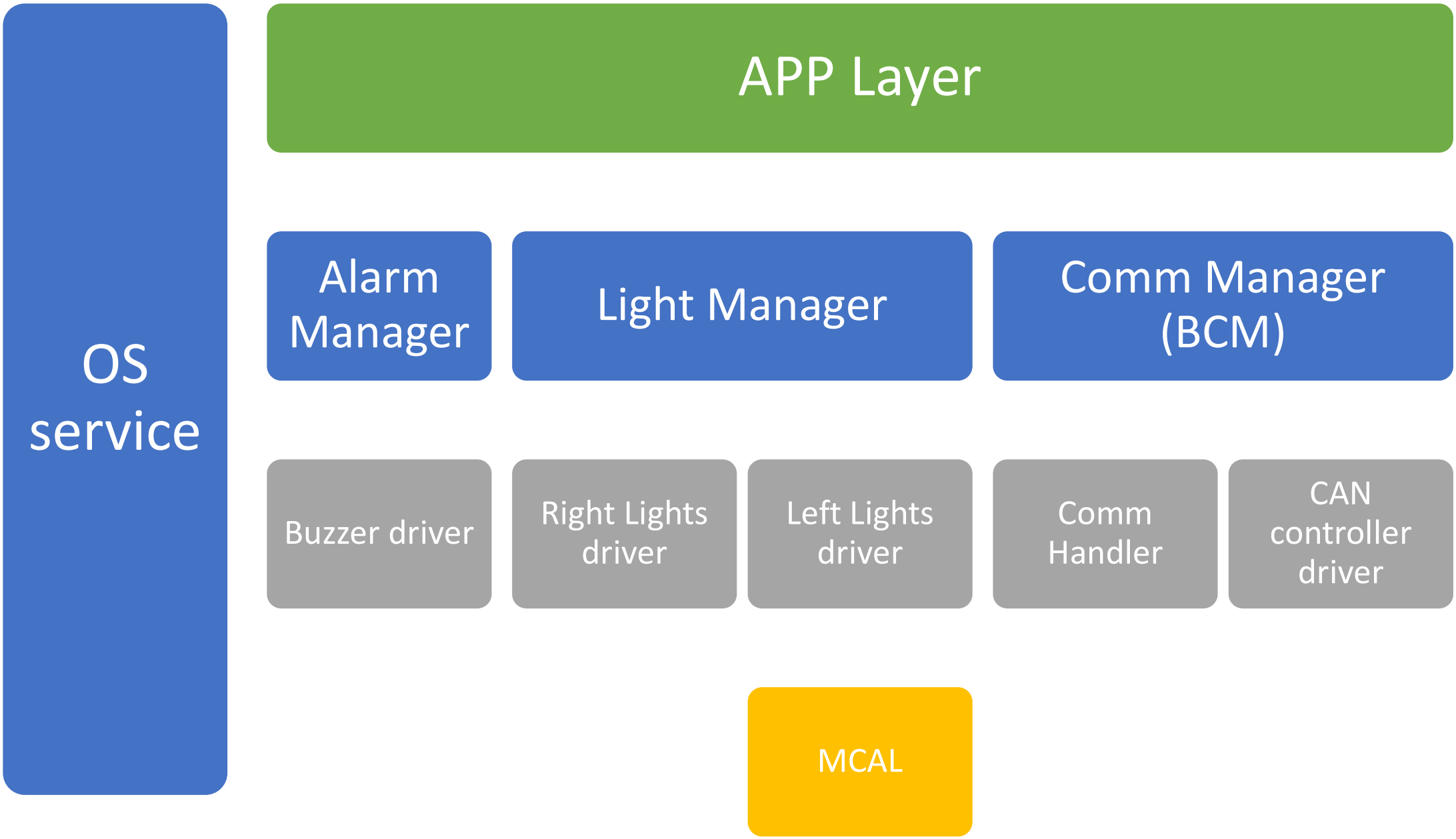

OS service
APP Layer
Alarm Manager
Light Manager
Comm Manager (BCM)
Buzzer driver
Right Lights driver
Left Lights driver
Comm Handler
CAN controller driver
MCAL

Modules :

- Alarm Manager
- Comm Manager (BCM)
- Light Manager
- Buzzer driver
- Right Lights driver
- Left Lights driver
- Comm Handler
- CAN controller driver
- DIO
- OS service

APIs description (ECU2)

- Alarm manager
  - InitAlarm(Alarm)

Arguments	**Alarm**: enum u16 to identify the id of an Alarm
Return	Status_type(OK or Er) -> enum of 0 or 1 to state the success of initialization
Type	Init
Re-entrant	Y
Asynchronous	N
Description	Initialize the needed HW and variables to access the Alarm

---

  - Get_Alarm_st(Alarm)

Arguments	**Alarm**: enum u16 to identify the id of an Alarm
Return	Digital, value u8 -> ON or OFF
Type	Getter
Re-entrant	Y
Asynchronous	N
Description	Used to read a boolean state of an ON/OFF Alarm

  - Set_Alarm_st(Alarm)

Arguments	**Alarm**: enum u16 to identify the id of an Alarm
Return	Status_type(OK or Er) -> enum of 0 or 1 to state the success of Setting
Type	Setter
Re-entrant	Y
Asynchronous	N
Description	Used to activate an ON/OFF Alarm

****

- Comm manager
  - InitComm (Protocol)

Arguments	**Protocol**: enum u16 to identify the id of the targeted communication protocol
Return	Status_type(OK or Er): enum of 0 or 1 to state the success of initialization
Type	Init
Re-entrant	Y
Asynchronous	N
Description	Initialize the needed HW to be ready for any comm

---

- Tx_msg(Protocol, TxBuffer,Len)

Arguments	-**Protocol**: enum u16 to identify the id of the targeted communication protocol -**TxBuffer**: pointer to a buffer of data to be transmitted -**Len**: buffer length
Return	Status_type(OK or Er): enum of 0 or 1 to state the success of transmitting
Type	Setter
Re-entrant	Y
Asynchronous	Y
Description	Used to send a transmission request to a targeted communication module

---

- Rx_msg(Protocol, TxBuffer,Len)

Arguments	-**Protocol**: enum u16 to identify the id of the targeted communication protocol -**TxBuffer**: pointer to a buffer of data to be transmitted -**Len**: buffer length
Return	Status_type(OK or Er): enum of 0 or 1 to state the success of receiving
Type	Getter
Re-entrant	Y
Asynchronous	Y
Description	Used to send a receive request to a targeted communication module

---

- DataIsThere(Protocol)

Arguments	-**Protocol**: enum u16 to identify the id of the targeted communication protocol
Return	Available data buffer length in the module targeted u32
Type	Getter
Re-entrant	Y
Asynchronous	N
Description	Used to check if there data ready to be received in a targeted communication module

****

- Light manager
  - LightUp (Light)

Arguments	**Light**: Light_type enum to identify the id of the targeted Light source
Return	Status_type(OK or Er): enum of 0 or 1 to state the success of Setting the light
Type	Setter
Re-entrant	Y
Asynchronous	N
Description	Used to Light up a targeted source of light

- GetLight_st(Light)

Arguments	**Light**: Light_type enum to identify the id of the targeted Light source
Return	Digital, value u8 -> High or Low
Type	Getter
Re-entrant	Y
Asynchronous	N
Description	Used to read a boolean state of a targeted source of light

****

- Buzzer driver
  - SetBuzzer_State (BuzzerId, BuzzState)

Arguments	**BuzzerId**: Buzzer_type enum to identify the id of the targeted Buzzer **BuzzState**: enum of 0 or 1 to identify the needed state of the buzzer
Return	Status_type(OK or Er): enum of 0 or 1 to state the success of Setting the buzzer
Type	Setter
Re-entrant	Y
Asynchronous	N
Description	Used to control a targeted buzzer pin

  - GetBuzzer_State (BuzzerId)

Arguments	**BuzzerId**: Buzzer_type enum to identify the id of the targeted Buzzer
Return	**BuzzState**: enum of 0 or 1 to identify the needed state of the buzzer
Type	Getter
Re-entrant	Y
Asynchronous	N
Description	Used to read a boolean state of a targeted Buzzer pin

****

- Comm Handler
  - InitComm (Protocol)

Arguments	**Protocol**: enum u16 to identify the id of the targeted communication protocol
Return	Status_type(OK or Er): enum of 0 or 1 to state the success of initialization
Type	Init
Re-entrant	Y
Asynchronous	N
Description	Initialize the needed HW (external or internal) to be ready for any comm

---

- Tx_msg(Protocol, TxBuffer,Len)

Arguments	-**Protocol**: enum u16 to identify the id of the targeted communication protocol -**TxBuffer**: pointer to a buffer of data to be transmitted -**Len**: buffer length
Return	Status_type(OK or Er): enum of 0 or 1 to state the success of transmitting
Type	Setter
Re-entrant	Y
Asynchronous	Y
Description	Used to send a transmission request to a targeted communication module

---

- Rx_msg(Protocol, TxBuffer,Len)

Arguments	-**Protocol**: enum u16 to identify the id of the targeted communication protocol -**TxBuffer**: pointer to a buffer of data to be transmitted -**Len**: buffer length
Return	Status_type(OK or Er): enum of 0 or 1 to state the success of receiving
Type	Getter
Re-entrant	Y
Asynchronous	Y
Description	Used to send a receive request to a targeted communication module

---

- DataIsThere(Protocol)

Arguments	-**Protocol**: enum u16 to identify the id of the targeted communication protocol
Return	Available data buffer length in the module targeted u32
Type	Getter
Re-entrant	Y
Asynchronous	N
Description	Used to check if there data ready to be received in a targeted communication module

****

- DIO
  - InitModule (void)

Arguments	**void**
Return	Status_type(OK or Er): enum of 0 or 1 to state the success of initialization
Type	Init
Re-entrant	Y
Asynchronous	N
Description	Initialize the needed HW to be ready for any DIO write

---

- Set_Pin (portNum,pinNum)

Arguments	**portNum**: enum of the targeted port id(u16) **pinNum**: enum of the targeted pin id (u16)
Return	Status_type(OK or Er): enum of 0 or 1 to state the success of setting the pin
Type	Setter
Re-entrant	Y
Asynchronous	N
Description	Sets the pin bit High

- Clr_Pin (portNum,pinNum)

Arguments	**portNum**: enum of the targeted port id(u16) **pinNum**: enum of the targeted pin id (u16)
Return	Status_type(OK or Er): enum of 0 or 1 to state the success of clearing the pin
Type	Setter
Re-entrant	Y
Asynchronous	N
Description	Clears the pin bit

- MaskPort (portNum,mask)

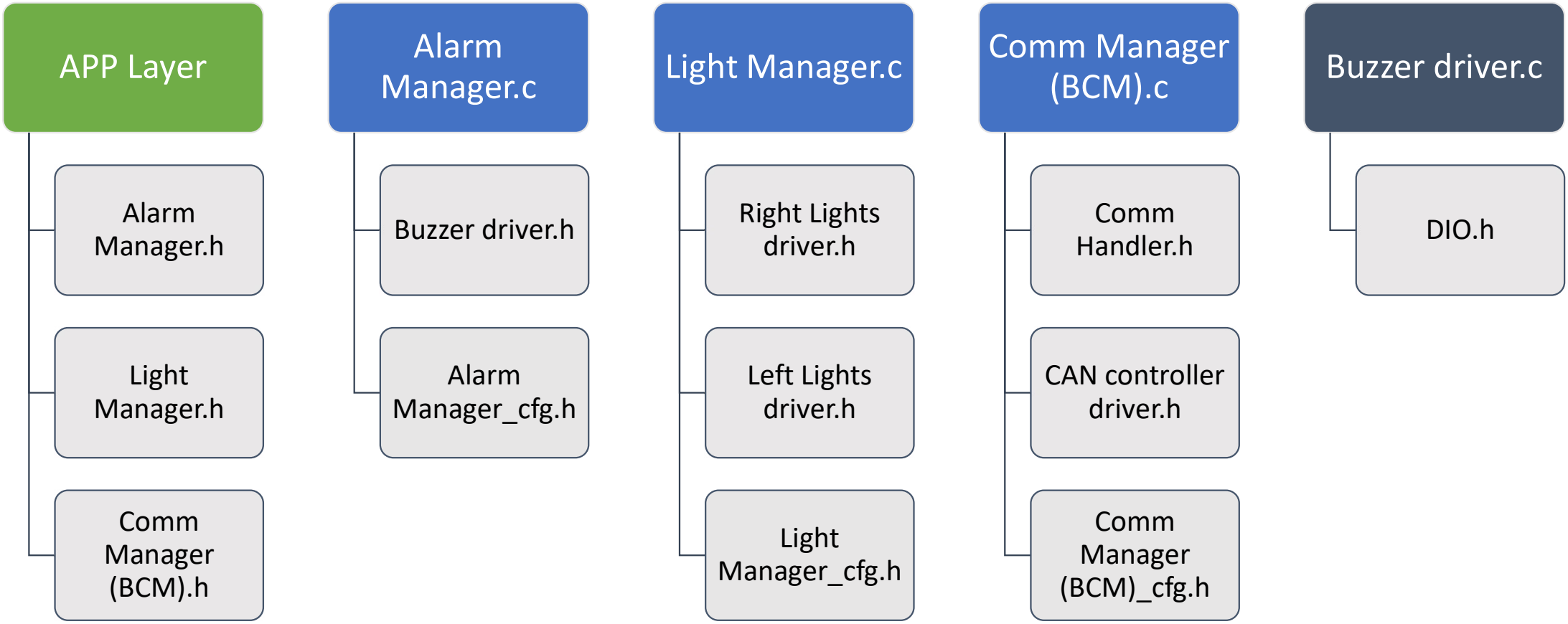
Arguments	**portNum**: enum of the targeted port id(u16) **Mask**: certain value needed to be written out on the port pins(u32)
Return	Status_type(OK or Er): enum of 0 or 1 to state the success of masking the port
Type	Setter
Re-entrant	Y
Asynchronous	N
Description	Masks the port to a certain value

- GetPin(portNum,pinNum)

Arguments	**portNum**: enum of the targeted port id(u16) **pinNum**: enum of the targeted pin id (u16)
Return	Digital, value u8 -> High or Low
Type	Getter
Re-entrant	Y
Asynchronous	N
Description	Used to read a boolean state of the targeted Pin

- GetPort(portNum)

Arguments	**portNum**: enum of the targeted port id(u16)
Return	Digital, value u32 -> 32 bit of the current state of the port pins
Type	Getter
Re-entrant	Y
Asynchronous	N
Description	Used to read a boolean state of the targeted Port


APP Layer
Alarm Manager.h
Light Manager.h
Comm Manager (BCM).h
Alarm Manager.c
Buzzer driver.h
Alarm Manager_cfg.h
Light Manager.c
Right Lights driver.h
Left Lights driver.h
Light Manager_cfg.h
Comm Manager (BCM).c
Comm Handler.h
CAN controller driver.h
Comm Manager (BCM)_cfg.h
Buzzer driver.c
DIO.h