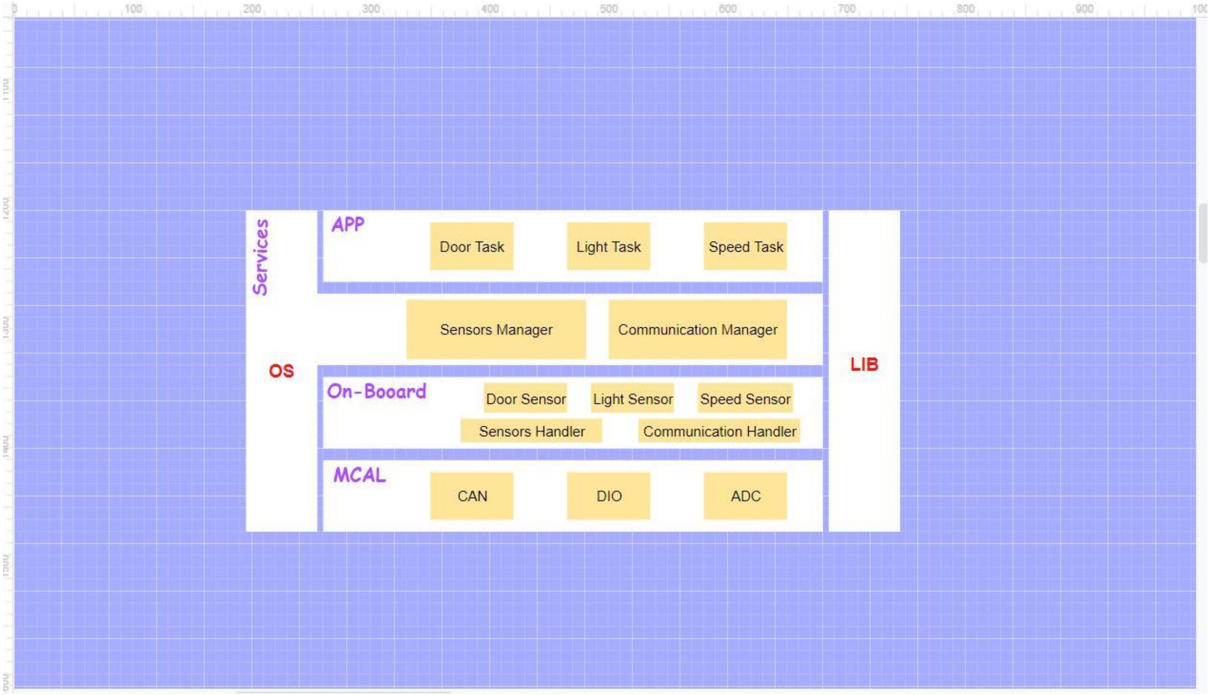
Static Design

ECU_1:

Layered Architecture:



Modules:

CAN

APIs:

Name	CAN_Init()
Syntax	**void CAN_Init (Can_ConfigType *config)**
Synchronization	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	Config
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	Initialize CAN module

Name	CAN_DeInit()
Syntax	**void CAN_DeInit (void)**
Synchronization	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None

Return	None
Description	De-initialize CAN module

Name	CAN_Send()
Syntax	**void Can_Send (u32 message)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	message
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	Sends Data through CAN module

Typdefs:

Name	CAN_ConfigType
Type	Struct
Description	A struct that holds all of the peripheral initialization

DIO

APIs:

Name	Dio_ReadChannel ()
Syntax	**Dio_LevelType Dio_ReadChannel(Dio_ChannelType channelId)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	channelId
Parameters (inout)	None
Parameters (out)	None
Return	Dio_LevelType
Description	Read a specific pin state

Name	Dio_WriteChannel ()
Syntax	**void Dio_WriteChannel(Dio_ChannelType channelId, Dio_LevelType level)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	channelId, level
Parameters (inout)	None
Parameters (out)	None
Return	Dio_LevelType
Description	Write on a specific pin

Typdefs:

Name	Dio_ChannelType
Type	u8
Description	identifies which pin on the MCU we need to write to or read from

Name	Dio_LevelType
Type	u8
Description	identifies The state of the pin either high or low

ADC

APIs:

Name	ADC_Init ()
Syntax	**void ADC_Init(void)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	Initialize ADC Module

Name	ADC_DeInit ()
Syntax	**void ADC_DeInit(void)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	De-Initialize ADC Module

Name	ADC_Read ()
Syntax	**u16 Adc_ReadData(u8 channel)**
Synchronization	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	channel
Parameters (inout)	None
Parameters (out)	None
Return	u16
Description	Get the current reading from the ADC

Speed Sensor

APIs:

Name	SpeedSensor_GetVal ()
Syntax	**u16 SpeedSensor_GetVal (void)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return	u16
Description	Gets the current speed value

Name	DoorSensor_GetState()
Syntax	**DoorSens_StateType DoorSensor_GetState (void)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return	DoorSens_StateType
Description	Gets the current state of the door

Typdefs:

Name	DoorSens_StateType
Type	enum
Description	identifies the state of the door either OPEN or CLOSED

Light Sensor

APIs:

Name	LightSensor_GetState()
Syntax	**LightSens_StateType LightSensor_GetState (void)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return	LightSens_StateType
Description	Gets the current state of the light switch

Typdefs:

Author: Esraa Ahmed

Name	LightSens_StateType
Type	enum
Description	identifies the state of the light either ON or OFF

Sensor Handler

APIs:

Name	Sensor_Handler_Select
Syntax	**u32 Sensor_Handler_Select (u8 Id)**
Synchronization	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	Id
Parameters (inout)	None
Parameters (out)	None
Return	u32
Description	Chooses which sensor to read the data from

Communication Handler

APIs:

Name	Communication_Handler_Send
Syntax	**void Communication_Handler_Send (u32 msg, u8 Id)**
Synchronization	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	msg, Id
Parameters (inout)	None
Parameters (out)	None
Return	u32
Description	Chooses which bus or peripheral through which it sends the data

Sensors Manager

APIs:

Name	Sensor_Manager_Select
Syntax	**u32 Sensor_Manager_Select (u8 Id)**
Synchronization	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	Id
Parameters (inout)	None
Parameters (out)	None
Return	u32
Description	An interface for the application layer to Chooses which sensor to read the data from

Communication Manager

APIs:

Name	Communication_Manager_Send
Syntax	**void Communication_Manager_Send (u8 Id, u32 msg)**
Synchronization	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	Id, msg
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	An interface for the application layer to Chooses which bus or peripheral through which it sends the data

Light Task

APIs:

Name	task_SendLightSwitchState
Syntax	**void task_SendLightSwitchState(void)**
Synchronization	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	A task that sends the light switch state through the can protocol to another ECU every 20ms

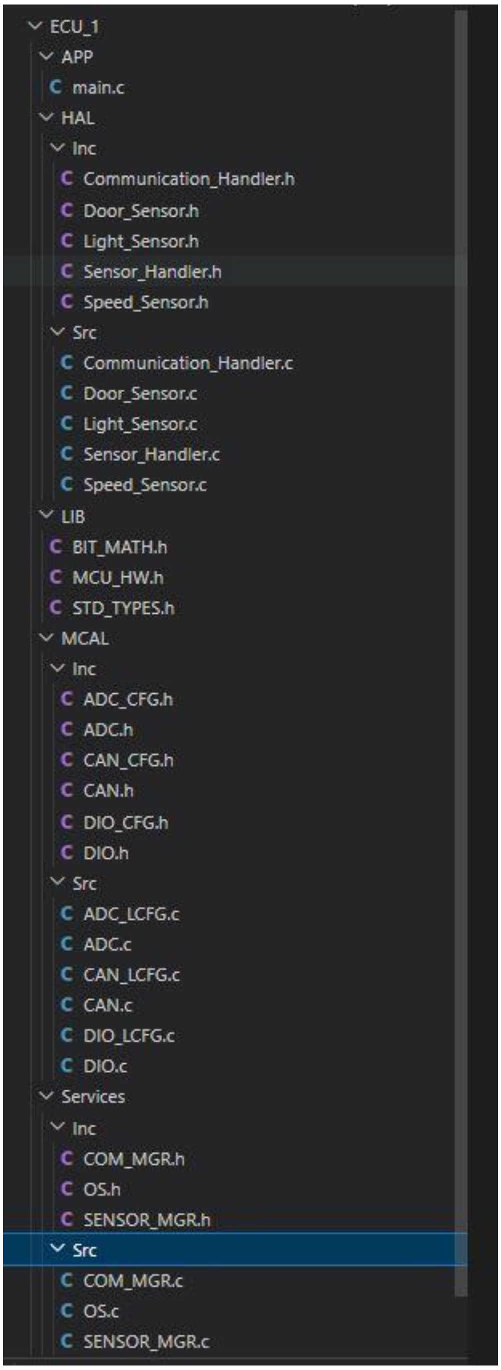
Speed Task

APIs:

Name	task_SendSpeedState
Syntax	**void task_SendSpeedState(void)**
Synchronization	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	A task that sends the speed state through the can protocol to another ECU every 5ms

Author: Esraa Ahmed

Folder Structure:

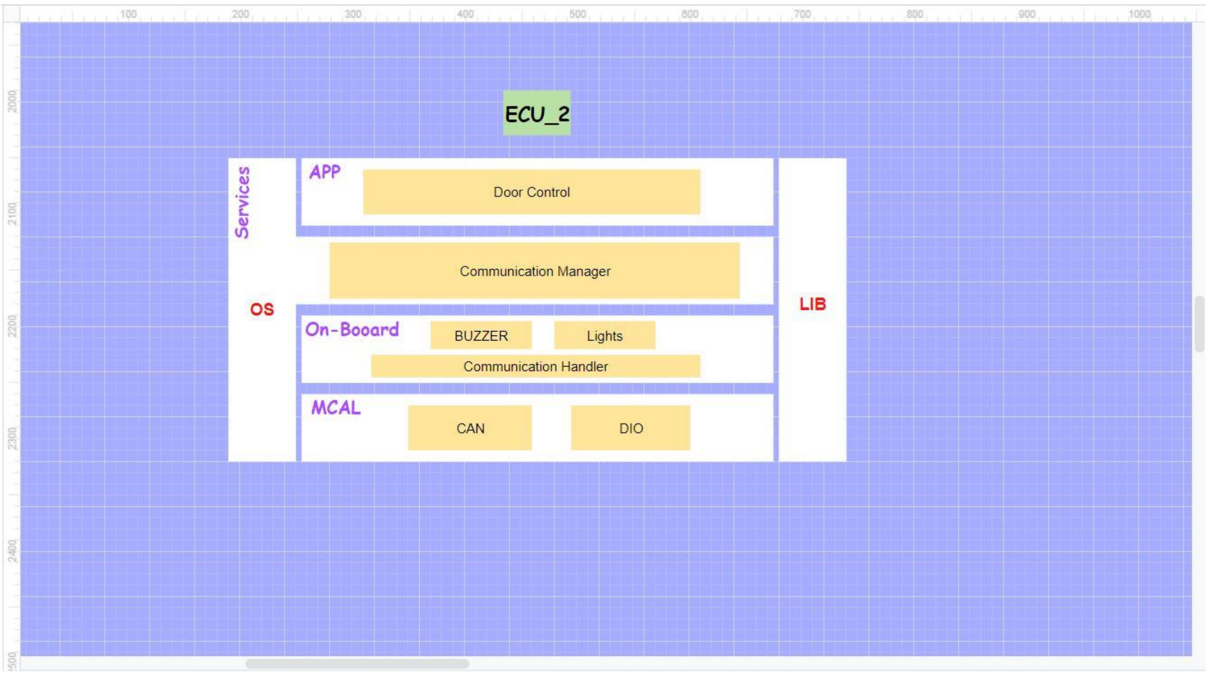


```
ECU_1
  APP
    main.c
  HAL
    Inc
      Communication_Handler.h
      Door_Sensor.h
      Light_Sensor.h
      Sensor_Handler.h
      Speed_Sensor.h
    Src
      Communication_Handler.c
      Door_Sensor.c
      Light_Sensor.c
      Sensor_Handler.c
      Speed_Sensor.c
  LIB
    BIT_MATH.h
    MCU_HW.h
    STD_TYPES.h
  MCAL
    Inc
      ADC_CFG.h
      ADC.h
      CAN_CFG.h
      CAN.h
      DIO_CFG.h
      DIO.h
    Src
      ADC_LCFG.c
      ADC.c
      CAN_LCFG.c
      CAN.c
      DIO_LCFG.c
      DIO.c
  Services
    Inc
      COM_MGR.h
      OS.h
      SENSOR_MGR.h
    Src
      COM_MGR.c
      OS.c
      SENSOR_MGR.c
```

Author: Esraa Ahmed

ECU_2:

Layered Architecture:



Modules:

CAN

APIs:

Name	CAN_Init()
Syntax	**void CAN_Init (Can_ConfigType *config)**
Synchronization	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	Config
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	Initialize CAN module

Name	CAN_DeInit()
Syntax	**void CAN_DeInit (void)**
Synchronization	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	De-initialize CAN module

Author: Esraa Ahmed

Name	CAN_Send()
Syntax	**void Can_Send (u32 message)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	message
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	Sends Data through CAN module

Typdefs:

Name	CAN_ConfigType
Type	Struct
Description	A struct that holds all of the peripheral initialization

DIO

APIs:

Name	Dio_ReadChannel ()
Syntax	**Dio_LevelType Dio_ReadChannel(Dio_ChannelType channelId)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	channelId
Parameters (inout)	None
Parameters (out)	None
Return	Dio_LevelType
Description	Read a specific pin state

Name	Dio_WriteChannel ()
Syntax	**void Dio_WriteChannel(Dio_ChannelType channelId, Dio_LevelType level)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	channelId, level
Parameters (inout)	None
Parameters (out)	None
Return	Dio_LevelType
Description	Write on a specific pin

Typdefs:

Name	Dio_ChannelType
Type	u8
Description	identifies which pin on the MCU we need to write to or read from

Author: Esraa Ahmed

Name	Dio_LevelType
Type	u8
Description	identifies The state of the pin either high or low

Lights

APIs:

Name	Lights_SetState
Syntax	**void lights_SetState(u8 Id, u8 level)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	Id, level
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	It sets the light determined by the provided ID to either on or off

Buzzer

APIs:

Name	Buzzer_SetState
Syntax	**void Buzzer_SetState(u8 level)**
Synchronization	Synchronous
Reentrancy	Reentrant
Parameters (in)	level
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	It sets the buzzer to either ON or OFF

Communication Handler

APIs:

Name	Communication_Handler_Send
Syntax	**void Communication_Handler_Send (u32 msg, u8 Id)**
Synchronization	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	msg, Id
Parameters (inout)	None
Parameters (out)	None
Return	u32
Description	Chooses which bus or peripheral through which it sends the data

Communication Manager

APIs:

Name	Communication_Manager_Send
Syntax	**void Communication_Manager_Send (u8 Id, u32 msg)**
Synchronization	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	Id, msg
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	An interface for the application layer to Chooses which bus or peripheral through which it sends the data

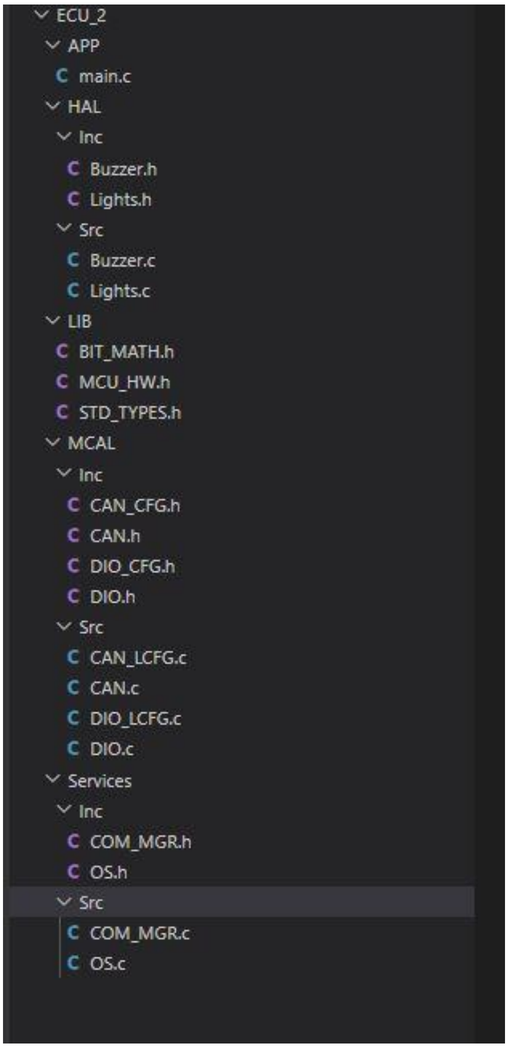
Door Control

APIs:

Name	DoorCtrl_UpdateState
Syntax	**void DoorCtrl_UpdateState (void)**
Synchronization	Synchronous
Reentrancy	Non-Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return	None
Description	A function That updates the door control module state with the newly received state from ECU1 and acts on the lights and the buzzer accordingly

Author: Esraa Ahmed

Folder Structure:



```
ECU_2
  APP
    main.c
  HAL
    Inc
      Buzzer.h
      Lights.h
    Src
      Buzzer.c
      Lights.c
  LIB
    BIT_MATH.h
    MCU_HW.h
    STD_TYPES.h
  MCAL
    Inc
      CAN_CFG.h
      CAN.h
      DIO_CFG.h
      DIO.h
    Src
      CAN_LCFG.c
      CAN.c
      DIO_LCFG.c
      DIO.c
  Services
    Inc
      COM_MGR.h
      OS.h
    Src
      COM_MGR.c
      OS.c
```

Author: Esraa Ahmed