchronization: synchronous
      - Return: void

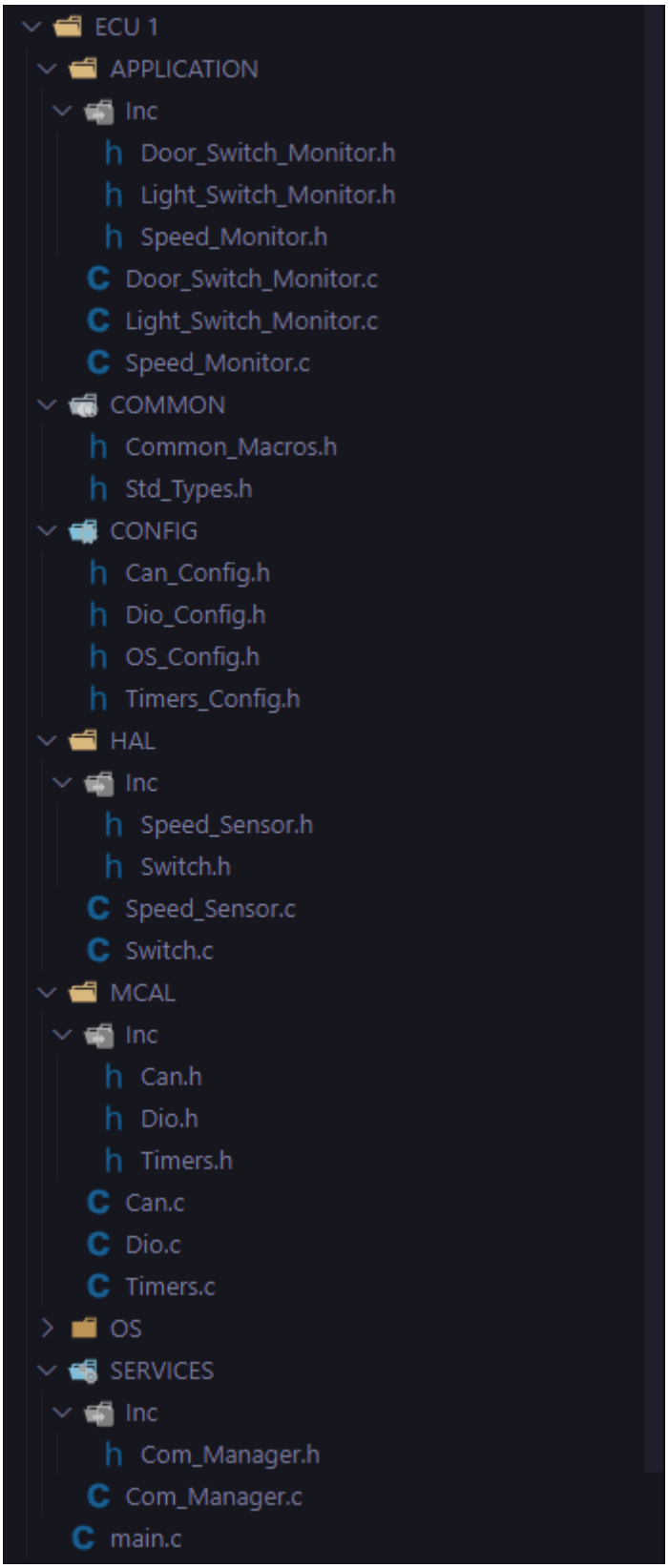
    - **Void BuzzerOFF()**
      - Description: Turn off the buzzer
      - Parameters: void
      - Reentrancy: Reentrant
      - Synchronization: synchronous
      - Return: void

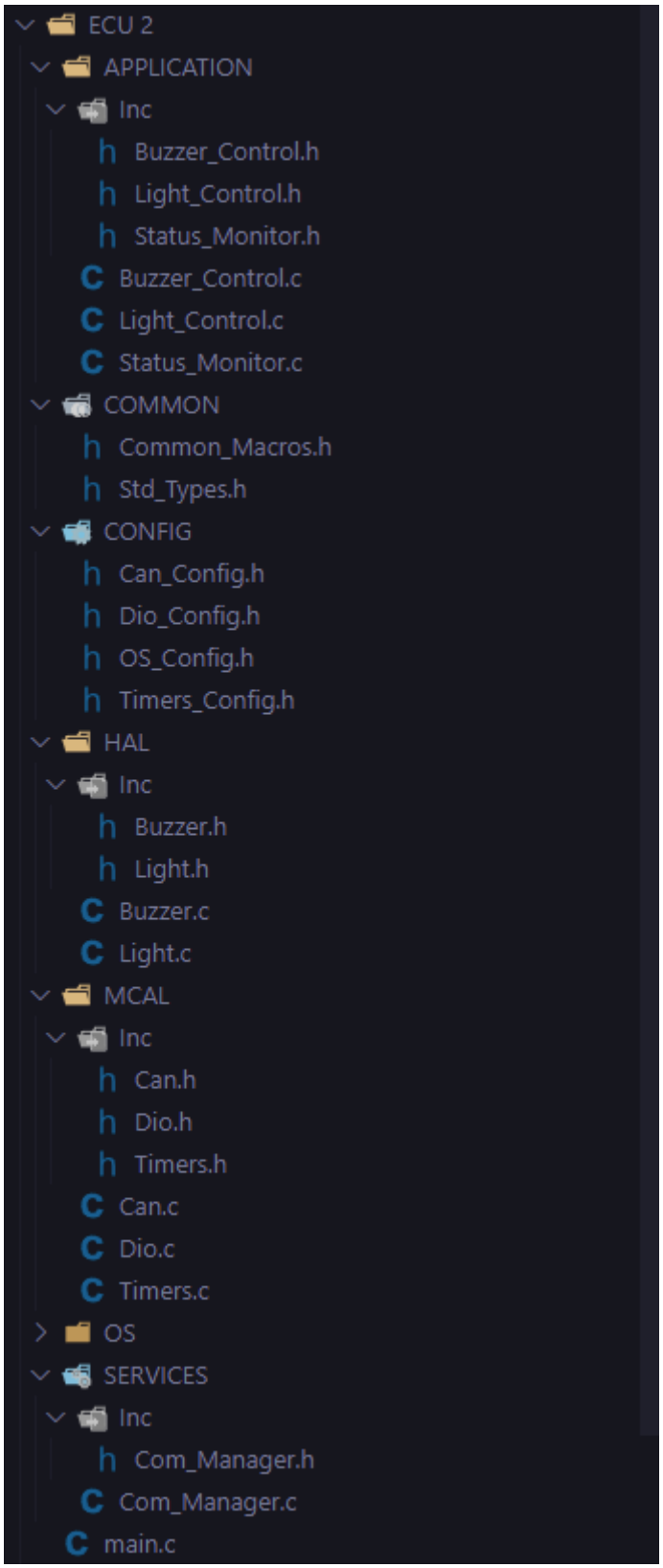
- **COM Manager:**
  - **Types:**
    - **Device_t: Enum**
      - Enumeration of available devices on communication buses

  - **APIs:**
    - **Return_t SendMsg(Device_t Device, uint8 *Buffer, uint8 BufferSize)**
      - Description: Routes the msg to the desired device
      - Parameters: Device_t Device, uint8 *Buffer, uint8 BufferSize
      - Reentrancy: Non-Reentrant
      - Synchronization: synchronous
      - Return: Return_t

    - **Return_t ReceiveMsg(Device_t Device, uint8 *Buffer, uint8 BufferSize)**
      - Description: Routes the msg to the desired device
      - Parameters: Device_t Device, uint8 *Buffer, uint8 BufferSize
      - Reentrancy: Non-Reentrant
      - Synchronization: synchronous
      - Return: Return_t

Folder structure




ECU 2
APPLICATION
Inc
Buzzer_Control.h
Light_Control.h
Status_Monitor.h
Buzzer_Control.c
Light_Control.c
Status_Monitor.c
COMMON
Common_Macros.h
Std_Types.h
CONFIG
Can_Config.h
Dio_Config.h
OS_Config.h
Timers_Config.h
HAL
Inc
Buzzer.h
Light.h
Buzzer.c
Light.c
MCAL
Inc
Can.h
Dio.h
Timers.h
Can.c
Dio.c
Timers.c
OS
SERVICES
Inc
Com_Manager.h
Com_Manager.c
main.c