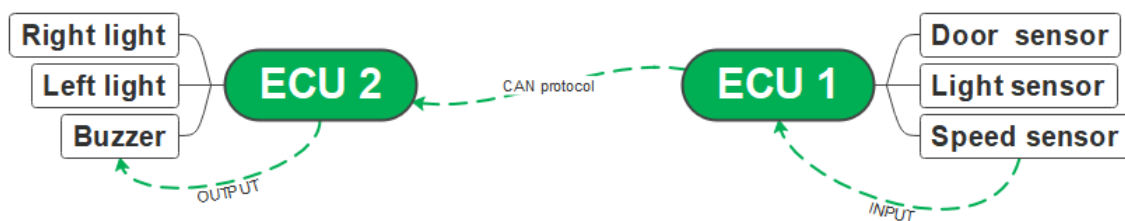


Embedded system design project

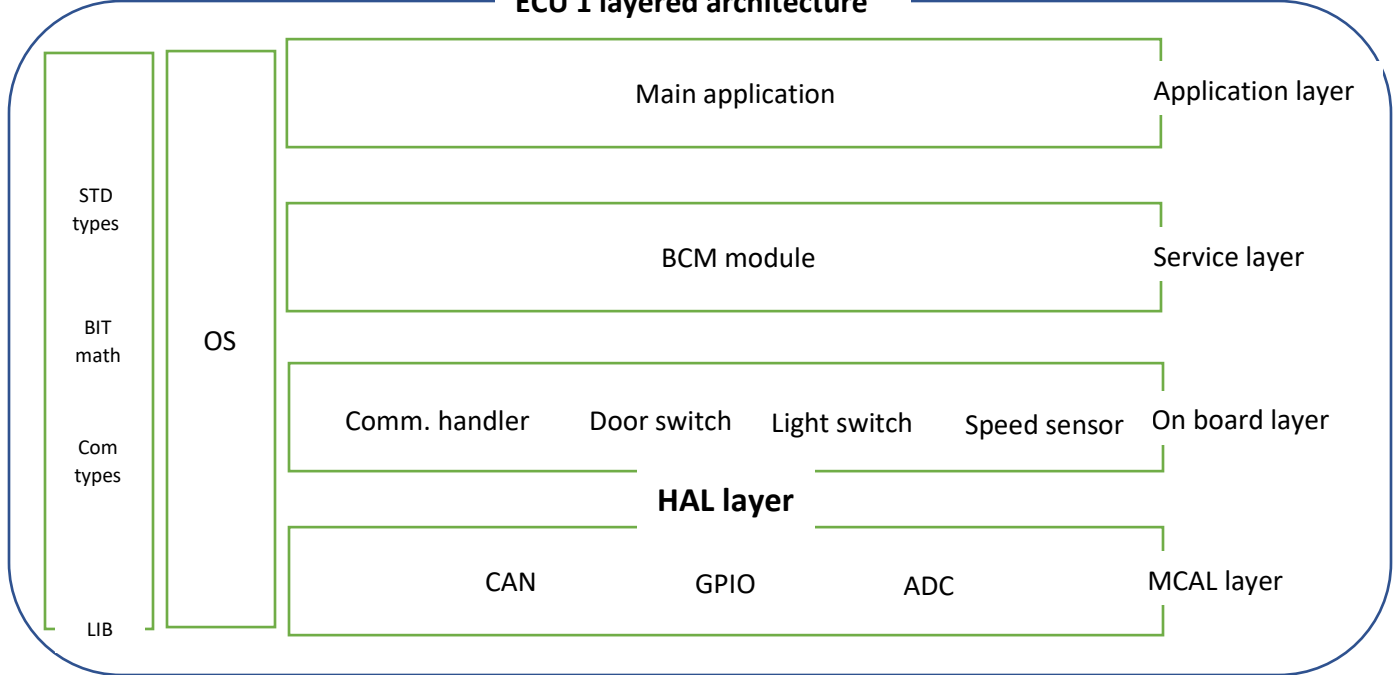
Content:

- Static design
 - System block diagram design
 - Layered architecture
 - ECUs modules and components used
 - Detailed APIs
- Dynamic design
 - State machine diagram of each component
 - State machine diagram of the ECU operation
 - Sequence diagram

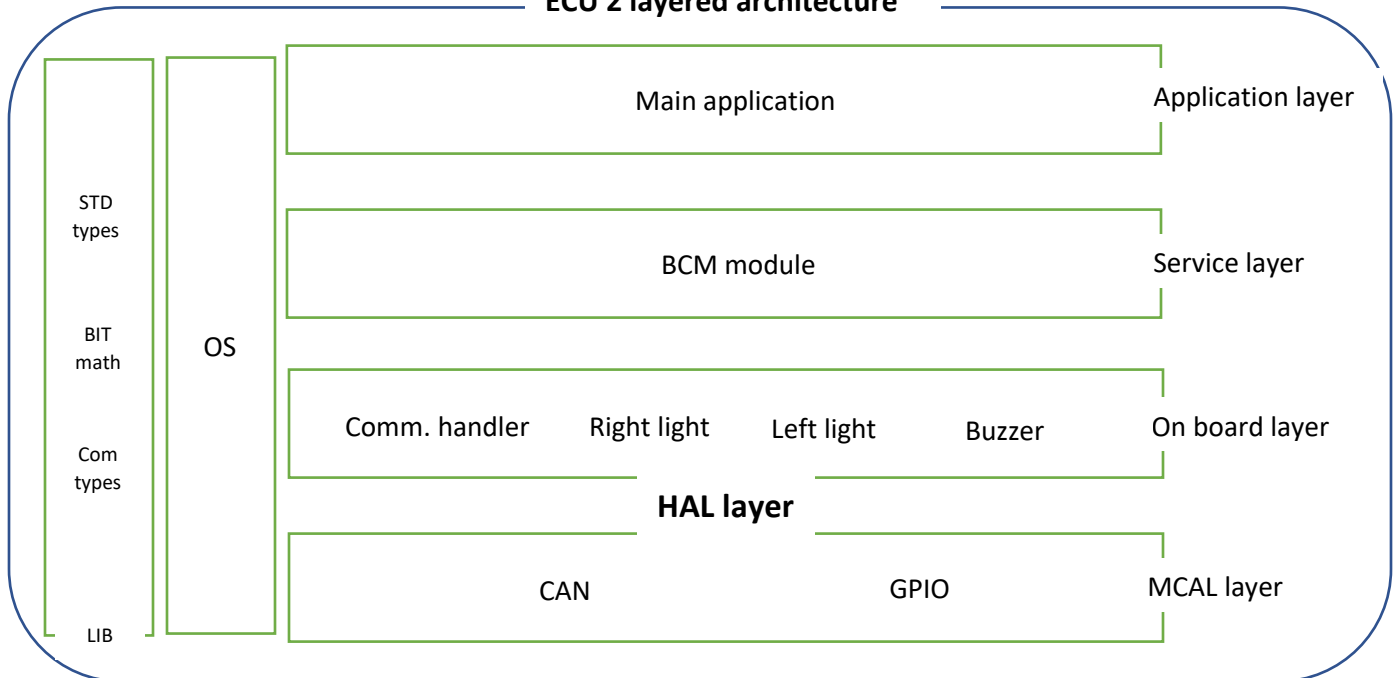
System block diagram design



ECU 1 layered architecture



ECU 2 layered architecture



ECU 1 components and modules:

- GPIO
- ADC
- CAN
- RTOS
- LIB
- BCM module
- Communication handler
- Door switch
- Light switch
- Speed sensor

ECU 2 components and modules:

- GPIO
- CAN
- RTOS
- LIB
- BCM module
- Communication handler
- Right light
- Left light
- Buzzer

ECU 1 APIs and data types:

Function name	GPIO_GetPinDirection(portID port, PinNo pin)			
Arguments	Inputs	port	portID	
		Description: Port number		
		pin	PinNo	
Return	GPIO_INPUT GPIO_OUTPUT	0		
		Description: Pin number		
		1		
Description	This function is used to get a pin's direction			

Function name	GPIO_SetPinValue(portID port, PinNo pin, PinValue value)			
Arguments	Inputs	port	portID	
		Description: Port number		
		pin	PinNo	
		Description: Pin number		
		value	PinValue	
Return	void	Description: Pin value (H/L)		
		N/A		
Description	This function is used to set a pin's value to high or low			

Function name	GPIO_INIT(void)			
Arguments	Inputs		N/A	
Return	void		N/A	
Description	This function is responsible for initializing GPIO driver			

Function name	GPIO_SetPinDirection(portID port, PinNo pin, PinDir direction)			
Arguments	Inputs	port	portID	
		Description: Port number		
		pin	PinNo	
		Description: Pin number		
		direction	PinDir	
Return	void	Description: Pin direction		
		N/A		
Description	This function is used to set a pin's direction to output or input			

Function name	GPIO_GetPinValue(portID port, PinNo pin)			
Arguments	Inputs	port	portID	
		Description: Port number		
		pin	PinNo	
Return	GPIO_LOW GPIO_HIGH	0		
		Description: Pin number		
		1		
Description	This function is used to get a pin's value			

Name	portID			
Type	Enumeration			
Range	PORTA	0	Description: choose PortA	
	PORTB	1	Description: choose PortB	
	PORTC	2	Description: choose PortC	
Description	This enumeration is used to help with choosing the port in code			

Name	PinNo			
Type	Enumeration			
Range	PIN0	0	Description: Choose pin 0	
	PIN1	1	Description: Choose pin 1	
	PIN2	2	Description: Choose pin 2	
	PIN3	3	Description: Choose pin 3	
	PIN4	4	Description: Choose pin 4	
	PIN5	5	Description: Choose pin 5	
	PIN6	6	Description: Choose pin 6	
	PIN7	7	Description: Choose pin 7	
Description	This enumeration is used to help with choosing the pins in code			

Name	PinDir			
Type	Enumeration			
Range	GPIO_INPUT	0	Description: Set the pin as input	
	GPIO_OUTPUT	1	Description: Set the pin as output	
Description	This enumeration is used to help with choosing the direction of a pin in code			

Name	PinValue		
Type	Enumeration		
Range	GPIO_LOW	0	Description: set the pin to low
	GPIO_HIGH	1	Description: set the pin to high
Description	This enumeration is used to help with choosing the value of a pin in code		

Function name	ADC_INIT(void)		
Arguments	Inputs		N/A
Return	void		N/A
Description	This function is responsible for initializing ADC driver		

Function name	ADC_StartContinuous(void)		
Arguments	Inputs		N/A
Return	void		N/A
Description	This function is used to start ADC conversion continuously		

Function name	ADC_StopContinuous(void)		
Arguments	Inputs		N/A
Return	void		N/A
Description	This function is used to stop continuous ADC conversion		

Function name	ADC_ReadValue(ADC_Channel_SEL ADC_Channel, ADC_Resolution_t ADC_Resolution)		
Arguments	Inputs	ADC_Channel	ADC_Channel_SEL
		Description: select ADC channel	
		ADC_Resolution	ADC_Resolution_t
Return	u_int16		Description: data to be read
Description	This function is used to get the data read on a specific channel		

Function name	CAN_INIT(void)		
Arguments	Inputs		N/A
Return	E_OK		Description: The operation done successfully
	E_NOK		Description: The operation wasn't successful
Description	This function is used to initialize CAN driver		

Function name	CAN_SendByte(u_int8 Byte)		
Arguments	Inputs	Byte	u_int8
		Description: the byte to be sent	
Return	E_OK		Description: The operation done successfully
	E_NOK		Description: The operation wasn't successful
Description	This function is used to send a byte of data		

Name	OutputState		
Type	Enumeration		
Range	E_OK	0	Description: The operation done successfully
	E_NOK	1	Description: The operation wasn't successful
Description	This enumeration is used to check if a function succeeded to run		

Name	ADC_Channel_SEL		
Type	Enumeration		
Range	A0	0	Description: Select channel A0
	A1	1	Description: Select channel A1
	A2	2	Description: Select channel A2
	A3	3	Description: Select channel A3
	A4	4	Description: Select channel A4
	A5	5	Description: Select channel A5
Description			

Name	ADC_Resolution_t		
Type	Enumeration		
Range	eight_bits	0	Description: Select 8 bits resolution
	ten_bits	1	Description: Select 10 bits resolution
Description	This Enumeration is used to set the ADC resolution		

Name	Byte		
Type	u_int8		
Range	0:255		
Description	The data we want to send or receive over CAN		
Function name	SpeedSensorRead()		
Arguments	Inputs		N/A
Return	u_int16		Description: speed sensor reading
Description	This function is used to get the data out of the speed sensor		
Function name	LightSwitchRead()		
Arguments	Inputs		N/A
Return	LOW		Description: The light switch state is low
	HIGH		Description: The light switch state is high
Description	This function is used to get the status of the light switch		
Function name	DoorSwitchRead()		
Arguments	Inputs		N/A
Return	LOW		Description: The door state is low
	HIGH		Description: The door state is high
Description	This function is used to get the status of the door switch		
Function name	BCM_SendByte(u_int8 Byte)		
Arguments	Inputs		Byte u_int8
			Description: byte to be sent
Return	E_OK		Description: The operation done successfully
	E_NOK		Description: The operation wasn't successful
Description	This function is used to send a byte of data through BCM module		
Function name	BCM_HandlerSendByteThroughCAN(u_int8 Byte)		
Arguments	Inputs		Byte u_int8
			Description: byte to be sent
Return	E_OK		Description: The operation done successfully
	E_NOK		Description: The operation wasn't successful
Description	This function is used to send a byte of data through BCM handler to CAN module		

ECU 2 APIs and data types:

Function name	GPIO_INIT(void)			
Arguments	Inputs		N/A	Desc: N/A
Return	void		N/A	
Description	This function is responsible for initializing GPIO driver			
Function name	GPIO_SetPinDirection(portID port, PinNo pin, PinDir direction)			
Arguments	Inputs		port	portID
			Description: Port number	
			pin	PinNo
			Description: Pin number	
			direction	PinDir
		Description: Pin direction		
Return	void		N/A	
Description	This function is used to set a pin's direction to output or input			
Function name	GPIO_GetPinDirection(portID port, PinNo pin)			
Arguments	Inputs		port	portID
			Description: Port number	
			pin	PinNo
			Description: Pin number	
Return	GPIO_INPUT		0	
	GPIO_OUTPUT		1	
Description	This function is used to get a pin's direction			
Function name	GPIO_SetPinValue(portID port, PinNo pin, PinValue value)			
Arguments	Inputs		port	portID
			Description: Port number	
			pin	PinNo
			Description: Pin number	
			value	PinValue
		Description: Pin value (H/L)		
Return	void		N/A	
Description	This function is used to set a pin's value to high or low			
Function name	GPIO_GetPinValue(portID port, PinNo pin)			
Arguments	Inputs		port	portID
			Description: Port number	
			pin	PinNo
			Description: Pin number	
Return	GPIO_LOW		0	
	GPIO_HIGH		1	
Description	This function is used to get a pin's value			
Name	portID			
Type	Enumeration			
Range	PORTA	0	Description: choose PortA	
	PORTB	1	Description: choose PortB	
	PORTC	2	Description: choose PortC	
Description	This enumeration is used to help with choosing the port in code			

Name	PinNo		
Type	Enumeration		
Range	PIN0	0	Description: Choose pin 0
	PIN1	1	Description: Choose pin 1
	PIN2	2	Description: Choose pin 2
	PIN3	3	Description: Choose pin 3
	PIN4	4	Description: Choose pin 4
	PIN5	5	Description: Choose pin 5
	PIN6	6	Description: Choose pin 6
	PIN7	7	Description: Choose pin 7
Description	This enumeration is used to help with choosing the pins in code		
Name	PinDir		
Type	Enumeration		
Range	GPIO_INPUT	0	Description: Set the pin as input
	GPIO_OUTPUT	1	Description: Set the pin as output
Description	This enumeration is used to help with choosing the direction of a pin in code		
Function name	BuzzerOFF()		
			N/A
Return	E_OK		Description: The operation done successfully
	E_NOK		Description: The operation wasn't successful
Description	This function is used to turn the buzzer off		
Function name	RightLightON()		
			N/A
Return	E_OK		Description: The operation done successfully
	E_NOK		Description: The operation wasn't successful
Description	This function is used to turn the right light on		
Function name	RightLightOFF()		
Arguments	Inputs		N/A
Return	E_OK		Description: The operation done successfully
	E_NOK		Description: The operation wasn't successful
Description	This function is used to turn the right light off		

Function name	LeftLightON()	
Arguments	Inputs	
Return	E_OK	Description: The operation done successfully
	E_NOK	Description: The operation wasn't successful
Description	This function is used to turn the left light on	
Function name	LeftLightOFF()	
Arguments	Inputs	
Return	E_OK	Description: The operation done successfully
	E_NOK	Description: The operation wasn't successful
Description	This function is used to turn the left light off	
Function name	BCM_ReadByte()	
Arguments	Inputs	
Return	E_OK	Description: The operation done successfully
	E_NOK	Description: The operation wasn't successful
Description	This function is used to read a byte of data through BCM module	
Function name	BCM_HandlerReadByteThroughCAN()	
Arguments	Inputs	
Return	E_OK	Description: The operation done successfully
	E_NOK	Description: The operation wasn't successful
Description	This function is used to read a byte of data through BCM handler to CAN module	

Name	PinValue		
Type	Enumeration		
Range	GPIO_LOW	0	Description: set the pin to low
	GPIO_HIGH	1	Description: set the pin to high
Description	This enumeration is used to help with choosing the value of a pin in code		

Function name	CAN_INIT(void)			
Arguments	Inputs	N/A		N/A
		Description: N/A		
Return	E_OK	0		
	E_NOK	1		
Description	This function is used to initialize CAN driver			

Function name	CAN_ReadByte()	
		N/A
Return	u_int8	Description: the byte to be read
Description	This function is used to read a byte of data	

Name	OutputState		
Type	Enumeration		
Range	E_OK	0	Description: The operation done successfully
	E_NOK	1	Description: The operation wasn't successful
Description	This function is used to check if a function succeeded to run		

Name	Byte
Type	u_int8
Range	0:255
Description	The data we want to send or receive over CAN

Function name	BuzzerON()	
		N/A
Return	E_OK	Description: The operation done successfully
	E_NOK	Description: The operation wasn't successful
Description	This function is used to turn the buzzer on	