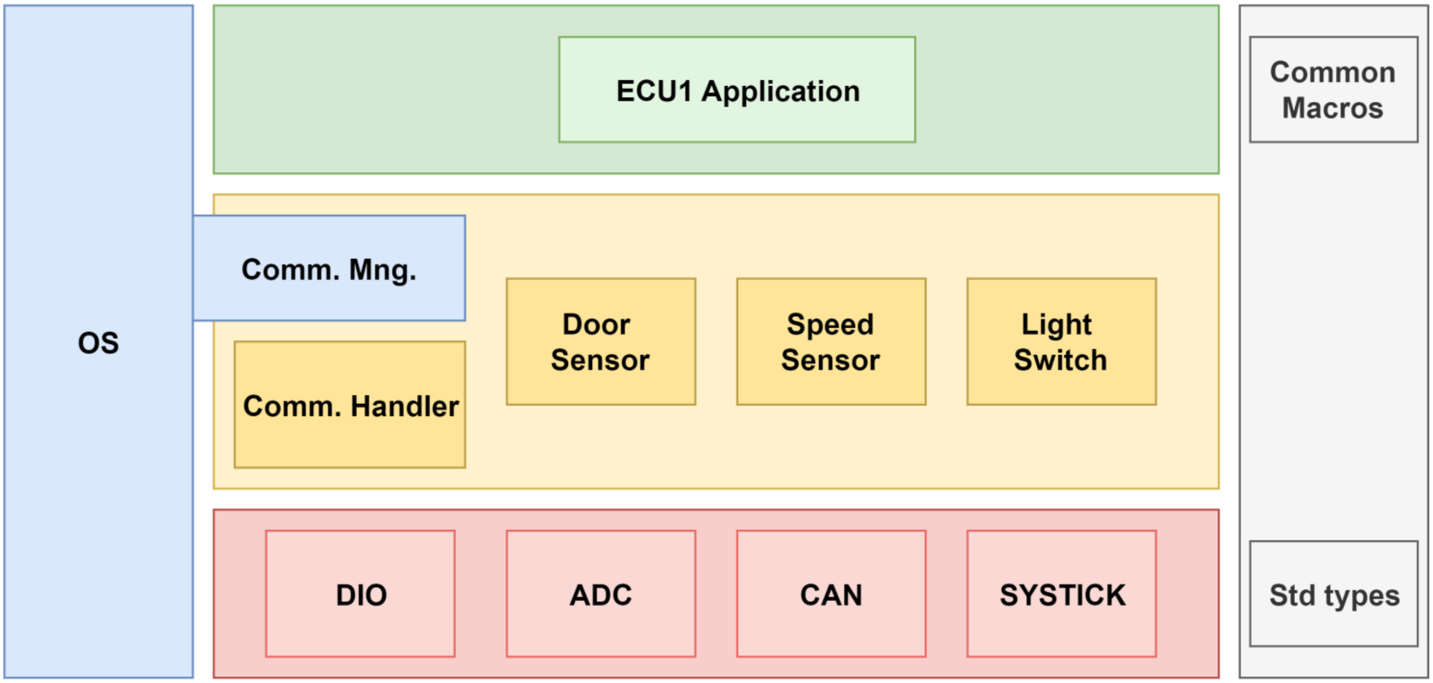
Project Static Design

Name	Mohab Osamah Sharafeldin
Email	mohabosamah2011@yahoo.com
Documentation target	Delivering a fully static design for Automotive door control system project

ECU1 Layered Architecture:



Application Layer:

- ECU1 Application

Hardware Abstraction Layer:

- Door Sensor
- Speed Sensor
- Light Switch
- Comm. Handler

Microcontroller Abstraction Layer:

- Digital Input Output (DIO)
- SysTick Timer (SYSTICK)
- Analog Digital Converter (ADC)
- Controller Area Network protocol (CAN)

Libraries:

- Common Macros
- Std types

Microcontroller Abstraction Layer (MCAL):

- **DIO Module APIs and typedefs**

Syntax	void DIO_Channel_Init (DIO_ConfigType ChannelConfig)
Description	Initialize the DIO Channel
Type	Init API
Parameters (in)	DIO_ConfigType ChannelConfig
Parameters (out)	None
Return	None

Syntax	void DIO_Write_Channel (DIO_ChannelType Channel, DIO_ChannelLevel Level)
Description	Write the DIO Channel
Type	Setter API
Parameters (in)	DIO_ChannelType Channel
	DIO_ChannelLevel Level
Parameters (out)	None
Return	None

Syntax	DIO_ChannelLevel DIO_Read_Channel (DIO_ChannelType Channel)
Description	Read the DIO Channel
Type	Getter API
Parameters (in)	DIO_ChannelType Channel
Parameters (out)	DIO_ChannelLevel
Return	Return The Channel level, High or Low

Name	DIO_ConfigType
Description	A struct which has all the configurations needed to configure the Channel
Type	struct

Name	DIO_ChannelType
Description	A struct which has the Channel info: Port ID, Pin ID and Channel level
Type	struct

<table>
<tr><td>Name</td><td colspan="2">DIO_ChannelLevel</td></tr>
<tr><td rowspan="3">Description</td><td colspan="2">An enum for the Channel level</td></tr>
<tr><td>LOW</td><td>0</td></tr>
<tr><td>HIGH</td><td>1</td></tr>
<tr><td>Type</td><td colspan="2">enum</td></tr>
</table>

- SYSTICK Module APIs and typedefs

Syntax	void SYSTICK_Init (SYSTICK_ConfigType SYSTICK_Config)
Description	Initialize the SysTick Timer
Type	Init API
Parameters (in)	SYSTICK_ConfigType SYSTICK_Config
Parameters (out)	None
Return	None

Name	SYSTICK_ConfigType
Description	A struct which has all the configurations needed to configure the SysTick Timer
Type	struct

- ADC Module APIs and typedefs

Syntax	void ADC_Init (ADC_ConfigType ADC_Config)
Description	Initialize the ADC Channel
Type	Init API
Parameters (in)	ADC_ConfigType ADC_Config
Parameters (out)	None
Return	None

Syntax	void ADC_Read (ADC_ChannelType ADC_Channel)
Description	Read the ADC Channel
Type	Getter API
Parameters (in)	ADC_ChannelType ADC_Channel
Parameters (out)	None
Return	None

Name	ADC_ConfigType
Description	A struct which has all the configurations needed to configure ADC Channel
Type	struct

Name	ADC_ChannelType
Description	Typedef that describes the ADC Channels
Type	uint32

- CAN Module APIs and typedefs

Syntax	void CAN_Init (CAN_ConfigType CAN_Config)
Description	Initialize the CAN Module
Type	Init API
Parameters (in)	CAN_ConfigType CAN_Config
Parameters (out)	None
Return	None

Syntax	void CAN_ SetBaudrate (uint8 Controller, uint16 BaudRateConfigID)
Description	Set the baud rate configuration of the CAN Controller
Type	
Parameters (in)	uint8 Controller
	uint16 BaudRateConfigID
Parameters (out)	None
Return	None

Syntax	void CAN_WriteData (uint32 CAN_Data)
Description	Write Data on CAN Bus
Type	Sender API
Parameters (in)	uint32 CAN_Data
Parameters (out)	None
Return	None

Syntax	**void CAN_ReadData (void)**
Description	**Read Data from CAN Bus**
Type	**Receiver API**
Parameters (in)	**None**
Parameters (out)	**None**
Return	**None**

Name	**CAN_ConfigType**
Description	**A struct which has all the configurations needed to configure CAN Module**
Type	**struct**

Hardware Abstraction Layer (HAL):

- Speed Sensor Module APIs and typedefs

Syntax	void SpeedSensor_Init (SpeedSensor_ConfigType SpeedSensor_Config)
Description	Initialize the Speed Sensor Module
Type	Init API
Parameters (in)	SpeedSensor_ConfigType SpeedSensor_Config
Parameters (out)	None
Return	None

Syntax	uint32 SpeedSensor_Read (void)
Description	Read the Speed Sensor Module
Type	Getter API
Parameters (in)	None
Parameters (out)	uint32
Return	Return The Speed Sensor Value

Name	SpeedSensor_ConfigType
Description	A struct which has all the configurations needed to configure Speed Sensor Module
Type	struct

Door Sensor Module APIs and typedefs

Syntax	void DoorSensor_Init (DoorSensor_ConfigType DoorSensor_Config)
Description	Initialize the Door Sensor Module
Type	Init API
Parameters (in)	DoorSensor_ConfigType DoorSensor_Config
Parameters (out)	None
Return	None

Syntax	DoorSensor_Status DoorSensor_Read (void)
Description	Read the Door Sensor Module
Type	Getter API
Parameters (in)	None
Parameters (out)	DoorSensor_Status
Return	Return The Door Sensor Status, Opened or Closed

Name	DoorSensor_ConfigType
Description	A struct which has all the configurations needed to configure Door Sensor Module
Type	struct

Name	DoorSensor_Status	
Description	An enum for the Door Sensor Status	
	Closed	0
	Opened	1
Type	enum	

- Light Switch Module APIs and typedefs

Syntax	void LightSwitch_Init (LightSwitch_ConfigType LightSwitch_Config)
Description	Initialize the Light Switch Module
Type	Init API
Parameters (in)	LightSwitch_ConfigType LightSwitch_Config
Parameters (out)	None
Return	None

Syntax	LightSwitch_Status LightSwitch_Read (void)
Description	Read the Light Switch Module
Type	Getter API
Parameters (in)	None
Parameters (out)	LightSwitch_Status
Return	Return The Light Switch Status, Pressed or NOT Pressed

Name	LightSwitch_ConfigType
Description	A struct which has all the configurations needed to configure Light Switch Module
Type	struct

Name	LightSwitch_Status	
Description	An enum for the Light Switch Status	
	NOT Pressed	0
	Pressed	1
Type	enum	

- Comm. Handler Module APIs and typedefs

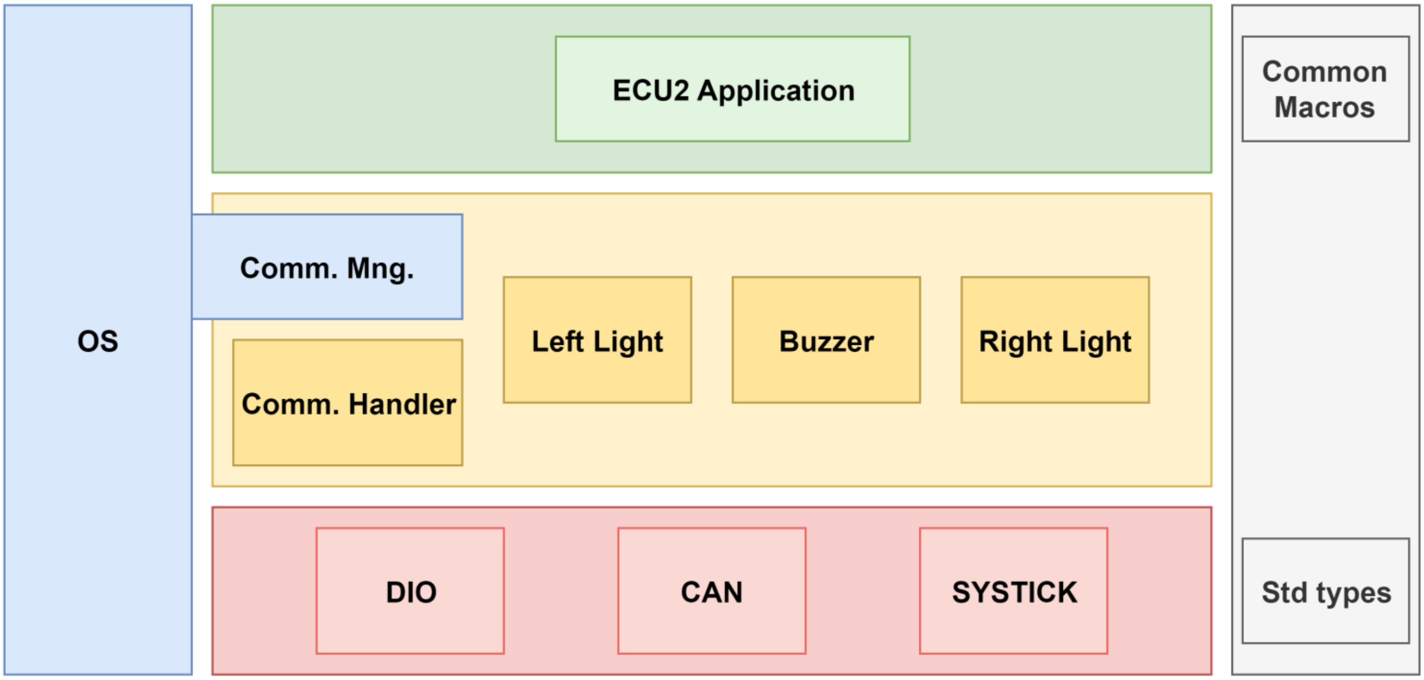
Syntax	void CommHandler_Init (CommHandler_ConfigType CommHandler_Config)
Description	Initialize the Comm. Handler Module
Type	Init API
Parameters (in)	CommHandler_ConfigType CommHandler_Config
Parameters (out)	None
Return	None

Syntax	void BCM_Handler (uint32 Message, uint8 Bus)
Description	Send the required message on the selected Bus type
Type	Sender API
Parameters (in)	uint32 Message
	uint8 Bus
Parameters (out)	None
Return	None

Syntax	uint32 BCM_HandleRead (uint8 Bus)
Description	Read the message from the selected Bus
Type	Receiver API
Parameters (in)	uint8 Bus
Parameters (out)	uint32
Return	Return the message, size of uint32

Name	CommHandler_ConfigType
Description	A struct which has all the configurations needed to configure Comm. Handler Module
Type	struct

ECU2 Layered Architecture:



Application Layer:

- ECU2 Application

Hardware Abstraction Layer:

- Left Light
- Buzzer
- Right Light
- Comm. Handler

Microcontroller Abstraction Layer:

- Digital Input Output (DIO)
- SysTick Timer (SYSTICK)
- Controller Area Network protocol (CAN)

Libraries:

- Common Macros
- Std types

Microcontroller Abstraction Layer (MCAL):

- DIO Module APIs and typedefs

Syntax	void DIO_Channel_Init (DIO_ConfigType ChannelConfig)
Description	Initialize the DIO Channel
Type	Init API
Parameters (in)	DIO_ConfigType ChannelConfig
Parameters (out)	None
Return	None

Syntax	void DIO_Write_Channel (DIO_ChannelType Channel, DIO_ChannelLevel Level)
Description	Write the DIO Channel
Type	Setter API
Parameters (in)	DIO_ChannelType Channel
	DIO_ChannelLevel Level
Parameters (out)	None
Return	None

Syntax	**DIO_ChannelLevel DIO_Read_Channel (DIO_ChannelType Channel)**
Description	**Read the DIO Channel**
Type	**Getter API**
Parameters (in)	**DIO_ChannelType Channel**
Parameters (out)	**DIO_ChannelLevel**
Return	**Return The Channel level, High or Low**

Name	**DIO_ConfigType**
Description	**A struct which has all the configurations needed to configure the Channel**
Type	**struct**

Name	**DIO_ChannelType**
Description	**A struct which has the Channel info: Port ID, Pin ID and Channel level**
Type	**struct**

<table>
<tr><td><b>Name</b></td><td colspan="2"><b>DIO_ChannelLevel</b></td></tr>
<tr><td rowspan="3"><b>Description</b></td><td colspan="2"><b>An enum for the Channel level</b></td></tr>
<tr><td><b>LOW</b></td><td><b>0</b></td></tr>
<tr><td><b>HIGH</b></td><td><b>1</b></td></tr>
<tr><td><b>Type</b></td><td colspan="2"><b>enum</b></td></tr>
</table>

- SYSTICK Module APIs and typedefs

Syntax	**void SYSTICK_Init (SYSTICK_ConfigType SYSTICK_Config)**
Description	**Initialize the SysTick Timer**
Type	**Init API**
Parameters (in)	**SYSTICK_ConfigType SYSTICK_Config**
Parameters (out)	**None**
Return	**None**

Name	**SYSTICK_ConfigType**
Description	**A struct which has all the configurations needed to configure the SysTick Timer**
Type	**struct**

CAN Module APIs and typedefs

Syntax	**void CAN_Init (CAN_ConfigType CAN_Config)**
Description	**Initialize the CAN Module**
Type	**Init API**
Parameters (in)	**CAN_ConfigType CAN_Config**
Parameters (out)	**None**
Return	**None**

Syntax	**void CAN_ SetBaudrate (uint8 Controller, uint16 BaudRateConfigID)**
Description	**Set the baud rate configuration of the CAN Controller**
Type	
Parameters (in)	**uint8 Controller**
	uint16 BaudRateConfigID
Parameters (out)	**None**
Return	**None**

Syntax	**void CAN_WriteData (uint32 CAN_Data)**
Description	**Write Data on CAN Bus**
Type	**Sender API**
Parameters (in)	**uint32 CAN_Data**
Parameters (out)	**None**
Return	**None**

Syntax	**void CAN_ReadData (void)**
Description	**Read Data from CAN Bus**
Type	**Receiver API**
Parameters (in)	**None**
Parameters (out)	**None**
Return	**None**

Name	**CAN_ConfigType**
Description	**A struct which has all the configurations needed to configure CAN Module**
Type	**struct**

Hardware Abstraction Layer (HAL):

- Buzzer Module APIs and typedefs

Syntax	void Buzzer_Init (Buzzer_ConfigType Buzzer_Config)
Description	Initialize the Buzzer Module
Type	Init API
Parameters (in)	Buzzer_ConfigType Buzzer_Config
Parameters (out)	None
Return	None

Syntax	void Buzzer_ChangeStatus (Buzzer_StatusType BuzzerStatus)
Description	Change the Buzzer status based on the written data, ON or OFF
Type	Setter API
Parameters (in)	Buzzer_StatusType BuzzerStatus
Parameters (out)	None
Return	None

Name	Buzzer_ConfigType
Description	A struct which has all the configurations needed to configure Buzzer Module
Type	struct

Name	Buzzer_StatusType	
Description	An enum for the Buzzer Status	
	OFF	0
	ON	1
Type	Enum	

- Left Light Module APIs and typedefs

Syntax	void LLight_Init (LLight_ConfigType LLight_Config)
Description	Initialize the Left Light Module
Type	Init API
Parameters (in)	LLight_ConfigType LLight_Config
Parameters (out)	None
Return	None

Syntax	void LLight_ChangeStatus (LLight_StatusType LLightStatus)
Description	Change the Left Light status based on the written data, ON or OFF
Type	Setter API
Parameters (in)	LLight_StatusType LLightStatus
Parameters (out)	None
Return	None

Name	LLight_ConfigType
Description	A struct which has all the configurations needed to configure Left Light Module
Type	struct

Name	LLight_StatusType	
Description	An enum for the Left Light Status	
	OFF	0
	ON	1
Type	enum	

- Right Light Module APIs and typedefs

Syntax	void RLight_Init (RLight_ConfigType RLight_Config)
Description	Initialize the Right Light Module
Type	Init API
Parameters (in)	RLight_ConfigType RLight_Config
Parameters (out)	None
Return	None

Syntax	void RLight_ChangeStatus (RLight_StatusType RLightStatus)
Description	Change the Right Light status based on the written data, ON or OFF
Type	Setter API
Parameters (in)	RLight_StatusType RLightStatus
Parameters (out)	None
Return	None

Name	RLight_ConfigType
Description	A struct which has all the configurations needed to configure Right Light Module
Type	struct

Name	RLight_StatusType	
Description	An enum for the Right Light Status	
	OFF	0
	ON	1
Type	enum	

Comm. Handler Module APIs and typedefs

Syntax	void CommHandler_Init (CommHandler_ConfigType CommHandler_Config)
Description	Initialize the Comm. Handler Module
Type	Init API
Parameters (in)	CommHandler_ConfigType CommHandler_Config
Parameters (out)	None
Return	None

Syntax	void BCM_Handler (uint32 Message, uint8 Bus)
Description	Send the required message on the selected Bus type
Type	Sender API
Parameters (in)	uint32 Message
	uint8 Bus
Parameters (out)	None
Return	None

Syntax	uint32 BCM_HandleRead (uint8 Bus)
Description	Read the message from the selected Bus
Type	Receiver API
Parameters (in)	uint8 Bus
Parameters (out)	uint32
Return	Return the message, size of uint32

Name	CommHandler_ConfigType
Description	A struct which has all the configurations needed to configure Comm. Handler Module
Type	struct