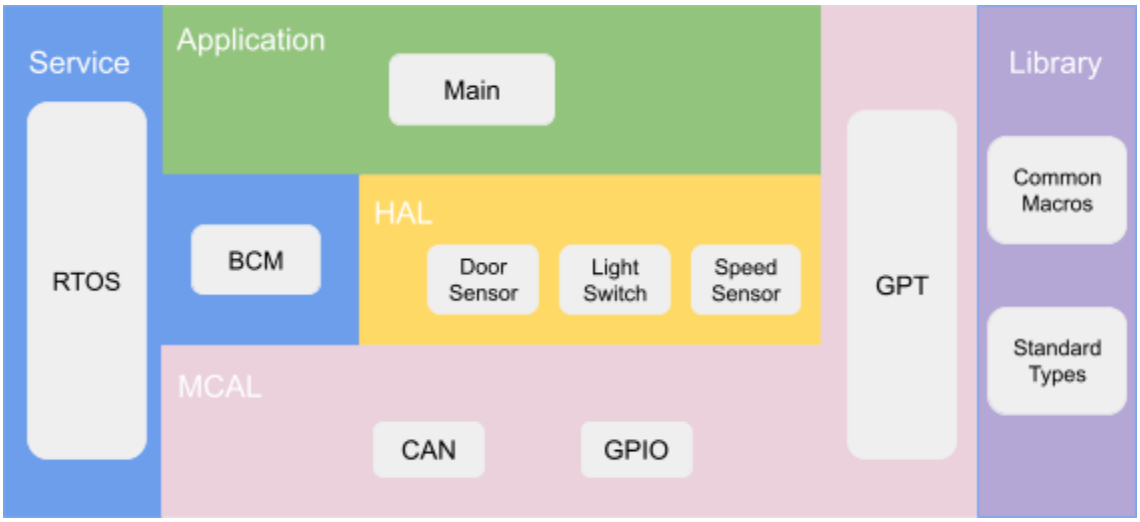
Name: Yasser Waleed Date: 12 Oct 2022

PROJECT 3
Automotive Door Control System Design
<u>Static Design</u>

- For ECU 1:

1. Make the layered architecture



2. Specify ECU components and modules

- Components: Door Sensor, Light Switch, Speed Sensor
- Modules:

Service (1):
- Basic Communication Module

HAL (1):
- Door Sensor Module
- Light Switch Module
- Speed Sensor Module

MCAL (1):
- GPIO Module
- CAN Module
- GPT Module

3. Provide full detailed APIs for each module as well as a detailed description for the used typedefs

Service (1): Basic Communication Module

APIs:

➔ BCM_Init (BCM_ConfigType * ConfigPtr)

Function Name	BCM_Init	
Argument(s)	*ConfigPtr	Points to a configuration struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the BCM	

➔ BCM_Send (uint8_t Data)

Function Name	BCM_Send	
Argument(s)	Data	The target data for sending
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Send data	

➔ BCM_Receive (void)

Function Name	BCM_Receive	
Argument(s)	-	-
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Receive data	

Typedefs:

Name	Type	Description
BCM_ConfigType	struct	Holds the set of configurations for the BCM

HAL (1): Door Sensor Module

APIs:

➔ DoorSensor_Init (Door_ConfigType * ConfigPtr)

Function Name	DoorSensor_Init	
Argument(s)	*ConfigPtr	Points to a configuration struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the Door Sensor	

➔ DoorSensor_Read (void)

Function Name	DoorSensor_Read	
Argument(s)	-	-
Return	Door_StateType State … opened or closed	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Read door state	

Typedefs:

Name	Type	Description
Door_StateType	enum	Defines OPENED and CLOSED
Door_ConfigType	struct	Holds the set of configurations for the Door Sensor

Name: Yasser Waleed Date: 12 Oct 2022

HAL (1): Light Switch Module

APIs:

➔ LightSwitch_Init (void)

Function Name	LightSwitch_Init	
Argument(s)	*ConfigPtr	Points to a configuration struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the Light Switch	

➔ LightSwitch_Read()

Function Name	LightSwitch_Read	
Argument(s)	-	-
Return	LSwitch_StateType … pressed or not pressed	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Read light switch state	

Typedefs:

Name	Type	Description
LSwitch_StateType	enum	Defines PRESSED and NOT PRESSED
LSwitch_ConfigType	struct	Holds the set of configurations for the Light Switch

Name: Yasser Waleed Date: 12 Oct 2022

HAL (1): Speed Sensor Module

➔ SpeedSensor_Init ()

Function Name	SpeedSensor_Init	
Argument(s)	*ConfigPtr	Points to a configuration struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the Speed Sensor	

➔ SpeedSensor_Read (void)

Function Name	SpeedSensor_Read	
Argument(s)	-	-
Return	Speed_StateType State … Moving or Stopped	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Read speed sensor state	

Typedefs:

Name	Type	Description
Speed_StateType	enum	Defines STOPPED and MOVING
Speed_ConfigType	struct	Holds the set of configurations for the Speed Sensor

Name: Yasser Waleed Date: 12 Oct 2022

MCAL (1): GPIO Module

APIs:

➔ GPIO_Init (GPIO_ConfigType * ConfigPtr)

Function Name	GPIO_Init	
Argument(s)	*ConfigPtr	Points to a configuration struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the GPIO port	

➔ GPIO_Write (GPIO_PortType Port, GPIO_PinType Pin, GPIO_LevelType Level)

Function Name	GPIO_Write	
Argument(s)	Port	The target port
	Pin	The target pin
	Level	High or Low
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Writes a level to a target GPIO pin	

➔ GPIO_Read(GPIO_PortType Port, GPIO_PinType Pin)

Function Name	GPIO_Write	
Argument(s)	Port	The target port
	Pin	The target pin
Return	Level (High or Low)	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Reads the level of a target GPIO pin	

Typedefs:

Name	Type	Description
GPIO_PortType	enum	Defines the available ports
GPIO_PinType	enum	Defines the available pins
GPIO_LevelType	enum	Defines LOW and HIGH
GPIO_ConfigType	struct	Holds the set of configurations for the GPIO

MCAL (1): CAN Module

APIs:

➔ CAN_Init (void)

Function Name	CAN_Init	
Argument(s)	*ConfigPtr	Points to a configuration struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the CAN	

➔ CAN_Transmit (CAN_ChannelType Channel, uint8_t Data)

Function Name	CAN_Transmit	
Argument(s)	Channel	The target channel
	Data	The target data for transmission
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Transmit data	

➔ CAN_Receive(CAN_ChannelType Channel)

Function Name	CAN_Receive	
Argument(s)	Channel	The target channel
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Receive data	

Typedefs:

Name	Type	Description
CAN_ChannelType	enum	Defines the available channels
CAN_ConfigType	struct	Holds the set of configurations for the CAN

MCAL (1): GPT Module

APIs:

➔ GPT_Init (GPT_ConfigType * ConfigPtr)

Function Name	GPT_Init	
Argument(s)	*ConfigPtr	Points to a config struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the timer	

➔ GPT_Start (GPT_ChannelType Channel, uint32 Time)

Function Name	GPT_Start	
Argument(s)	Channel	The target channel
	Time	The target time (tick count)
Return	-	
Reentrancy	Reentrant (but not for the same channel)	
Synchronicity	Synchronous	
Description	Starts the timer in a target channel	

➔ GPT_Stop (GPT_ChannelType Channel)

Function Name	GPT_Stop	
Argument(s)	Channel	The target channel
Return	-	
Reentrancy	Reentrant (but not for the same channel)	
Synchronicity	Synchronous	
Description	Stops the timer in a target channel	

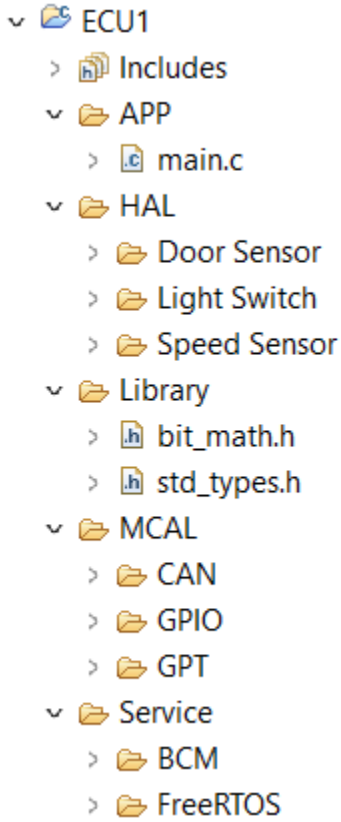
➔ GPT_NotificationCtrl (GPT_ChannelType Channel, GPT_ModeType Mode)

Function Name	GPT_NotificationCtrl	
Argument(s)	Channel	The target channel
	Mode	Enable / Disable
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Enable/Disable the interrupt for a target channel	

Typedefs:

Name	Type	Description
GPT_ConfigType	struct	Holds the set of configurations for the GPT
GPT_ChannelType	enum	Defines the available channels
GPT_ModeType	enum	Defines DISABLE and ENABLE

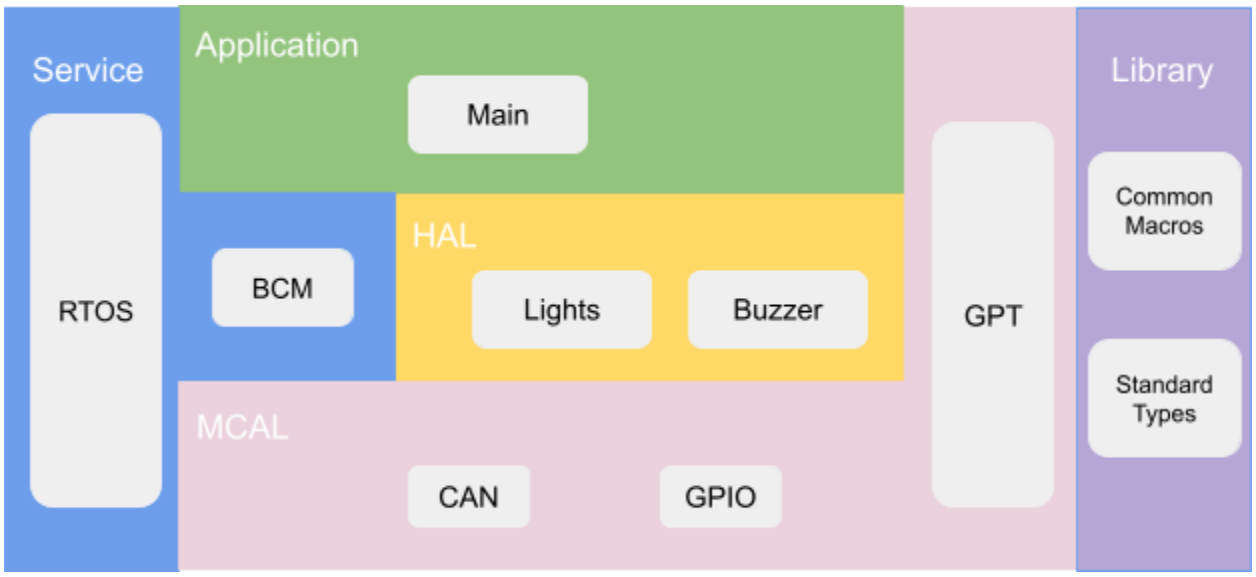
4. Prepare your folder structure according to the previous points



Name: Yasser Waleed Date: 12 Oct 2022

- For ECU 2:

1. Make the layered architecture



2. Specify ECU components and modules

   - Components: Buzzer, Lights (Right and Left lights are in sync, so treated as one output)
   - Modules:

     Service (2):
     - Basic Communication Module

     HAL (2):
     - Buzzer Module
     - Lights Module

     MCAL (2):
     - GPIO Module
     - CAN Module
     - GPT Module

3. Provide full detailed APIs for each module as well as a detailed description for the used typedefs

Service (2): Basic Communication Module

APIs:

➔ BCM_Init (void)

Function Name	BCM_Init	
Argument(s)	*ConfigPtr	Points to a configuration struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the BCM	

➔ BCM_Send (uint8_t Data)

Function Name	BCM_Send	
Argument(s)	Data	The target data for sending
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Send data	

➔ BCM_Receive (void)

Function Name	BCM_Receive	
Argument(s)	-	-
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Receive data	

Typedefs:

Name	Type	Description
BCM_ConfigType	struct	Holds the set of configurations for the BCM

HAL (2): Buzzer Module

a) Buzzer_Init (void)

Function Name	Buzzer_Init	
Argument(s)s	*ConfigPtr	Points to a configuration struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the Buzzer	

b) Buzzer_Write (Buzzer_ModeType Mode)

Function Name	Buzzer_Write	
Argument(s)	Mode	On or Off
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Switches the buzzer to the target mode	

Typedefs:

Name	Type	Description
Buzzer_ModeType	enum	Defines OFF and ON
Buzzer_ConfigType	struct	Holds the set of configurations for the Buzzer

Name: Yasser Waleed Date: 12 Oct 2022

HAL (2): Lights Module

➔ Lights_Init (void)

Function Name	Lights_Init	
Argument(s)s	*ConfigPtr	Points to a configuration struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the Lights	

➔ Lights_(Lights_ModeType Mode)

Function Name	Lights_Write	
Argument(s)	Mode	On or Off
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Switches the Lights to the target mode	

Typedefs:

Name	Type	Description
Lights_ModeType	enum	Defines OFF and ON
Lights_ConfigType	struct	Holds the set of configurations for the Lights

Name: Yasser Waleed Date: 12 Oct 2022

MCAL (2): GPIO Module

APIs:

➔ GPIO_Init (GPIO_ConfigType * ConfigPtr)

Function Name	GPIO_Init	
Argument(s)	*ConfigPtr	Points to a config struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the GPIO port	

➔ GPIO_Write (GPIO_PortType Port, GPIO_PinType Pin, GPIO_LevelType Level)

Function Name	GPIO_Write	
Argument(s)	Port	The target port
	Pin	The target pin
	Level	High or Low
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Writes a level to a target GPIO pin	

➔ GPIO_Read (GPIO_PortType Port, GPIO_PinType Pin)

Function Name	GPIO_Write	
Argument(s)	Port	The target port
	Pin	The target pin
Return	Level (High or Low)	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Reads the level of a target GPIO pin	

Typedefs:

Name	Type	Description
GPIO_PortType	enum	Defines the available ports
GPIO_PinType	enum	Defines the available pins
GPIO_LevelType	enum	Defines LOW and HIGH
GPIO_ConfigType	struct	Holds the set of configurations for the GPIO

MCAL (2): CAN Module

APIs:

➔ CAN_Init (void)

Function Name	CAN_Init	
Argument(s)	*ConfigPtr	Points to a configuration struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the CAN	

➔ CAN_Transmit (CAN_ChannelType Channel, uint8_t Data)

Function Name	CAN_Transmit	
Argument(s)	Channel	The target channel
	Data	The target data for transmission
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Transmit data	

➔ CAN_Receive (CAN_ChannelType Channel)

Function Name	CAN_Receive	
Argument(s)	Channel	The target channel
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Receive data	

Typedefs:

Name	Type	Description
CAN_ChannelType	enum	Defines the available channels
CAN_ConfigType	struct	Holds the set of configurations for the CAN

MCAL (2): GPT Module

APIs:

➔ GPT_Init (GPT_ConfigType * ConfigPtr)

Function Name	GPT_Init	
Argument(s)	*ConfigPtr	Points to a config struct
Return	-	
Reentrancy	Non-Reentrant	
Synchronicity	Synchronous	
Description	Initializes the timer	

➔ GPT_Start (GPT_ChannelType Channel, GPT_ValueType Time)

Function Name	GPT_Start	
Argument(s)	Channel	The target channel
	Time	The target time (tick count)
Return	-	
Reentrancy	Reentrant (but not for the same channel)	
Synchronicity	Synchronous	
Description	Starts the timer in a target channel	

➔ GPT_Stop (GPT_ChannelType Channel)

Function Name	GPT_Stop	
Argument(s)	Channel	The target channel
Return	-	
Reentrancy	Reentrant (but not for the same channel)	
Synchronicity	Synchronous	
Description	Stops the timer in a target channel	

➔ GPT_NotificationCtrl (GPT_ChannelType Channel, GPT_ModeType Mode)

Function Name	GPT_NotificationCtrl	
Argument(s)s	Channel	The target channel
	Mode	Enable / Disable
Return	-	
Reentrancy	Reentrant	
Synchronicity	Synchronous	
Description	Enable/Disable the interrupt for a target channel	

Typedefs:

Name	Type	Description
GPT_ValueType	uint_32	Holds the value of time as an integer.
GPT_ConfigType	struct	Holds the set of configurations for the GPT
GPT_ModeType	enum	Defines DISABLE and ENABLE
GPT_ConfigType	struct	Holds the set of configurations for the GPT

4. Prepare your folder structure according to the previous points