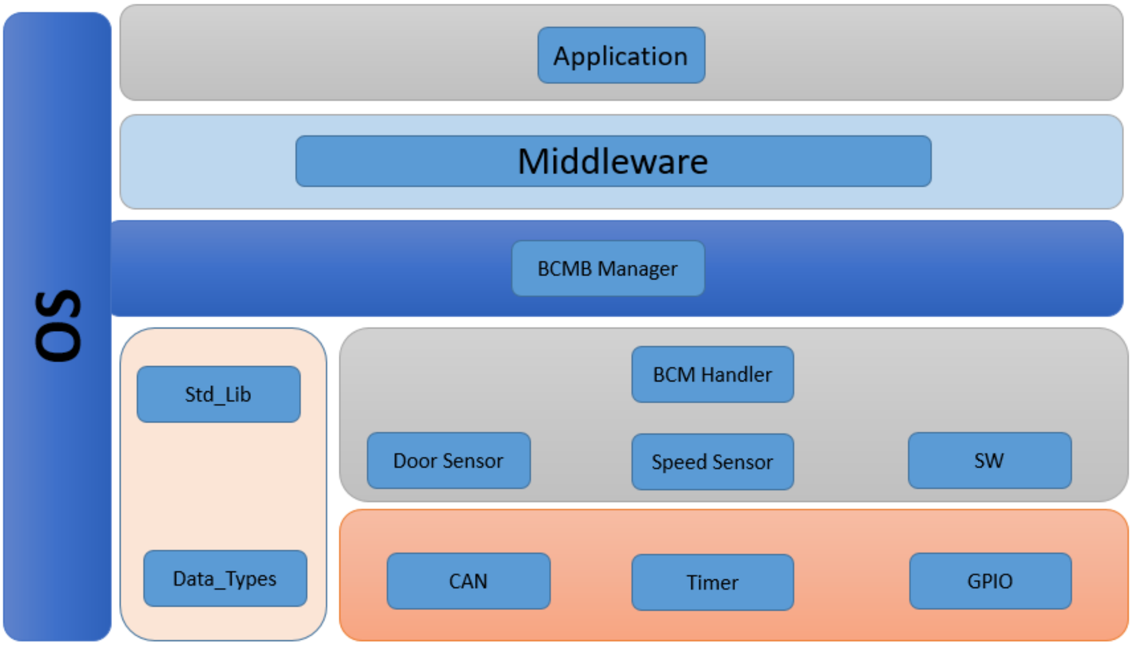
Static Design

For ECU 1

1. Layered Architecture



2. Main components and Modules

   a. Components
      i. Door Sensor
      ii. Speed Sensor
      iii. Light Switch
      iv. Can Unit

   b. Modules
      i. GPIO Module
      ii. Timer Module
      iii. Light Switch Module
      iv. Door Sensor Module
      v. Speed Sensor Module
      vi. Can Module

3. APIs Details

Application Layer			
API Prototype	**Parameters**	**Return**	**Description**
Void TaskSpeedSensor(void)	No parameters	No return	This task will send Speed Sensor read to BCM every 5ms
Void TaskDoorSensor(void)	No parameters	No return	This task will send Door Sensor read to BCM every 10ms
Void TaskLightSW (void)	No parameters	No return	This task will send Switch State read to BCM every 5ms

Service Layer			
API Prototype	**Parameters**	**Return**	**Description**
State_t BCM_Manager(uint8 Data , uint8 ID)	Data wanted to send and Node ID	Return Enum State of sending to Can unit ok or not	It will send all received data from all tasks to Can unit and return ACK ok or not

On Board Layer : Speed Sensor

API Prototype	Parameters	Return	Description
Void SpeedSensorInit(uint8 Channel)	Uint8 Channel which sensor connected to it	Not return	It will initiate ADC and GPIO configurations for Speed Sensor
Uint32 SpeedSensorRead(uint8 Channel)	Uint8 Channel which sensor connected to it	Return channel Reading Value	It will read analog value of sensor pin and represent it to physical value and return it

On Board Layer : Door Sensor

API Prototype	Parameters	Return	Description
Void DoorSensorInit(uint8 Pin)	Uint8 Pin which sensor connected to it	Not return	It will initiate GPIO configurations for Door Sensor
DoorState_t DoorSensorRead(void)	No parameters	Return DoorState_t Opened or closed	It will read Digital value of sensor pin high or low and return state

On Board Layer : Light Switch

API Prototype	Parameters	Return	Description
Void SwitchInit(uint8 Pin)	Uint8 Pin which sensor connected to it	Not return	It will initiate GPIO configurations for switch
SwState_t SwitchReadState(void)	No parameters	Return SwState_t Pressed or not	It will read Digital value of switch pin high or low and return state

MCAL Layer : GPIO

API Prototype	Parameters	Return	Description
Void GPIO_SetDirection(uint8 port , uint8 pin , PinDir_t pinDir)	Uint8 port name Uint8 pin umber PinDir_t pinDir Input or Output	Not return	It will set direction of any pin at any wanted port
Void GPIO_SetValue(uint8 port , uint8 pin , PinState_t pinState)	Uint8 port name Uint8 pin umber PinState_t pinState high or low	Not return	It will set state of pin at any wanted register

MCAL Layer : Timer

API Prototype	Parameters	Return	Description
Void Timer_Init(TimerType_t tType , TimerMode_t tMode)	TimerType_t tType TimerMode_t tMode	Not return	It will Set Timer Configuration based on Timer type and timer mode
Void TimerStart(uint32 StartValue)	uint32 StartValue of Timer Counter Register	Not return	It will set start value of the timer and setting prescaller of it based on wanted time
Void TimerStop(void)	No parameters	No return	It will stop Timer count
Uint32 TimerRead(void)	No parameters	Timer Counter Register Value	It will return Timer Counter Register Value

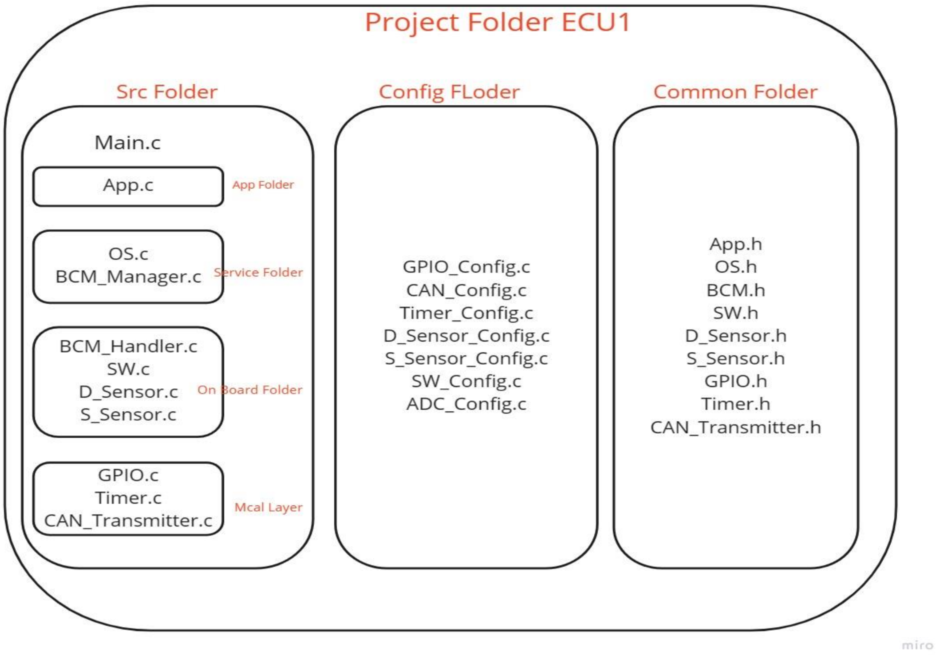
MCAL Layer : CAN Transmitter Unit

API Prototype	Parameters	Return	Description
Void CAN_Init(void)	No parameters	Not return	It will Set Can Uint Configurations
Void CAN_Send(uint32 Data)	uint32 Data wanted sent over CAN	Not return	It will send data to CAN Receiver Unit

MCAL Layer : ADC

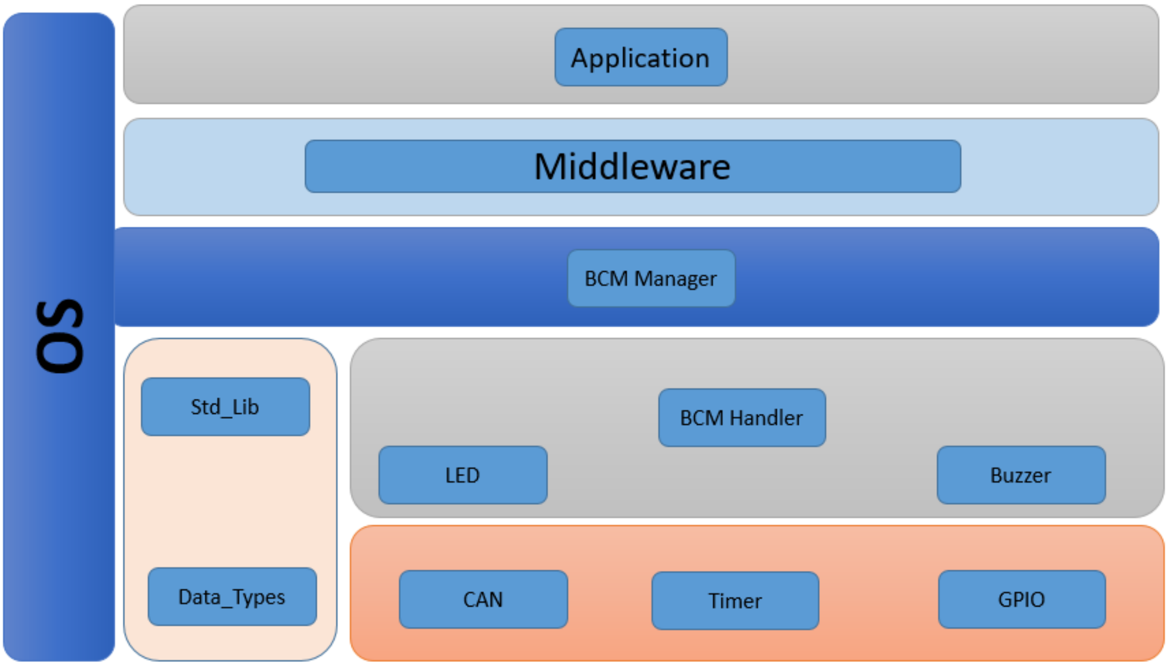
API Prototype	Parameters	Return	Description
Void ADC_Init(uint8 Channel)	uint8 Channel wanted ADC channel to init	Not return	It will Set ADC Configurations Based on channel number
Uint32 ADC_Read(void)	No parameters	ADC Reading value	It will return Reading value of ADC Channel

4. Folder Structure



For ECU 2

1. Layered Architecture



2. Main components and Modules
   a. Components
      i. LED
      ii. Buzzer
   b. Modules
      i. GPIO Module
      ii. Timer Module
      iii. LED Module
      iv. Buzzer Module
      v. Can Receiver Module

3. APIs Details

Application Layer			
API Prototype	**Parameters**	**Return**	**Description**
Uint32 TaskSpeedSensorReceive(void)	No parameters	Return Receiving Data	This task will receive Speed Sensor read from BCM
DoorState_t TaskDoorSensorReceive(void)	No parameters	Return Door State	This task will receive Door Sensor read from BCM
PinState_t TaskLightSWReceive(void)	No parameters	Return SW State	This task will receive Switch State read from BCM

Service Layer			
API Prototype	**Parameters**	**Return**	**Description**
State_t BCM_Manager(uint8 Data , uint8 ID)	Data wanted to send and Node ID	Return Enum State of sending to Can unit ok or not	It will send all received data from all tasks to Can unit and return ACK ok or not

On Board Layer : LED

API Prototype	Parameters	Return	Description
Void LED_Init(uint8 port , uint8 pin)	uint8 port uint8 pin connected to LED	Not return	It will initiate GPIO configurations for LED
Void LED_ON(void)	No parameters	No return	It will turn on the led
Void LED_OFF(void)	No parameters	No return	It will turn off the led

On Board Layer : Buzzer

API Prototype	Parameters	Return	Description
Void LED_Buzzer(uint8 port , uint8 pin)	uint8 port uint8 pin connected to Buzzer	Not return	It will initiate GPIO configurations for Buzzer
Void Buzzer_ON(void)	No parameters	No return	It will turn on the Buzzer
Void Buzzer_OFF(void)	No parameters	No return	It will turn off the Buzzer

MCAL Layer : GPIO

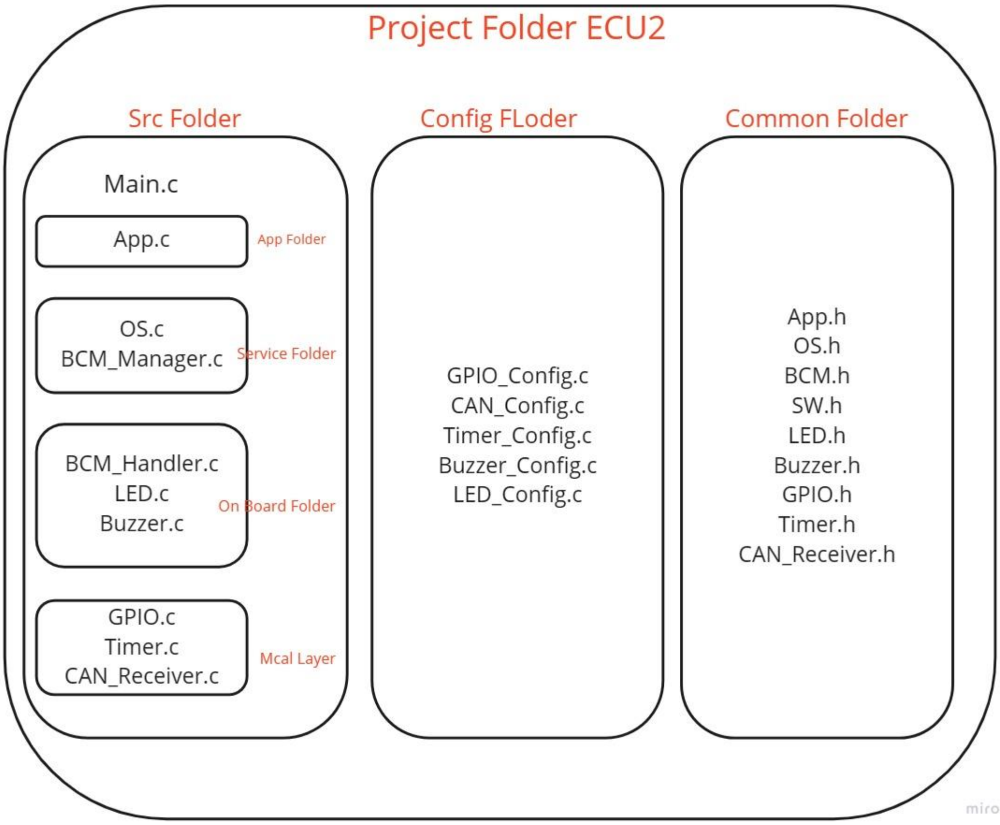
API Prototype	Parameters	Return	Description
Void GPIO_SetDirection(uint8 port , uint8 pin , PinDir_t pinDir)	Uint8 port name Uint8 pin umber PinDir_t pinDir Input or Output	Not return	It will set direction of any pin at any wanted port
Void GPIO_SetValue(uint8 port , uint8 pin , PinState_t pinState)	Uint8 port name Uint8 pin umber PinState_t pinState high or low	Not return	It will set state of pin at any wanted register

MCAL Layer : Timer

API Prototype	Parameters	Return	Description
Void Timer_Init(TimerType_t tType , TimerMode_t tMode)	TimerType_t tType TimerMode_t tMode	Not return	It will Set Timer Configuration based on Timer type and timer mode
Void TimerStart(uint32 StartValue)	uint32 StartValue of Timer Counter Register	Not return	It will set start value of the timer and setting prescaller of it based on wanted time
Void TimerStop(void)	No parameters	No return	It will stop Timer count
Uint32 TimerRead(void)	No parameters	Timer Counter Register Value	It will return Timer Counter Register Value

MCAL Layer : CAN Receiver Unit			
API Prototype	**Parameters**	**Return**	**Description**
Void CAN_Init(void)	No parameters	Not return	It will Set Can Uint Configurations
Uint32 CAN_Receive(void)	No parameters	uint32 Data Received over CAN	It will Receive data from CAN Transmitter Unit

4. Folder Structure


Project Folder ECU2
Src Folder
Main.c
App.c
App Folder
OS.c
BCM_Manager.c
Service Folder
BCM_Handler.c
LED.c
Buzzer.c
On Board Folder
GPIO.c
Timer.c
CAN_Receiver.c
Mcal Layer
Config FLoder
GPIO_Config.c
CAN_Config.c
Timer_Config.c
Buzzer_Config.c
LED_Config.c
Common Folder
App.h
OS.h
BCM.h
SW.h
LED.h
Buzzer.h
GPIO.h
Timer.h
CAN_Receiver.h
miro