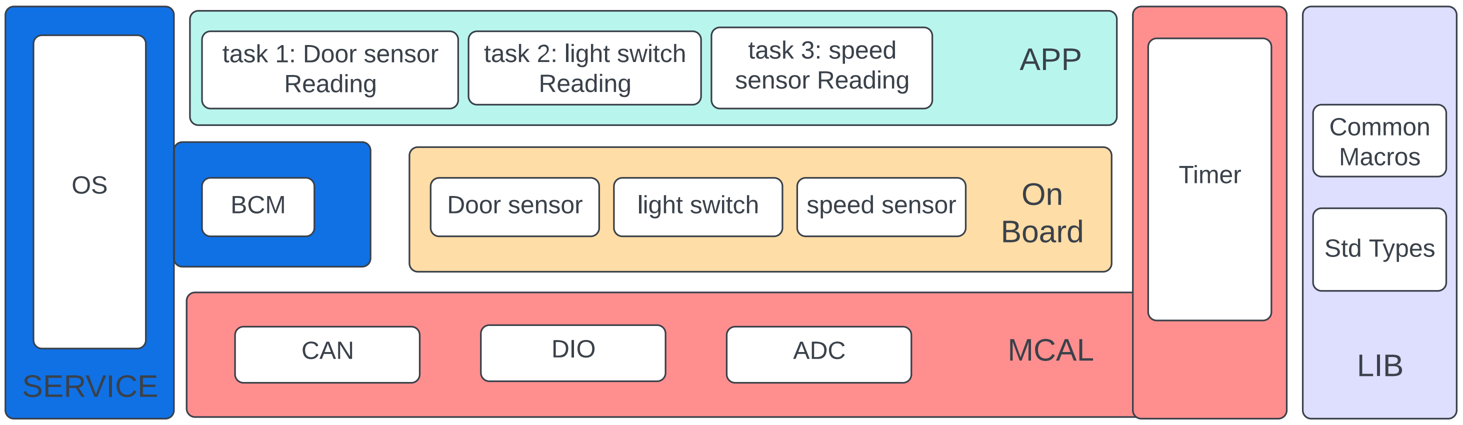
Static Design

ECU 1



Full Detailed APIs:

Name	**DIO Pin Init**
Description	Pin initialization
Syntax	void DIO_pin_init(DIOPort_t Port, uint8 Pin , DIODIR_t dir)
Parameters (in)	Port, Pin, direction
Parameters (out)	None
Return Value	void

Name	DIOPort_t
Type	Enum
Description	Contains all ports of the ECU

Name	DIODIR_t
Type	Enum
Description	Contains the directions of I/O

Name	**DIO Pin read**
Description	Read pin value
Syntax	Uint8 DIO_pin_init(DIOPort_t Port, uint8 Pin)
Parameters (in)	Port, Pin
Parameters (out)	None
Return Value	uint8

Name	**DIO Pin write**
Description	write pin value
Syntax	void DIO_pin_init(DIOPort_t Port, uint8 Pin , uint8 val)
Parameters (in)	Port, pin, val
Parameters (out)	None
Return Value	void

Name	**Timer Init**
Description	Initializing timer
Syntax	Void Timer_Init(Timer_Config* config)
Parameters (in)	config
Parameters (out)	None
Return Value	void

Name	Timer_Config
Type	Struct
Description	Contains the main configurations of the targeted timer

Name	**Timer delay**
Description	Timer delay in ms
Syntax	Void Timer_delay_m (uint32 delayValue)
Parameters (in)	delayValue
Parameters (out)	None
Return Value	void

Name	**Timer delay callback**
Description	Timer delay in ms with callback
Syntax	Void Timer_delay_m_call (uint32 delayValue , callbackPtr* callbackFunc)
Parameters (in)	delayValue, callbackFunc
Parameters (out)	None
Return Value	void

Name	**ADC Init**
Description	ADC initialization
Syntax	Void ADC_Init(uint8 ADC_Channel, ADC_Config* config)
Parameters (in)	ADC_Channel, config
Parameters (out)	None
Return Value	void

Name	ADC_Config
Type	Struct
Description	Contains the main configurations of the targeted ADC

Name	**ADC Get Value**
Description	Getting the ADC value
Syntax	Uint8 ADC_Read(uint8 ADC_Channel)
Parameters (in)	ADC_Channel
Parameters (out)	None
Return Value	Uint8

Name	**CAN Init**
Description	CAN initialization
Syntax	Void CAN_Init(CAN_Config_t* Config)
Parameters (in)	Config
Parameters (out)	None
Return Value	void

Name	CAN_Config_t
Type	Struct
Description	Contains the main configurations of the CAN

Name	**CAN Send**
Description	Sending 1 byte of data
Syntax	Void CAN_Send(uint8 data)
Parameters (in)	Data
Parameters (out)	None
Return Value	void

Name	**CAN recieve**
Description	Receiving 1 byte of data
Syntax	Uint8 CAN_Recieve(void)
Parameters (in)	void
Parameters (out)	None
Return Value	Uint8

Name	**Door sensor init**
Description	Initializing the door sensor
Syntax	Void doorSen_init(void)
Parameters (in)	void
Parameters (out)	None
Return Value	void

Name	**Door sensor read**
Description	Reading the door sensor value
Syntax	Uint8 doorSen_read(void)
Parameters (in)	void
Parameters (out)	None
Return Value	Uint8

Name	**Speed sensor init**
Description	Initializing the speed sensor
Syntax	Void speedSen_init(void)
Parameters (in)	void
Parameters (out)	None
Return Value	Void

Name	**speed sensor read**
Description	Reading the speed sensor value
Syntax	Uint16 doorSen_read(void)
Parameters (in)	void
Parameters (out)	None
Return Value	Uint16

Name	**Light switch init**
Description	Initializing the light switch
Syntax	Void lightSwitch_init(void)
Parameters (in)	void
Parameters (out)	None

Return Value	Void

Name	**Light switch read**
Description	Reading the light switch value
Syntax	Uint8 lightSwitch_read(void)
Parameters (in)	void
Parameters (out)	None
Return Value	Uint8

Name	**Send data**
Description	Sending 1 byte of data and choosing the protocol
Syntax	Void sendData(uint8 data , BCM_protocol_t protocol)
Parameters (in)	Data , protocol
Parameters (out)	None
Return Value	void

Name	BCM_protocol_t
Type	Enum
Description	Contains all protocols that needed to communicate

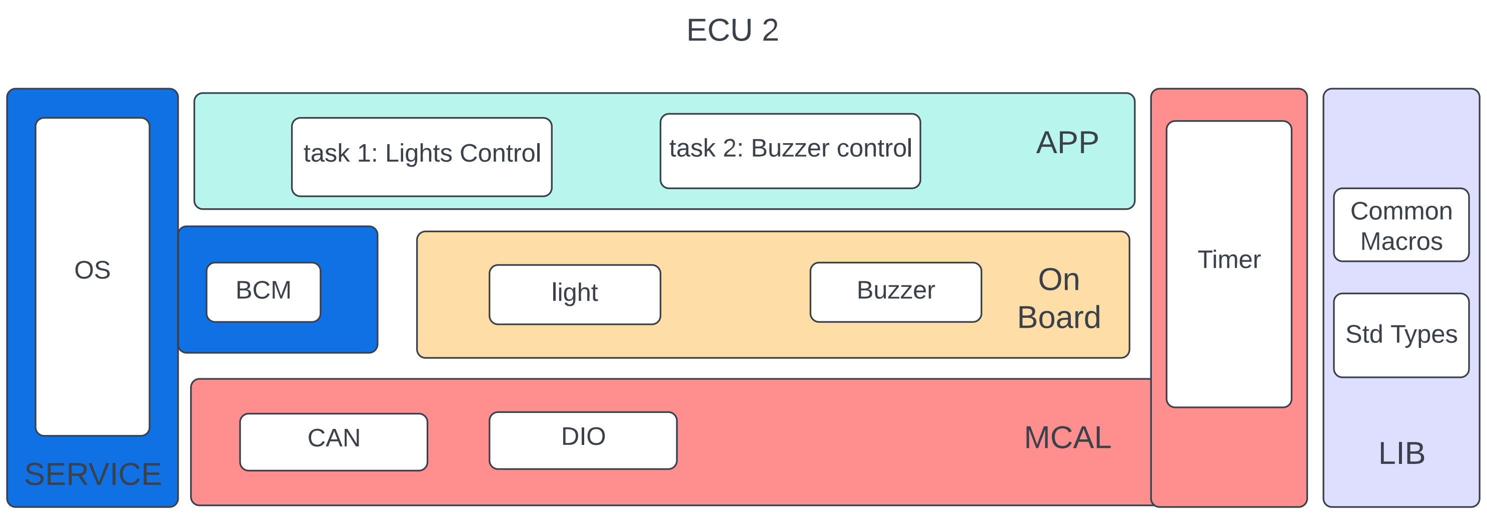
Name	**Receive data**
Description	receiving 1 byte of data and choosing the protocol
Syntax	Uint8 receiveData(BCM_protocol_t protocol)
Parameters (in)	protocol
Parameters (out)	None
Return Value	Uint8

Name	**Door sensor task**
Description	Reading and sending the door sensor value
Syntax	void doorSen_task(void)

Parameters (in)	void
Parameters (out)	None
Return Value	void

Name	**speed sensor task**
Description	Reading and sending the speed sensor value
Syntax	void speedSen_task(void)
Parameters (in)	void
Parameters (out)	None
Return Value	void

Name	**Light switch task**
Description	Reading and sending the Light switch state
Syntax	void LightSwitch _task(void)
Parameters (in)	void
Parameters (out)	None
Return Value	void



MCAL and SERVICE drivers are common with ECU 1.

Name	**Lights init**
Description	Initializing the lights
Syntax	Void lights_init(void)
Parameters (in)	void
Parameters (out)	None
Return Value	Void

Name	**Lights state**
Description	Reading the light’s state
Syntax	Uint8 static volatile lights_state;
Parameters (in)	None
Parameters (out)	None
Return Value	None

Name	**Lights write**
Description	Changing the lights state
Syntax	Void lights_write (uint8 state)
Parameters (in)	State
Parameters (out)	None
Return Value	Void

Name	**buzzer init**
Description	Initializing the buzzer
Syntax	Void buzzer_init(void)
Parameters (in)	void
Parameters (out)	None
Return Value	Void

Name	**Buzzer state**
Description	Reading the buzzer's state
Syntax	Uint8 static volatile buzzer_state;
Parameters (in)	None
Parameters (out)	None
Return Value	None

Name	**buzzer write**
Description	Changing the buzzer state
Syntax	Void buzzer_write (uint8 state)
Parameters (in)	State
Parameters (out)	None
Return Value	Void

Name	**Lights task**
Description	Controlling the lights state
Syntax	void lights_task(void)
Parameters (in)	void
Parameters (out)	None
Return Value	void

Name	**Buzzer task**
Description	Controlling the buzzer state
Syntax	void buzzer _task(void)
Parameters (in)	void
Parameters (out)	None
Return Value	void