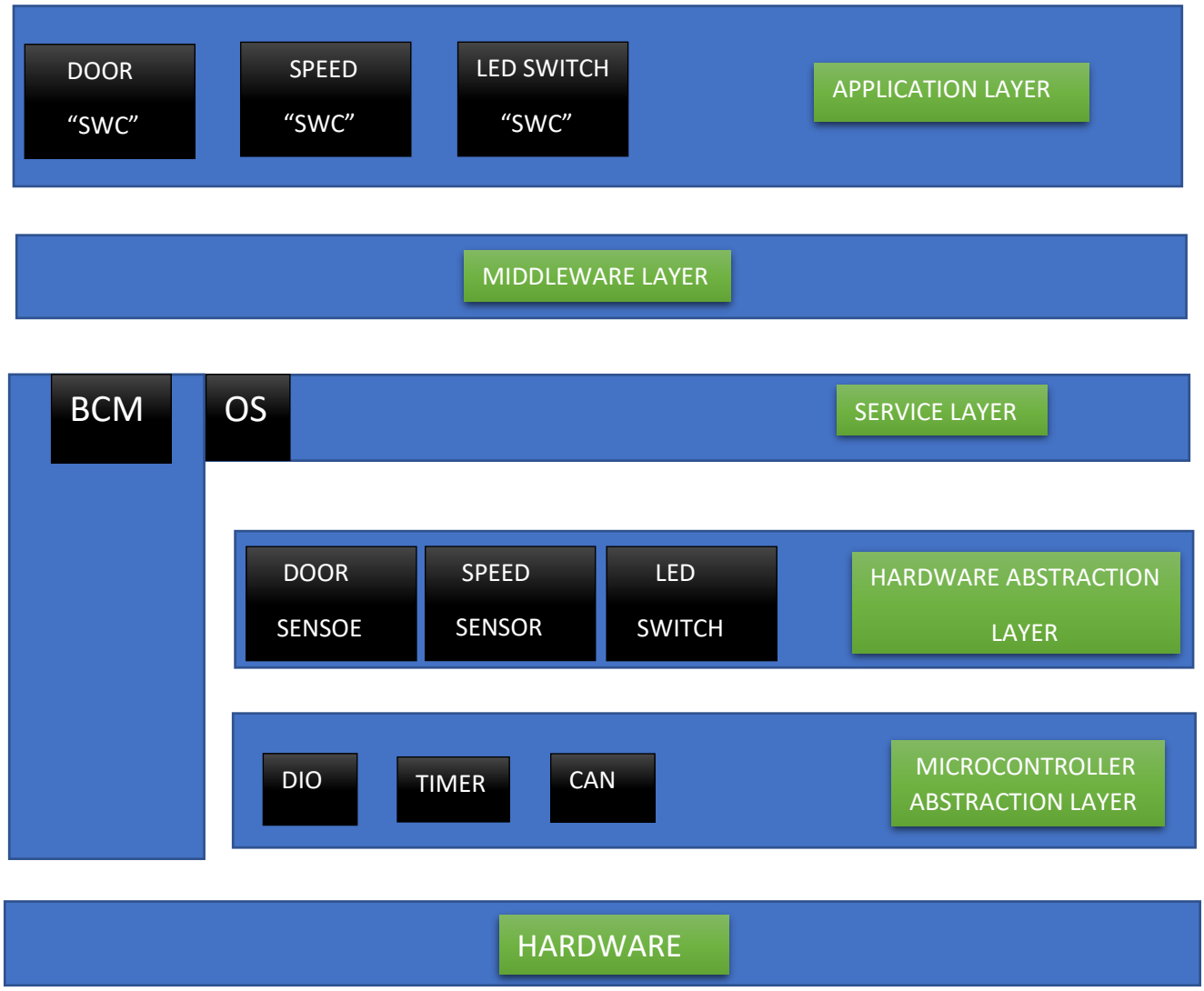
ECU_1

1- The Layered Architecture:



2- ECU_1 Components and Modules:

- ECU_1 consists of 3 components: Door SWC, Speed SWC, Led Switch SWC.
- It also has an operating system which will be used for scheduling the periodicity of sending the status of each of the three components.
- the Basic Communication Module (BCM) which is used to send the status of the three modules to the other side.

- It has a CAN drive as it will send messages through the CAN bus to the other side.

3- APIs and Typedefs for each Module:

*API:

Door Status Is Door Open (void).

- Description: This function gets the status of the door whether it's open or closed.
- Input parameters: None
- Return Value: Door Status

  - Typedef: Boolean Door Status.
  - Typedef: char Boolean.

*Speed:

API: Speed Status (void).

- Description: This function shows the value of the car's speed.
- Input Parameters: void
- Return: Speed Status.

  - Typedef: uint8_t Speed Status.
  - Typedef: char uint8_t.

* Led Switch:

API: Led Switch Status (void).

- Description: This function gets the status of the car's Led Switch whether it is pressed or not.
- Input Parameters: None
- Return: Led Switch Status.

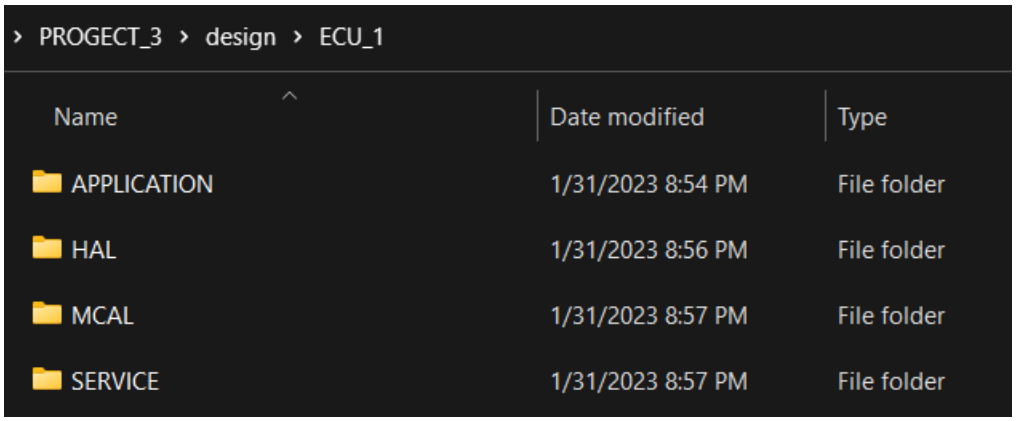
  - Typedef: Boolean Led Switch Status.
  - Typedef: uint8_t Boolean.
  - Typedef: char uint8_t.

* Operating System:
API: void Start Scheduler ().

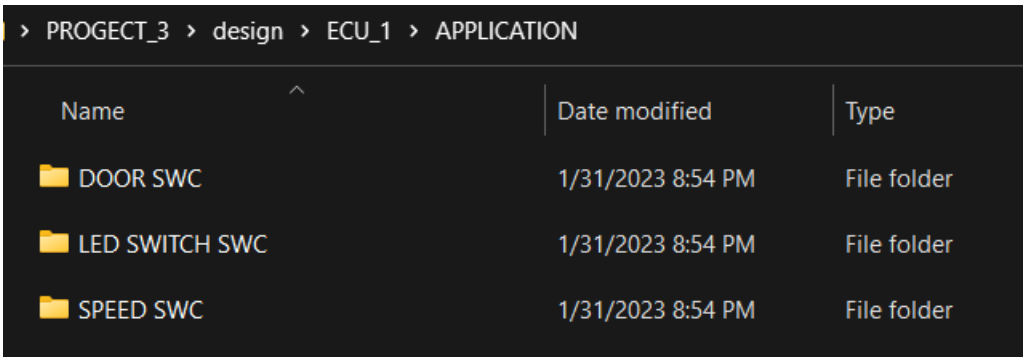
- Description: This function starts the scheduler of the system which handles the tasks of the system.
- Input Parameters: None
- Return value: void

4- Folders Architecture:

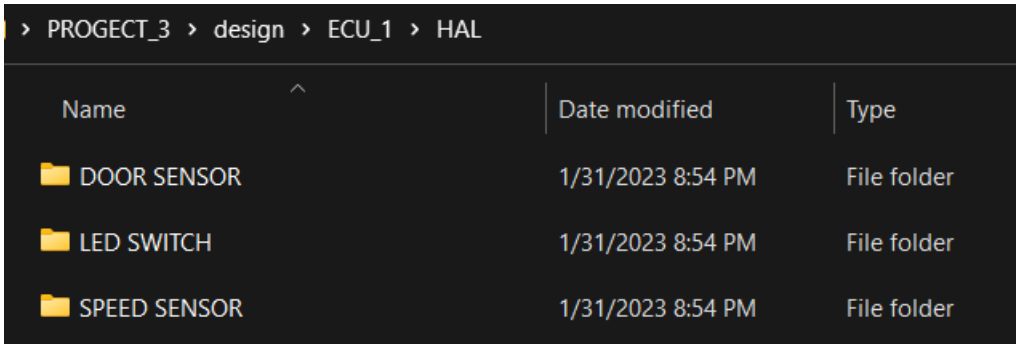
- ECU 1 Folder:



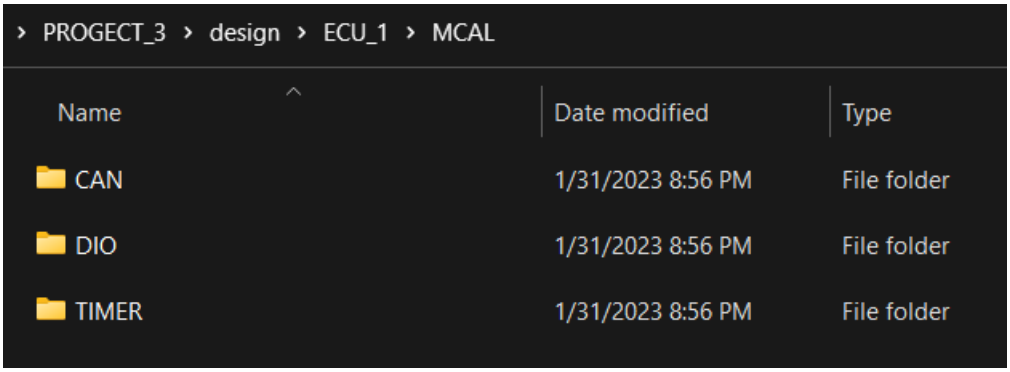
- ECU1 Application Folder:



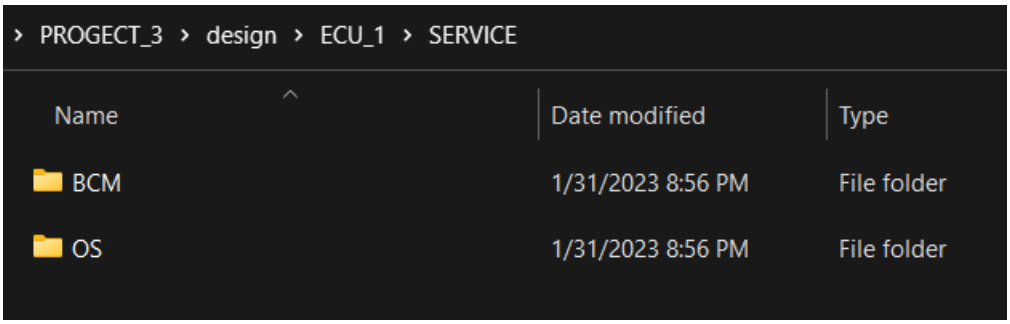
- ECU1 HAL Folder:



- ECU1 MCAL Folder:

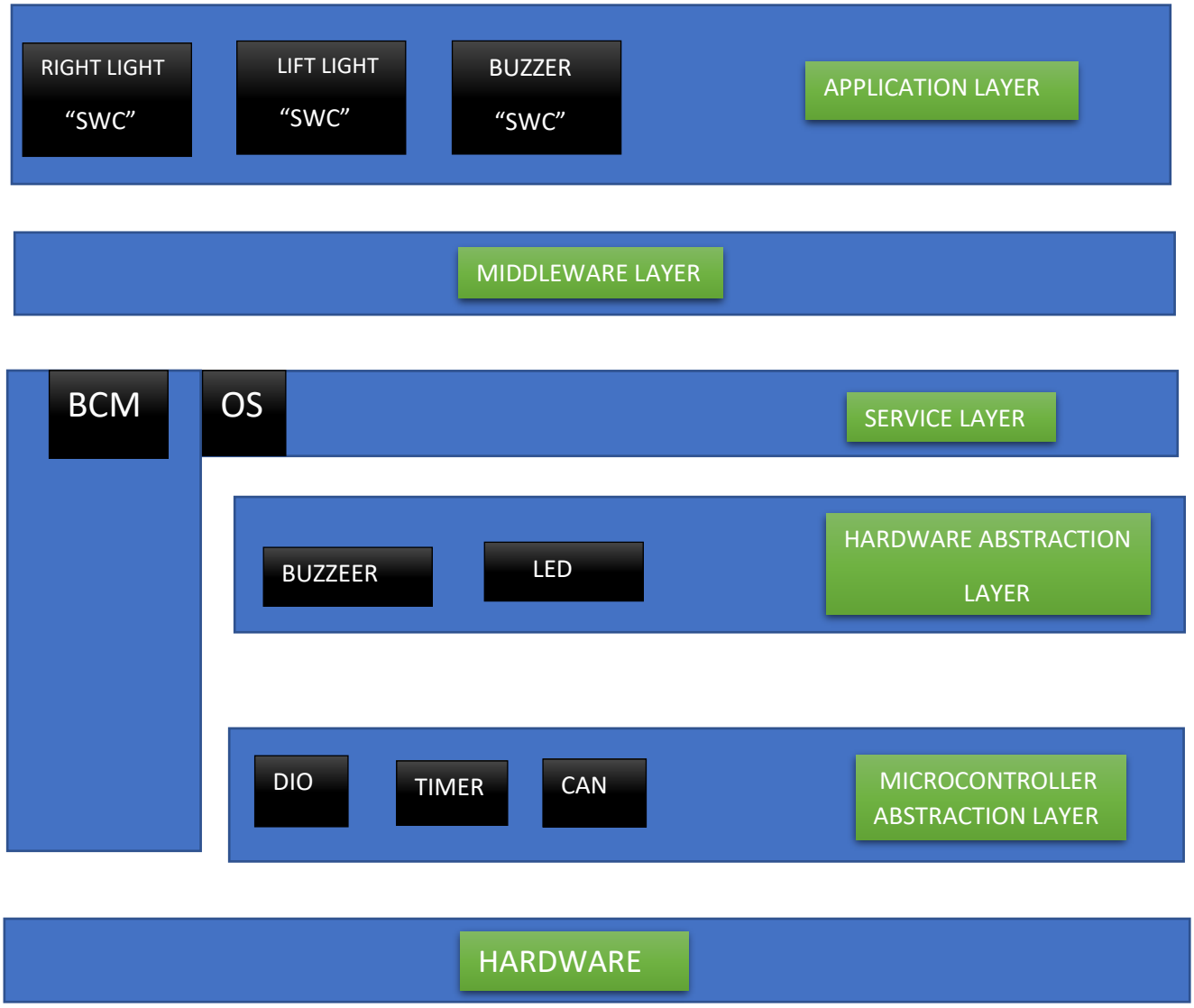


- ECU1 Services Folder:



ECU_2

1- The Layered Architecture:



2- ECU 2 Components and Modules:

- ECU2 consists of 3 main modules: Right Light Component, Left Right Component and Buzzer Component.
- It also has a Basic Communication Module (BCM) which is responsible for receiving the messages from the sending ECU.
- It has CAN driver as well which will receive the messages through the CAN bus.

3- APIs and Typedefs for each module:

- Left Light:

API: Led State Is Left Light ON (void).

- Description: This function checks if Left light led is on or off.
- Input parameters: None
- Return: Led State that can be on or off.

- API: void Left Light Led Turn On (void).
- Description: This function turns on the left light led.
- Input Parameters: None
- Return: void

- Typedef: Boolean Led State.

- Right Light: API: Led State Is Right Light ON (void).
- Description: This function checks if the right light led is on or off.
- Input Parameters: None
- Return: Led State .

- API: void Right Light Led Turn On (void).
- Description: This function turn on the right light led.
- Input parameters: None
- Return: void

- Typedef: Boolean Led State.

- Buzzer:
- API: void Buzzer Turn On (uint8_t time).

- Description: This function turn the buzzer on.
- Input parameters: time which represents the time for the buzzer to be turned on.
- Return: void

o API: void Buzzer Turn Off (void).
- Description: This function turn the buzzer off.
- Input parameters: None
- Return: void

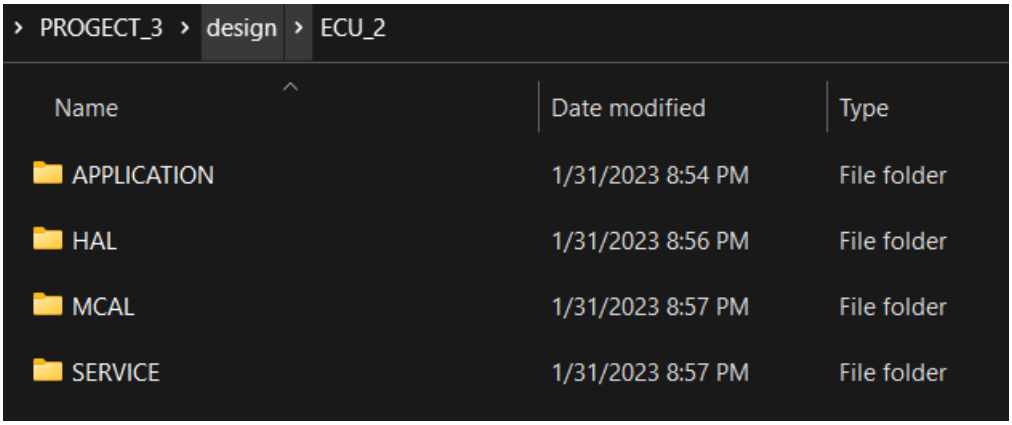
o API: Buzzer State (void).
- Description: This function checks if the buzzer is on or off.
- Input parameters: None
- Return: void

- Typedef: Boolean Buzzer State.

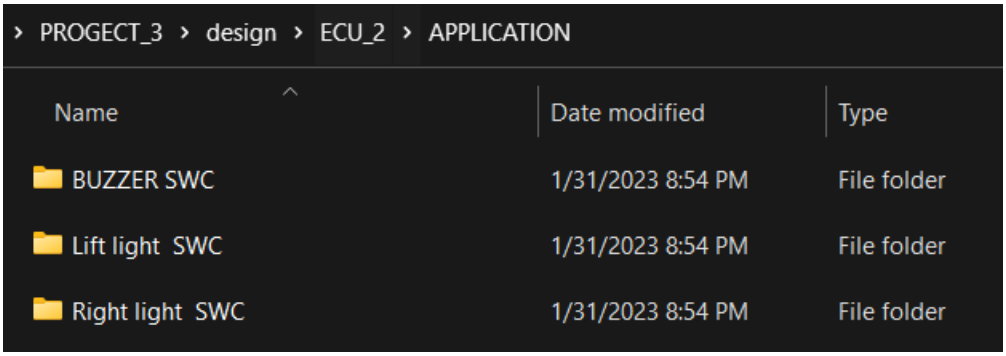
- CAN:
o API: Reception State CAN Receive Message (void).
- Description: This function receives a message from the CAN bus.
- Input Parameters: None
- Return: this function returns a value of type Reception State which marks reception ok or not ok.

4- Folders Architecture:

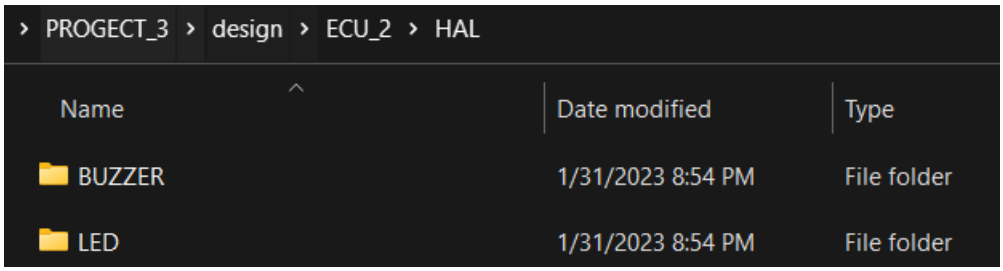
- ECU2 Folder:



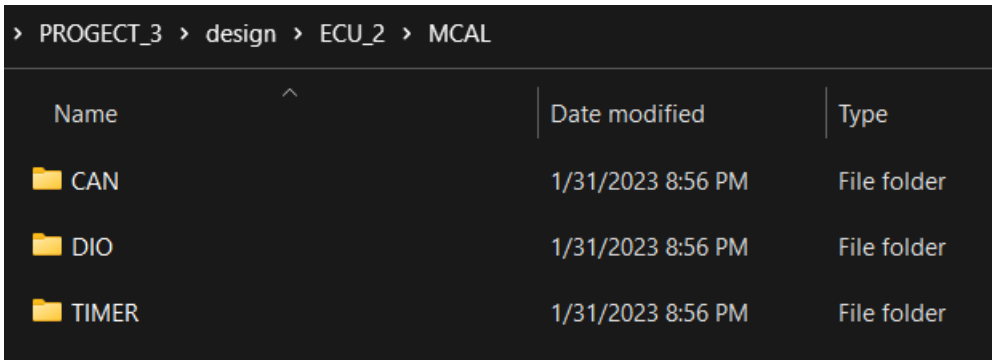
- ECU2 Application Folder:



- ECU2 HAL Folder:



- ECU2 MCAL Folder:



- ECU2 Services Folder: