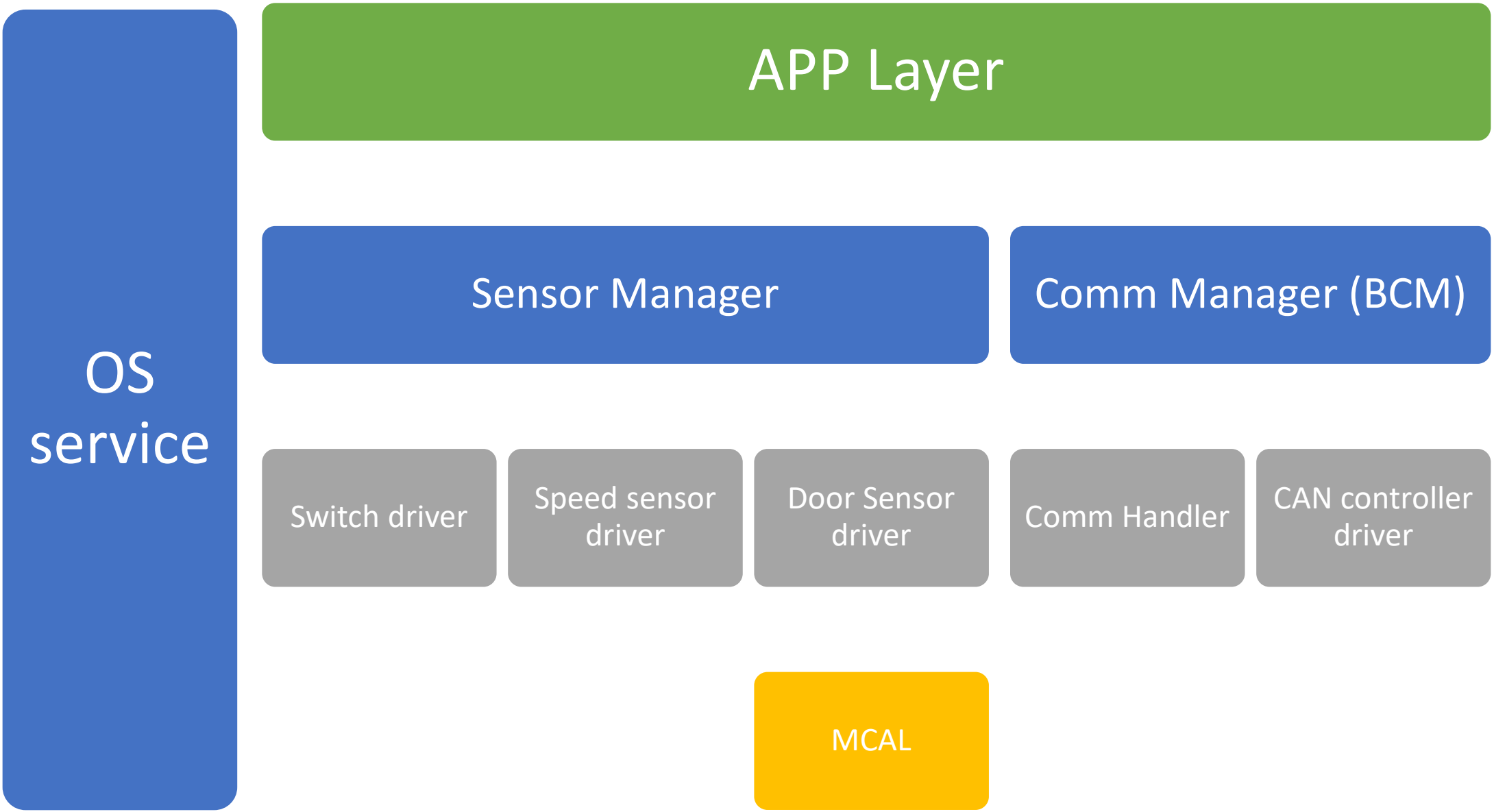
OS service
APP Layer
Sensor Manager
Comm Manager (BCM)
Switch driver
Speed sensor driver
Door Sensor driver
Comm Handler
CAN controller driver
MCAL

Modules :

- Sensor Manager
- Comm Manager (BCM)
- Switch driver
- Speed sensor driver
- Door Sensor driver
- Comm Handler
- CAN controller driver
- DIO
- OS service

APIs description (ECU1)

- Sensor manager
  - InitSensor(sensor)

Arguments	**Sensor** -> enum u16 to identify the id of a sensor
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 hold and access the sensor readings

---

  - Get_digitalSensor_st(sensor)

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

  - Get_AnlgSensor_st (sensor)

Arguments	**Sensor** -> enum to identify the id of a sensor
Return	Analog, u16 value -> to specify the current reading of an analog sensor
Type	Getter
Re-entrant	Y
Asynchronous	N
Description	Used to read analog value of multi valued sensor

****

- 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

****

- Door Sensor driver
  - Get_Door_st(void)

Arguments	**void**
Return	Digital, value u8 -> High or Low
Type	Getter
Re-entrant	Y
Asynchronous	N
Description	Used to read a boolean state of the door sensor input pin

****

- Speed Sensor driver
  - Get_Speed(void)

Arguments	**void**
Return	Digital, value u16 -> speed reading
Type	Getter
Re-entrant	Y
Asynchronous	N
Description	Used to read a analog state of the speed sensor input

****

- 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

****

Folder structure