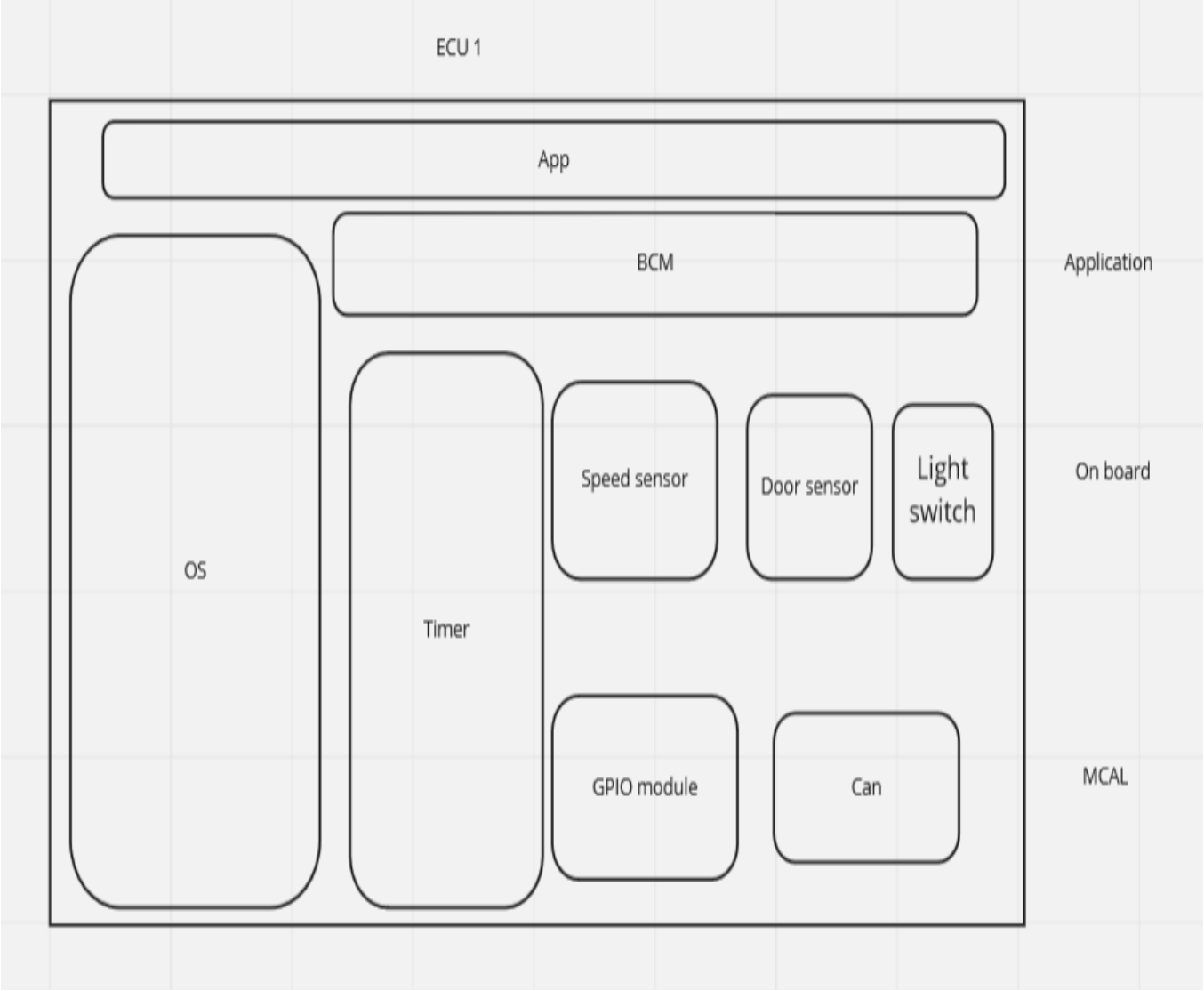
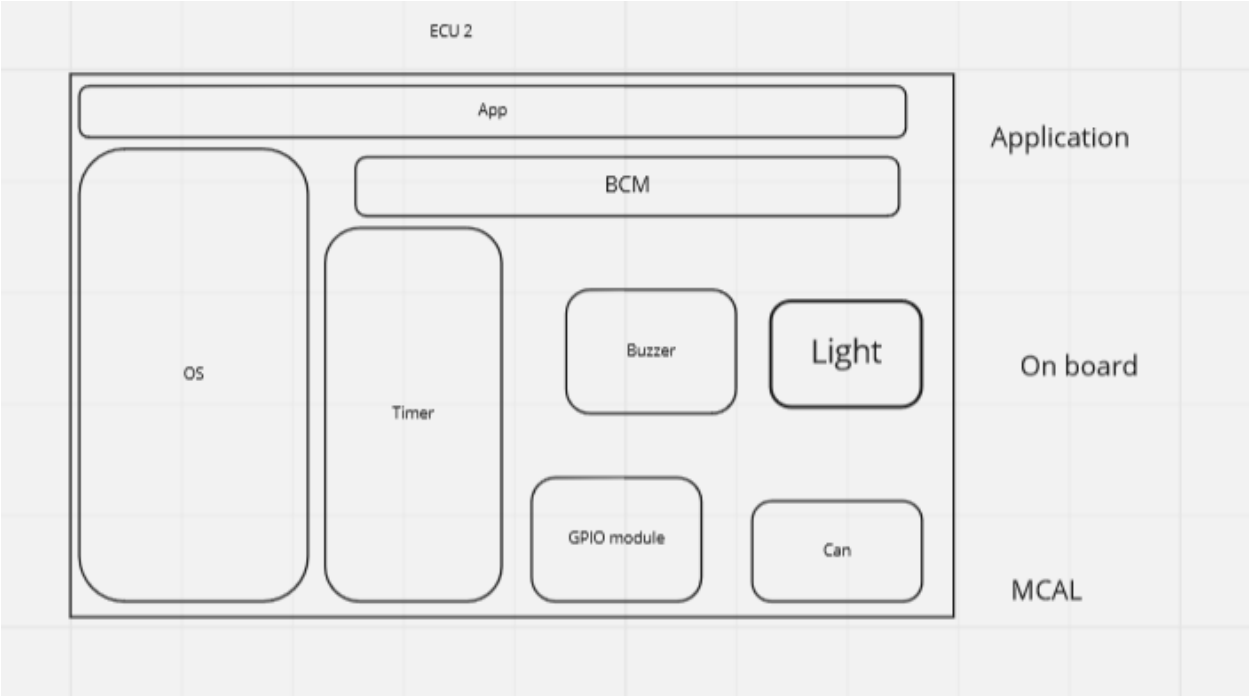
1-Layered architecture




ECU 2
App
BCM
OS
Timer
Buzzer
Light
GPIO module
Can
Application
On board
MCAL

2-ECU components and modules for ECU1

1-GPIO module
2-CANmodule
3-Door state module
4-Speedsensor module
5-Timer module
7-OS
8-Light switch modules

3-APIS
For GPIO:

Function	**GPIO_init**
syntax	**Void GPIO_init(GPIO_ConfigType * Config_Ptr)**
Input	**Config_Ptr**
Description	**Initialize GPIO driver**

Function	**GPIO_ReadPin**
Syntax	**GPIO_LevelType GPIO_ReadPin(GPIO_PinType GPIO_Pin)**
Input	**GPIO_Pin**
Output	**GPIO_LevelType**
Description	**Read state of pin (STD_high or STD_low)**

Function	**GPIO_WritePin**
Syntax	**void GPIO_ReadPin(GPIO_PinType GPIO_Pin, GPIO_LevelType Level_Type)**
Input	**GPIO_Pin , Level_Type**
Description	**Write state of pin (STD_high or STD_low)**

For BCM:

Function	**BCM_Writemsg**
Syntax	**Void BCM_Writemsg(CAN_messageType *message)**
Input	**message**
Desciption	**Send CAN message**

Function	**BCM_Readmsg**
Syntax	**Void BCM_Readmsg(CAN_messageType *message)**
Input	**message**
Desciption	**Recieve CAN message**

For CAN module:

Function	**Can_Init**
Syntax	**Void Can_Write(CAN_type *Config_Ptr)**
Input	**Config_Ptr**
Desciption	**Init CAN with Configuration required**

Function	**Can_Write**

Syntax	**Void Can_Write(CAN_messageType *message)**
Input	**message**
Desciption	**Send CAN message**

Function	**Can_Read**
Syntax	**Void Can_Read(CAN_messageType *message)**
Input	**message**
Desciption	**Recieve CAN message**

Function	**Can_TxConfirmation**
Syntax	**STD_RETURN Can_TxConfirmation(void)**
Input	**CAN_messageType *message**
Desciption	**Return OK or N_OK to indicate if msg sent succefully**

For Door Sensor Module:

Function	**Door_Init**
Syntax	**void Door_Init(DoorSensor_Type *Config_Ptr)**
Input	**Config_Ptr**
Desciption	**Initialize the Door**

Function	**Door_GetState**

Syntax	**DoorState_type Door_GetState(void)**
Output	**DoorState_type**
Desciption	**Read Door state(STD_high, STD_low)**

For SpeedSensor Module:

Function	**SpeedSensor_Init**
Syntax	**void SpeedSensor_Init(SpeedSensor_Type *Config_Ptr)**
Input	**Config_Ptr**
Desciption	**Initialize the SpeedSensor**

Function	**Speed_GetState**
Syntax	**SpeedState_type Speed_GetState(void)**
Output	**SpeedState_type**
Desciption	**Read Speed sensor state(STD_high, STD_low)**

For LightSwitch Module:

Function	**LightSwitch_Init**
Syntax	**void LightSensor_Init(LightSensor_type *config_ptr)**
Output	**config_ptr**
Desciption	**Initialize the light switch**

Function	**LightSwitch_GetState**
Syntax	**LightSwitchState_type LightSensor_GetState(void)**
Output	**LightSwitchState_type**

Desciption	**Read Light Switch state (STD_high, STD_low)**

For timer:

Function	**Timer_Init**
Syntax	**Void Timer_Init(TimerConfig_Type *Config_Ptr)**
Input	**Config_Ptr**
Desciption	**Initialize the timer**

Function	**Timer_Notification**
Syntax	**void Timer_Notification(void(*Ptr2Func)(void));**
Input	**Ptr2Func**
Desciption	**Set the CallBack function**

2-ECU components and modules for ECU1:

For GPIO:

Function	**GPIO_init**
syntax	**Void GPIO_init(GPIO_ConfigType * Config_Ptr)**
Input	**Config_Ptr**
Description	**Initialize GPIO driver**

Function	**GPIO_ReadPin**
Syntax	**GPIO_LevelType GPIO_ReadPin(GPIO_PinType GPIO_Pin)**

Input	**GPIO_Pin**
Output	**GPIO_LevelType**
Description	**Read state of pin (STD_high or STD_low)**

Function	**GPIO_WritePin**
Syntax	**void GPIO_ReadPin(GPIO_PinType GPIO_Pin, GPIO_LevelType Level_Type)**
Input	**GPIO_Pin , Level_Type**
Description	**Write state of pin (STD_high or STD_low)**

For BCM:

Function	**BCM_Writemsg**
Syntax	**Void BCM_Writemsg(CAN_messageType *message)**
Input	**message**
Desciption	**Send CAN message**

Function	**BCM_Readmsg**
Syntax	**Void BCM_Readmsg(CAN_messageType *message)**
Input	**message**
Desciption	**Recieve CAN message**

For CAN module:

Function	**Can_Write**
Syntax	**Void Can_Write(CAN_messageType *message)**
Input	**message**

Desciption	**Send CAN message**

Function	**Can_Read**
Syntax	**Void Can_Read(CAN_messageType *message)**
Input	**message**
Desciption	**Recieve CAN message**

For timer:

Function	**Timer_Init**
Syntax	**Void Timer_Init(TimerConfig_Type *Config_Ptr)**
Input	**Config_Ptr**
Desciption	**Initialize the timer**

Function	**Timer_Notification**
Syntax	**void Timer_Notification(void(*Ptr2Func)(void));**
Input	**Ptr2Func**
Desciption	**Set the CallBack function**

For Buzzer Module

Function	**Buzzer_SetState**
Syntax	**void Buzzer _SetState(BuzzerState_Type Buzzer_Status)**

Input	**Buzzer_Status**
Desciption	**Set buzzer state (STD_high / STD_low)**

For Light Module

Function	**Light_SetState**
Syntax	**void Light _SetState(LightState_Type Light _Status)**
Input	**Light_Status**
Desciption	**Read Light Switch state (STD_high /STD_low)**

TypeDefs:

Name	Type	Range
GPIO_LevelType	uint8	STD_high 1U STD_LOW 0U
GPIO_ConfigType	Structure	he contents of the initialization data structure are specific to the microcontroller. (from autsar)
CAN_messageType	Structure	
TimerConfig_Type	Structure	
Door_GetState	uint8	STD_high 1U STD_LOW 0U
SpeedState_type	uint8	STD_high 1U STD_LOW 0U
SpeedState_type	uint8	STD_high 1U STD_LOW 0U
BuzzerState_Type	uint8	STD_high 1U STD_LOW 0U
LightState_Type	uint8	STD_high 1U

		STD_LOW 0U