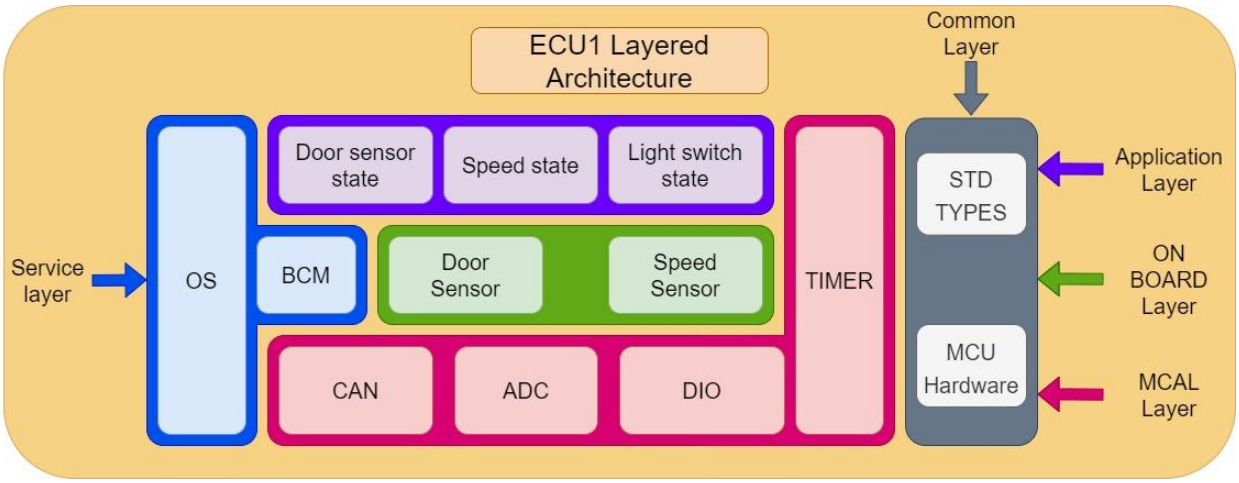
Embedded Software Engineering
Static Design

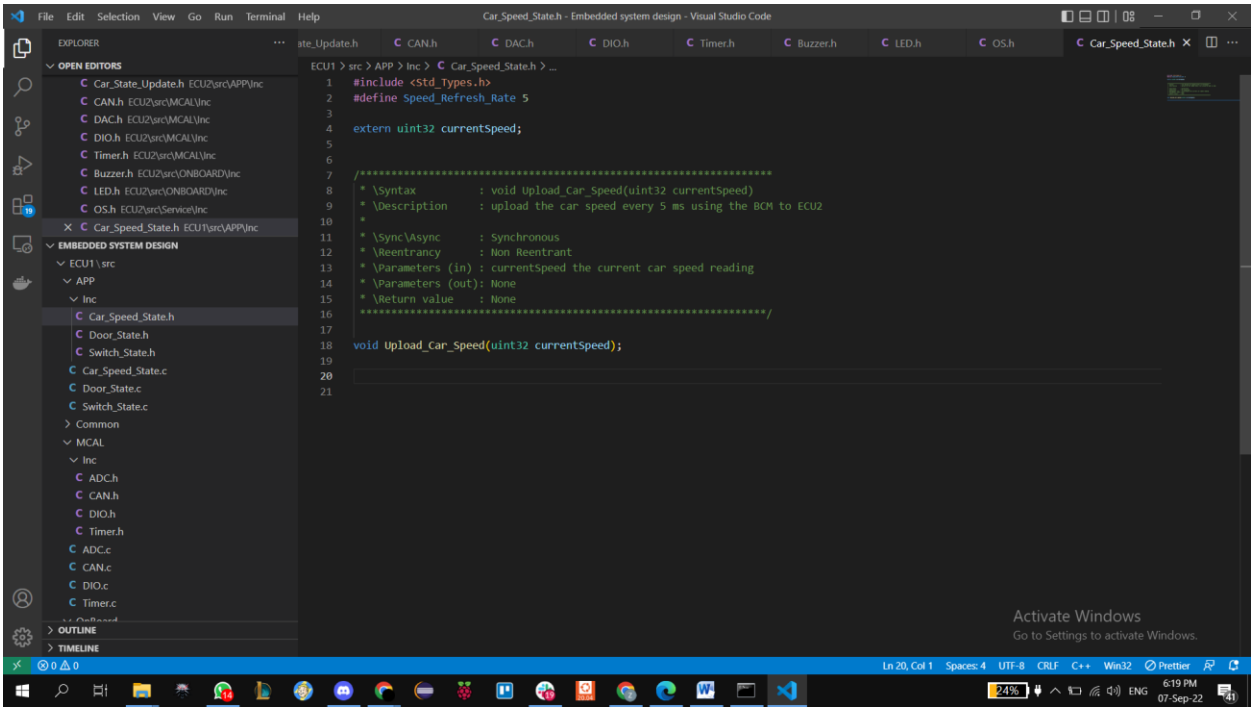
ECU1:

layered architecture:

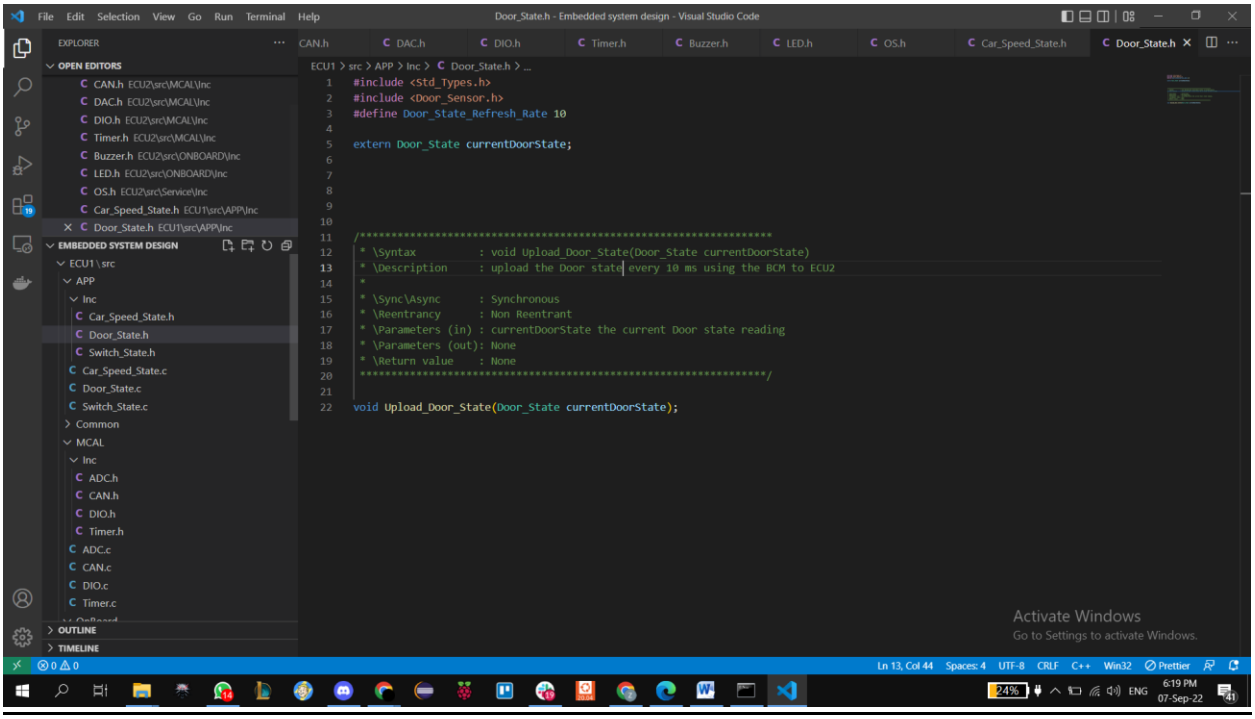


Full API details and File structure:

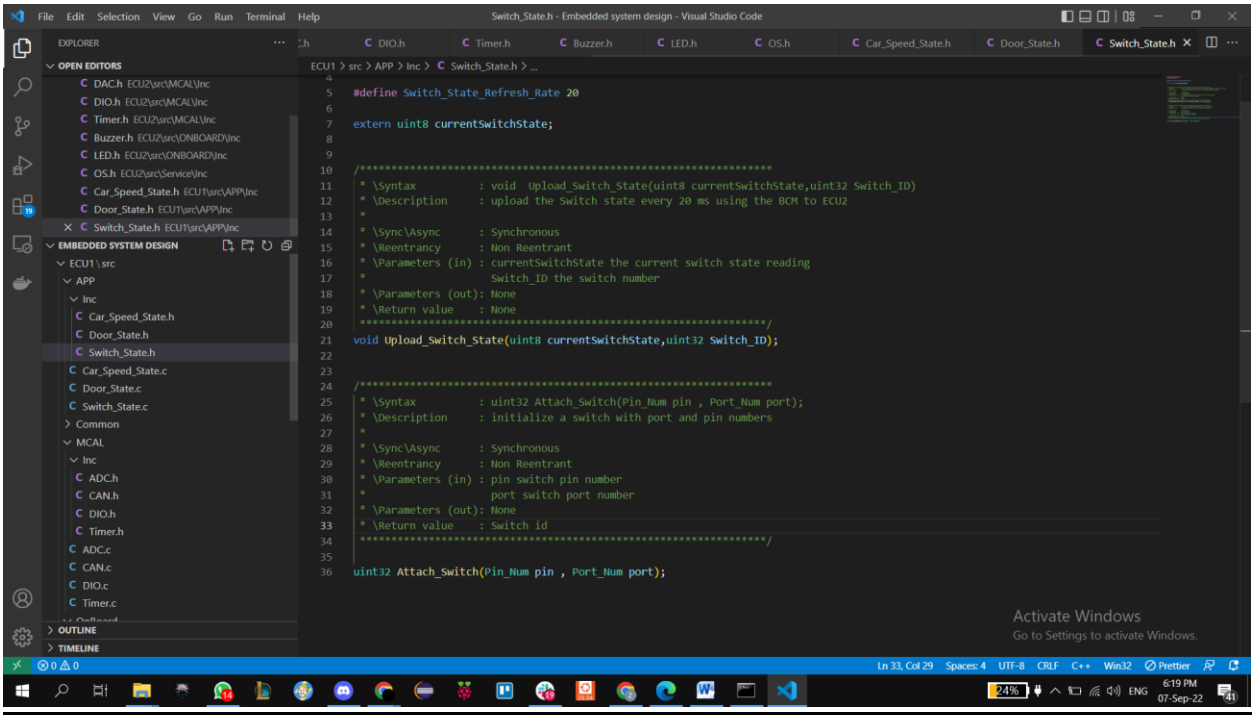
1.Car_Speed_state_module_APIs:



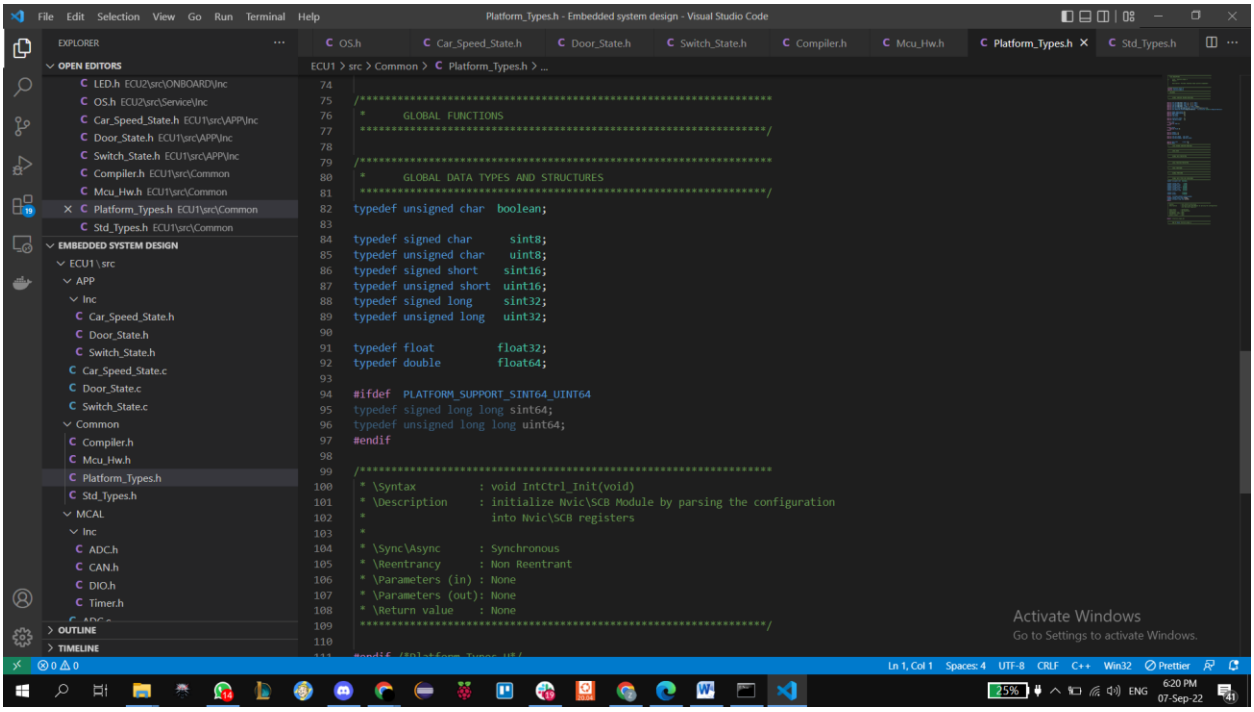
2. Door_state_module_APIs:



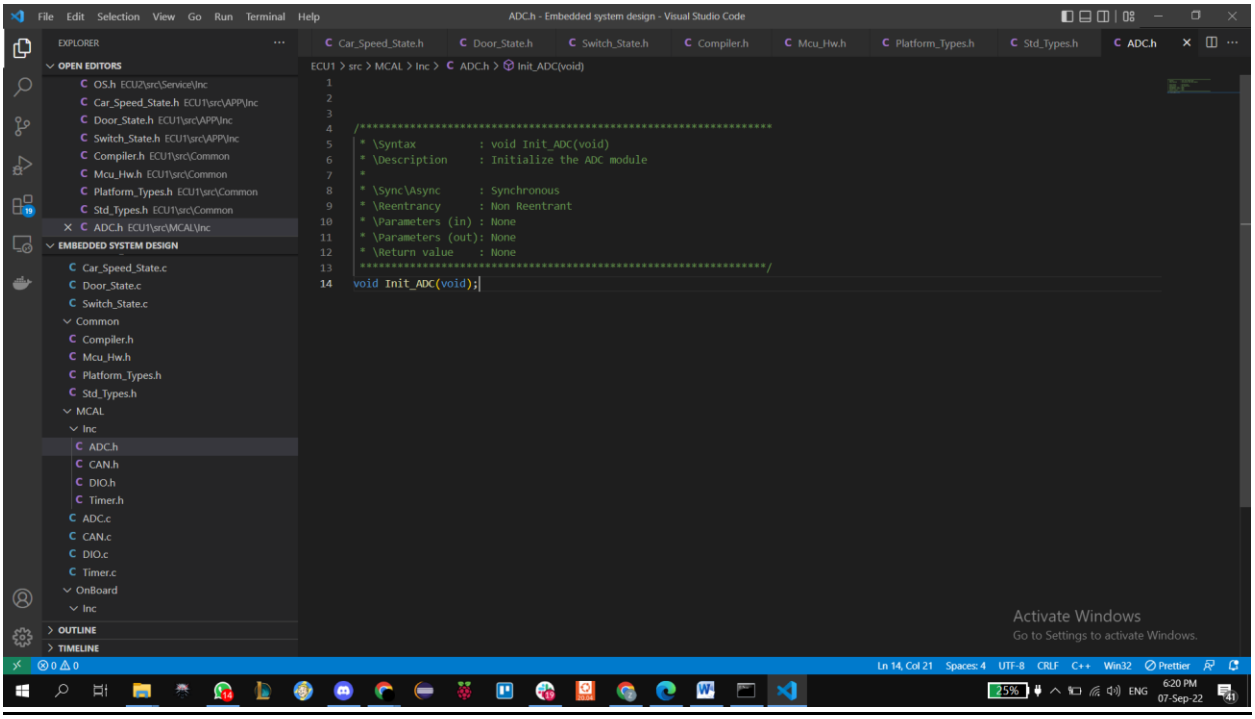
3. Switch_state_module_APIs:



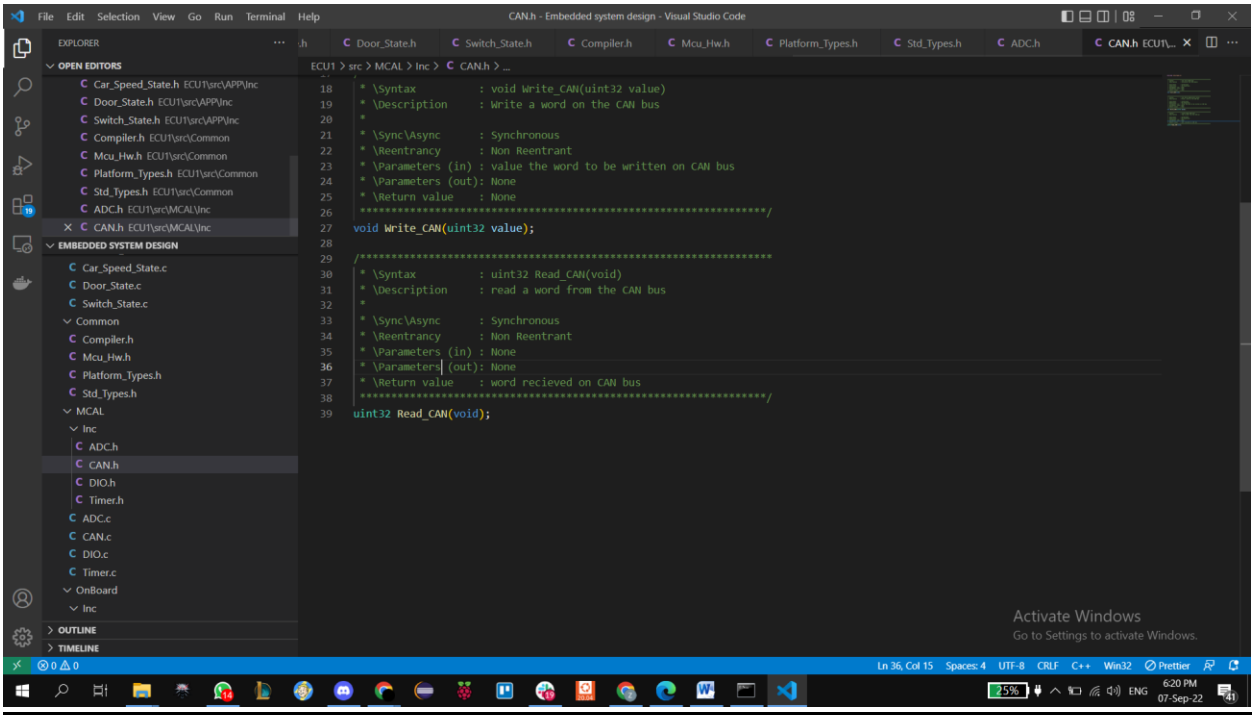
4. Std_Types_Module_APIs:



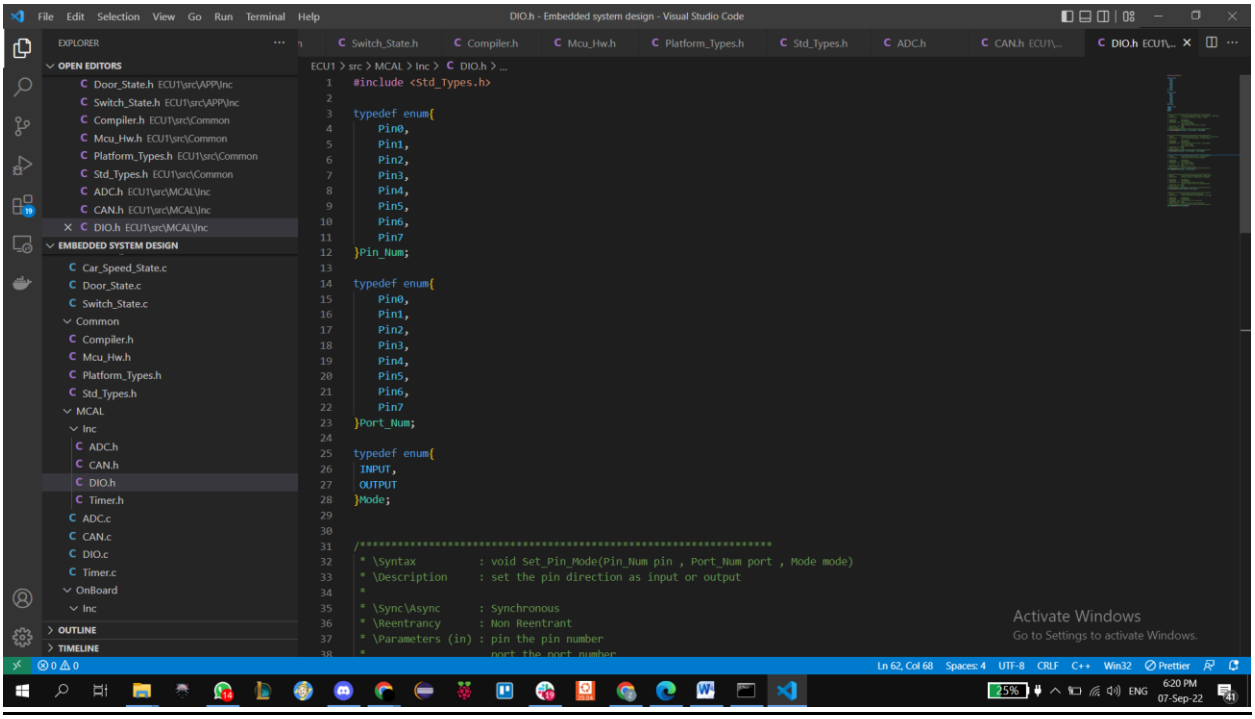
5.ADC_Module_APIs:

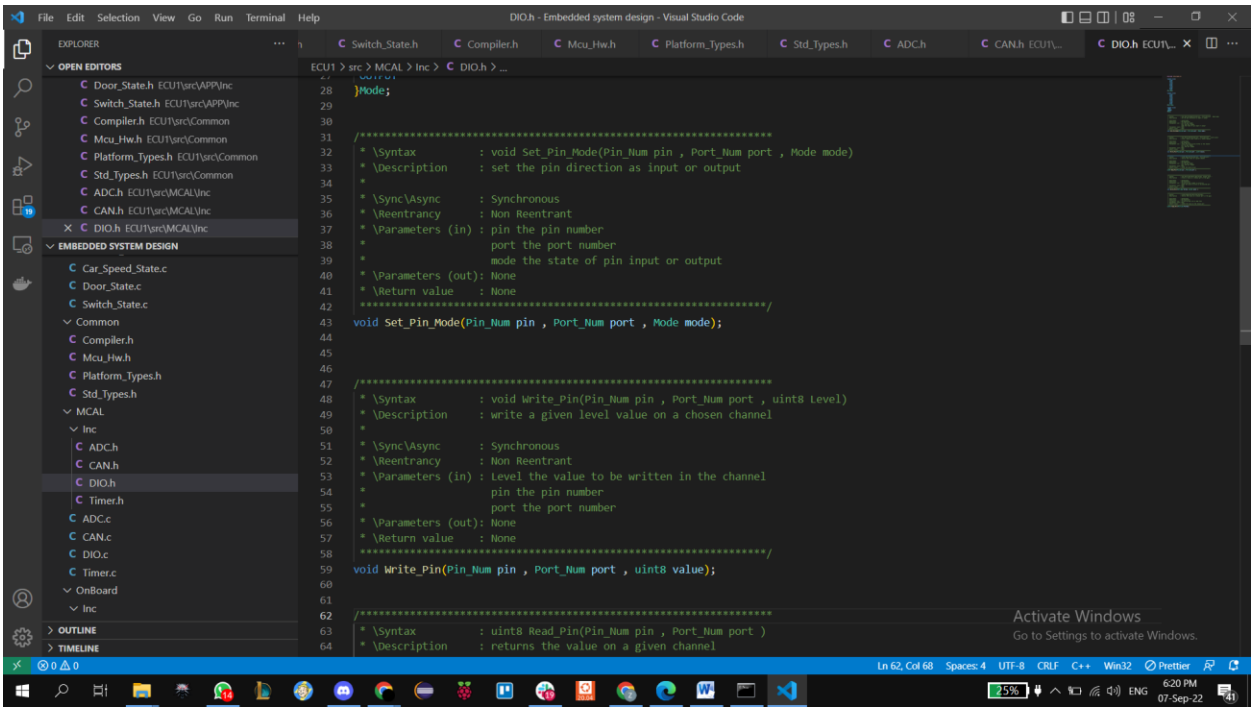


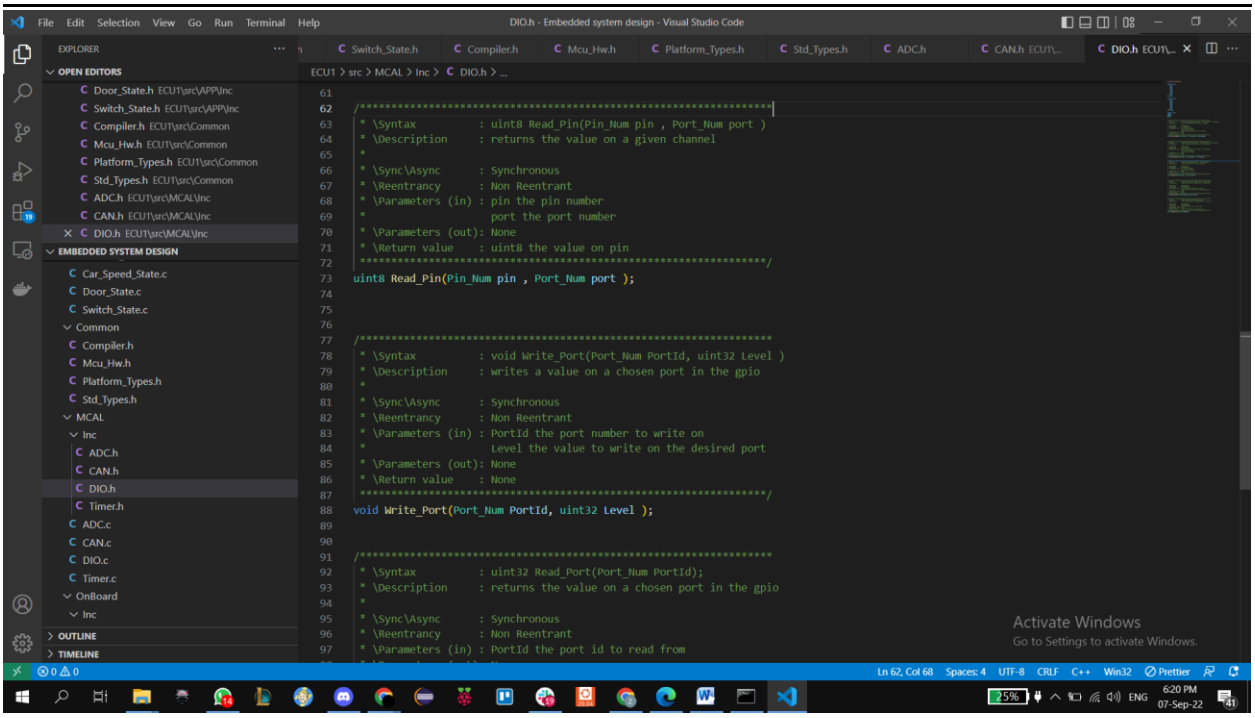
6.CAN_Module_APIs:

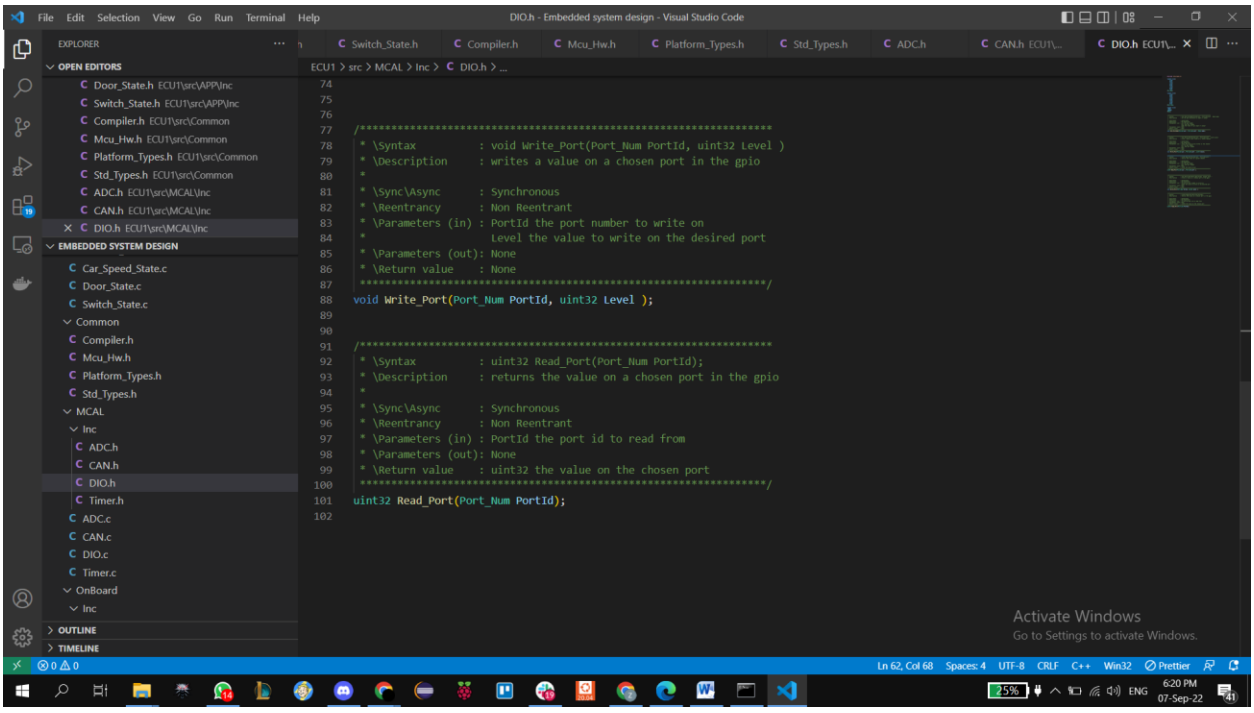


7.DIO_Module_APIs:

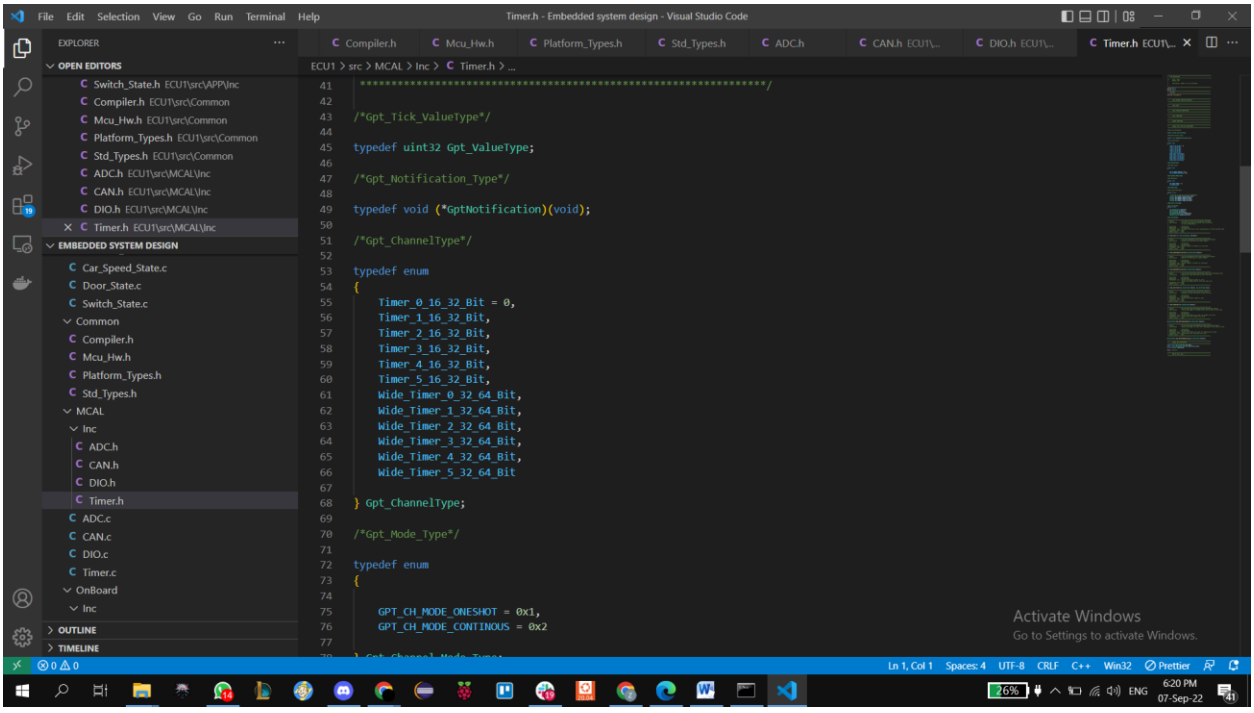


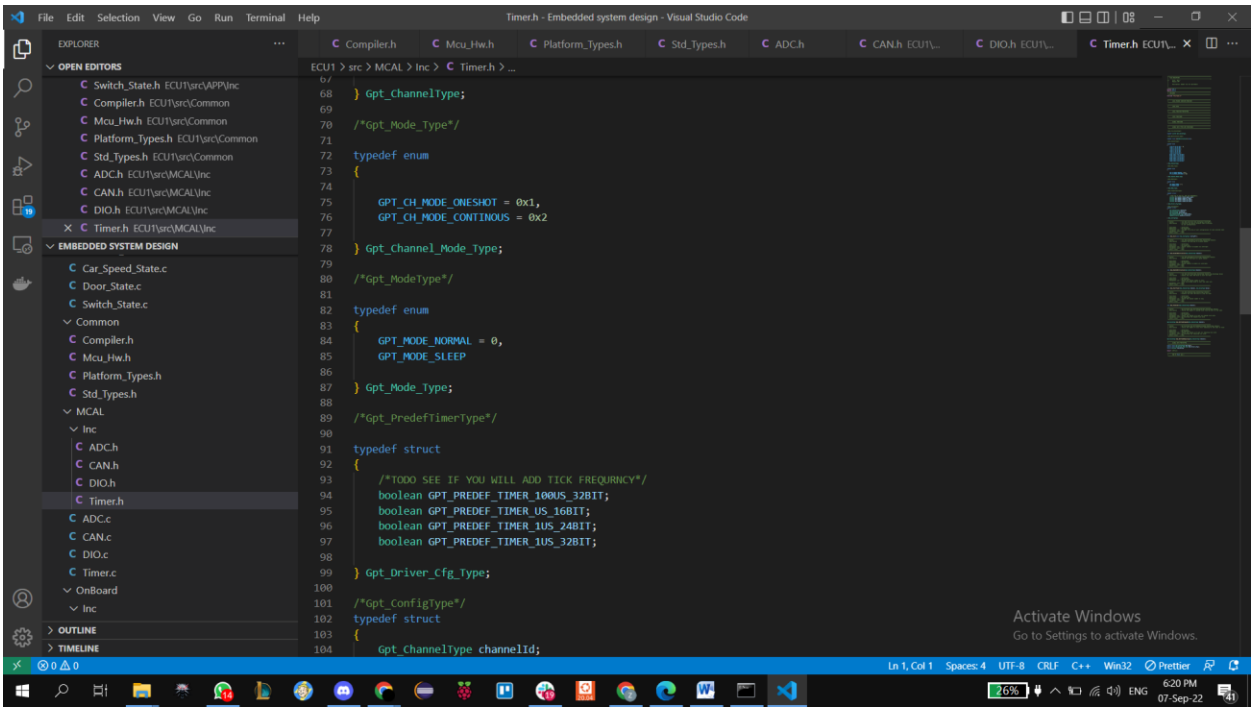

File Edit Selection View Go Run Terminal Help
DIO.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS
Door_State.h ECU1\src\APP\Inc
Switch_State.h ECU1\src\APP\Inc
Compiler.h ECU1\src\Common
Mcu_Hw.h ECU1\src\Common
Platform_Types.h ECU1\src\Common
Std_Types.h ECU1\src\Common
ADC.h ECU1\src\MCAL\Inc
CAN.h ECU1\src\MCAL\Inc
DIO.h ECU1\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Car_Speed_State.c
Door_State.c
Switch_State.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
ADC.h
CAN.h
DIO.h
Timer.h
ADC.c
CAN.c
DIO.c
Timer.c
OnBoard
Inc
OUTLINE
TIMELINE
ECU1 > src > MCAL > Inc > DIO.h > ...
}Mode;
/**********
* \Syntax : void Set_Pin_Mode(Pin_Num pin , Port_Num port , Mode mode)
* \Description : set the pin direction as input or output
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : pin the pin number
* port the port number
* mode the state of pin input or output
* \Parameters (out): None
* \Return value : None
**********/
void Set_Pin_Mode(Pin_Num pin , Port_Num port , Mode mode);
/**********
* \Syntax : void Write_Pin(Pin_Num pin , Port_Num port , uint8 Level)
* \Description : write a given level value on a chosen channel
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : Level the value to be written in the channel
* pin the pin number
* port the port number
* \Parameters (out): None
* \Return value : None
**********/
void Write_Pin(Pin_Num pin , Port_Num port , uint8 value);
/**********
* \Syntax : uint8 Read_Pin(Pin_Num pin , Port_Num port)
* \Description : returns the value on a given channel
Activate Windows
Go to Settings to activate Windows.
Ln 62, Col 68 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
25% ENG 6:20 PM 07-Sep-22

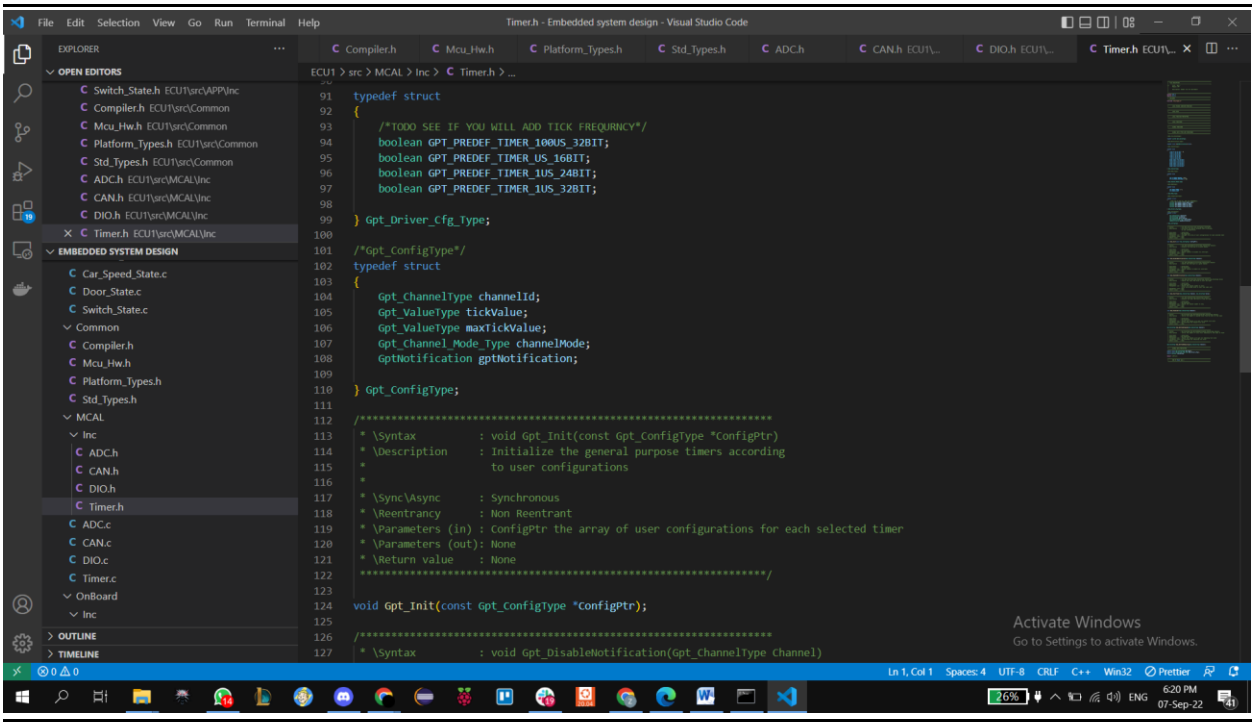

File Edit Selection View Go Run Terminal Help
DIO.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS
Door_State.h ECU1\src\APP\Inc
Switch_State.h ECU1\src\APP\Inc
Compiler.h ECU1\src\Common
Mcu_Hw.h ECU1\src\Common
Platform_Types.h ECU1\src\Common
Std_Types.h ECU1\src\Common
ADC.h ECU1\src\MCAL\Inc
CAN.h ECU1\src\MCAL\Inc
DIO.h ECU1\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Car_Speed_State.c
Door_State.c
Switch_State.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
ADC.h
CAN.h
DIO.h
Timer.h
ADC.c
CAN.c
DIO.c
Timer.c
OnBoard
Inc
OUTLINE
TIMELINE
ECU1 > src > MCAL > Inc > DIO.h > ...
/**********
* \Syntax : uint8 Read_Pin(Pin_Num pin , Port_Num port)
* \Description : returns the value on a given channel
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : pin the pin number
* port the port number
* \Parameters (out): None
* \Return value : uint8 the value on pin
**********/
uint8 Read_Pin(Pin_Num pin , Port_Num port);
/**********
* \Syntax : void Write_Port(Port_Num PortId, uint32 Level)
* \Description : writes a value on a chosen port in the gpio
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : PortId the port number to write on
* Level the value to write on the desired port
* \Parameters (out): None
* \Return value : None
**********/
void Write_Port(Port_Num PortId, uint32 Level);
/**********
* \Syntax : uint32 Read_Port(Port_Num PortId);
* \Description : returns the value on a chosen port in the gpio
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : PortId the port id to read from
Activate Windows
Go to Settings to activate Windows.
Ln 62, Col 68 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
25% ENG 6:20 PM 07-Sep-22

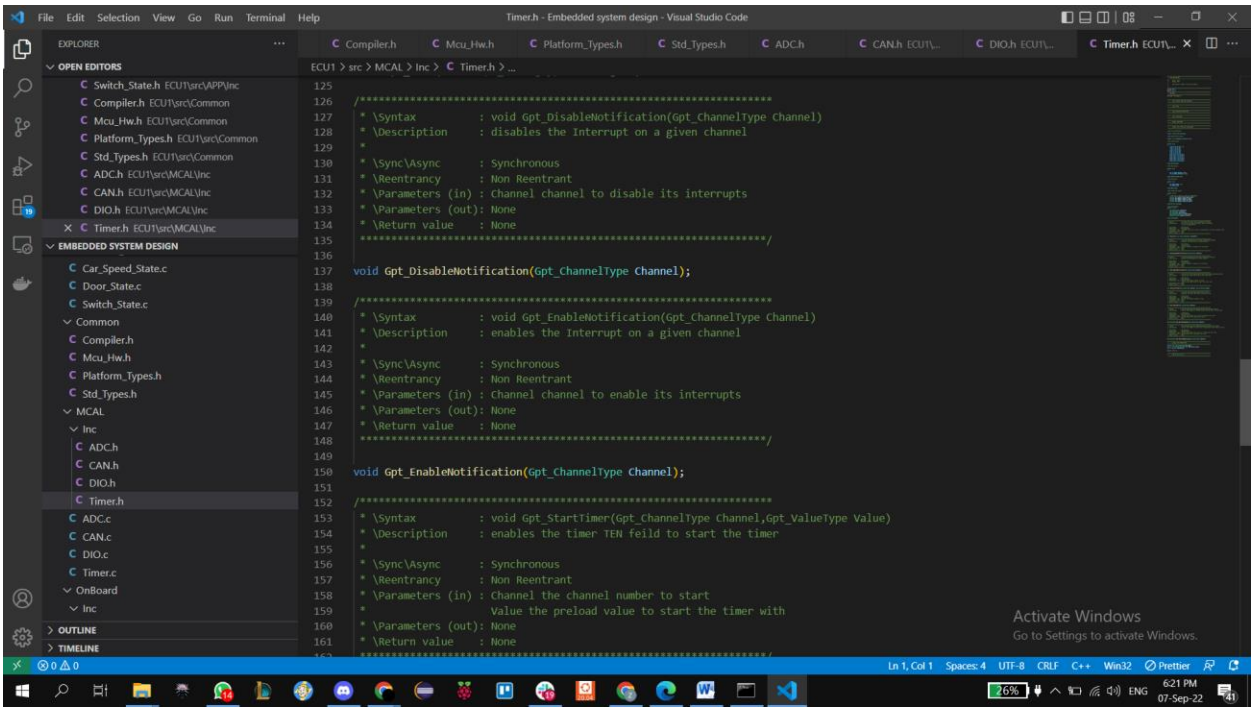


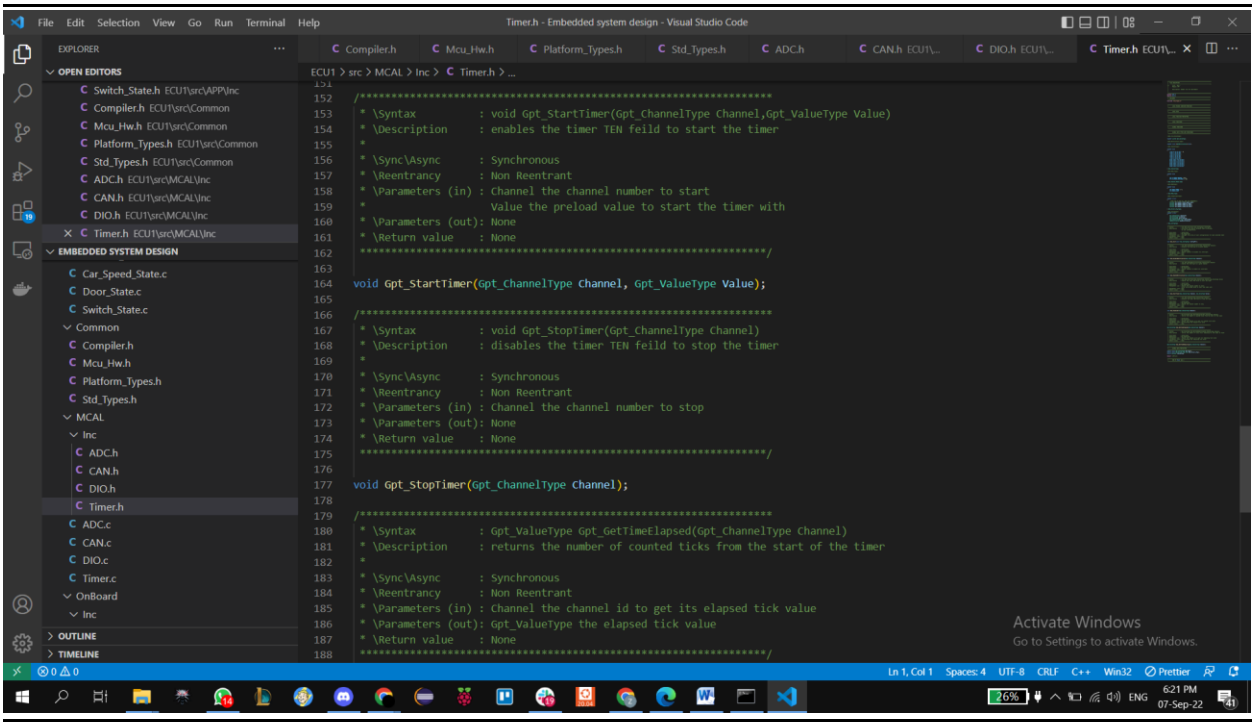
8.Timer_Module_APIs:

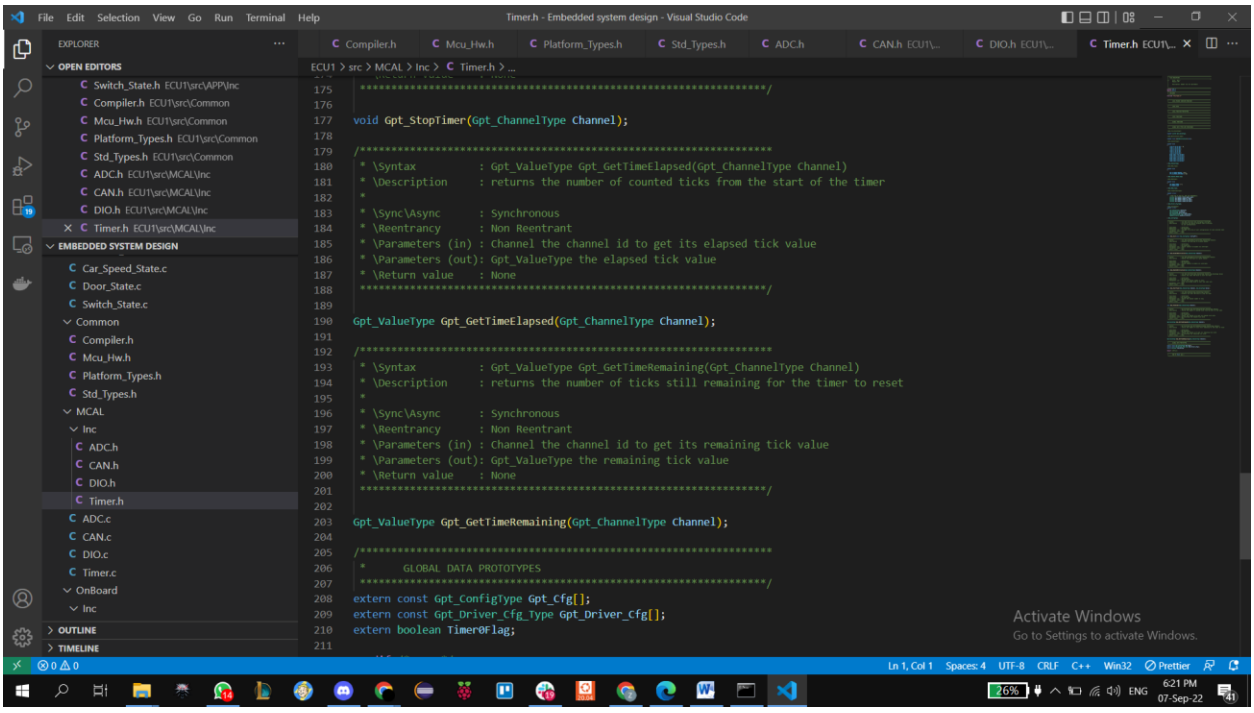


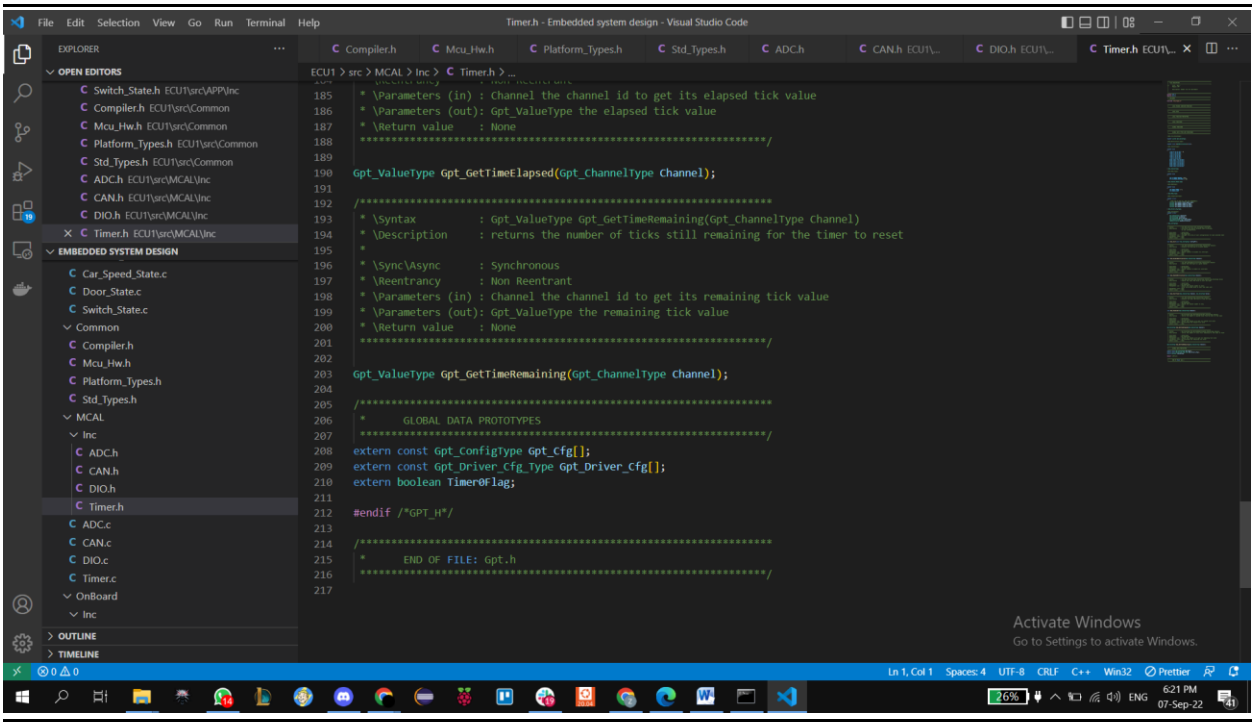

File Edit Selection View Go Run Terminal Help
Timer.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS
Switch_State.h ECU1\src\APP\Inc
Compiler.h ECU1\src\Common
Mcu_Hw.h ECU1\src\Common
Platform_Types.h ECU1\src\Common
Std_Types.h ECU1\src\Common
ADC.h ECU1\src\MCAL\Inc
CAN.h ECU1\src\MCAL\Inc
DIO.h ECU1\src\MCAL\Inc
Timer.h ECU1\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Car_Speed_State.c
Door_State.c
Switch_State.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
ADC.h
CAN.h
DIO.h
Timer.h
ADC.c
CAN.c
DIO.c
Timer.c
OnBoard
Inc
OUTLINE
TIMELINE
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
ADC.h
CAN.h ECU1\...
DIO.h ECU1\...
Timer.h ECU1\...
ECU1 > src > MCAL > Inc > Timer.h > ...
} Gpt_ChannelType;
/*Gpt_Mode_Type*/
typedef enum
{
GPT_CH_MODE_ONESHOT = 0x1,
GPT_CH_MODE_CONTINOUS = 0x2
} Gpt_Channel_Mode_Type;
/*Gpt_ModeType*/
typedef enum
{
GPT_MODE_NORMAL = 0,
GPT_MODE_SLEEP
} Gpt_Mode_Type;
/*Gpt_PredefTimerType*/
typedef struct
{
/*TODO SEE IF YOU WILL ADD TICK FREQURNCY*/
boolean GPT_PREDEF_TIMER_100US_32BIT;
boolean GPT_PREDEF_TIMER_US_16BIT;
boolean GPT_PREDEF_TIMER_1US_24BIT;
boolean GPT_PREDEF_TIMER_1US_32BIT;
} Gpt_Driver_Cfg_Type;
/*Gpt_ConfigType*/
typedef struct
{
Gpt_ChannelType channelId;
Activate Windows
Go to Settings to activate Windows.
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
26% ENG 6:20 PM 07-Sep-22


File Edit Selection View Go Run Terminal Help
Timer.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS
Switch_State.h ECU1\src\APP\Inc
Compiler.h ECU1\src\Common
Mcu_Hw.h ECU1\src\Common
Platform_Types.h ECU1\src\Common
Std_Types.h ECU1\src\Common
ADC.h ECU1\src\MCAL\Inc
CAN.h ECU1\src\MCAL\Inc
DIO.h ECU1\src\MCAL\Inc
Timer.h ECU1\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Car_Speed_State.c
Door_State.c
Switch_State.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
ADC.h
CAN.h
DIO.h
Timer.h
ADC.c
CAN.c
DIO.c
Timer.c
OnBoard
Inc
OUTLINE
TIMELINE
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
ADC.h
CAN.h ECU1\...
DIO.h ECU1\...
Timer.h ECU1\...
ECU1 > src > MCAL > Inc > Timer.h > ...
typedef struct
{
/*TODO SEE IF YOU WILL ADD TICK FREQURNCY*/
boolean GPT_PREDEF_TIMER_100US_32BIT;
boolean GPT_PREDEF_TIMER_US_16BIT;
boolean GPT_PREDEF_TIMER_1US_24BIT;
boolean GPT_PREDEF_TIMER_1US_32BIT;
} Gpt_Driver_Cfg_Type;
/*Gpt_ConfigType*/
typedef struct
{
Gpt_ChannelType channelId;
Gpt_ValueType tickValue;
Gpt_ValueType maxTickValue;
Gpt_Channel_Mode_Type channelMode;
GptNotification gptNotification;
} Gpt_ConfigType;
/**
* \Syntax : void Gpt_Init(const Gpt_ConfigType *ConfigPtr)
* \Description : Initialize the general purpose timers according
* to user configurations
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : ConfigPtr the array of user configurations for each selected timer
* \Parameters (out): None
* \Return value : None
**/
void Gpt_Init(const Gpt_ConfigType *ConfigPtr);
/**
* \Syntax : void Gpt_DisableNotification(Gpt_ChannelType Channel)
Activate Windows
Go to Settings to activate Windows.
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
26% ENG 6:20 PM 07-Sep-22

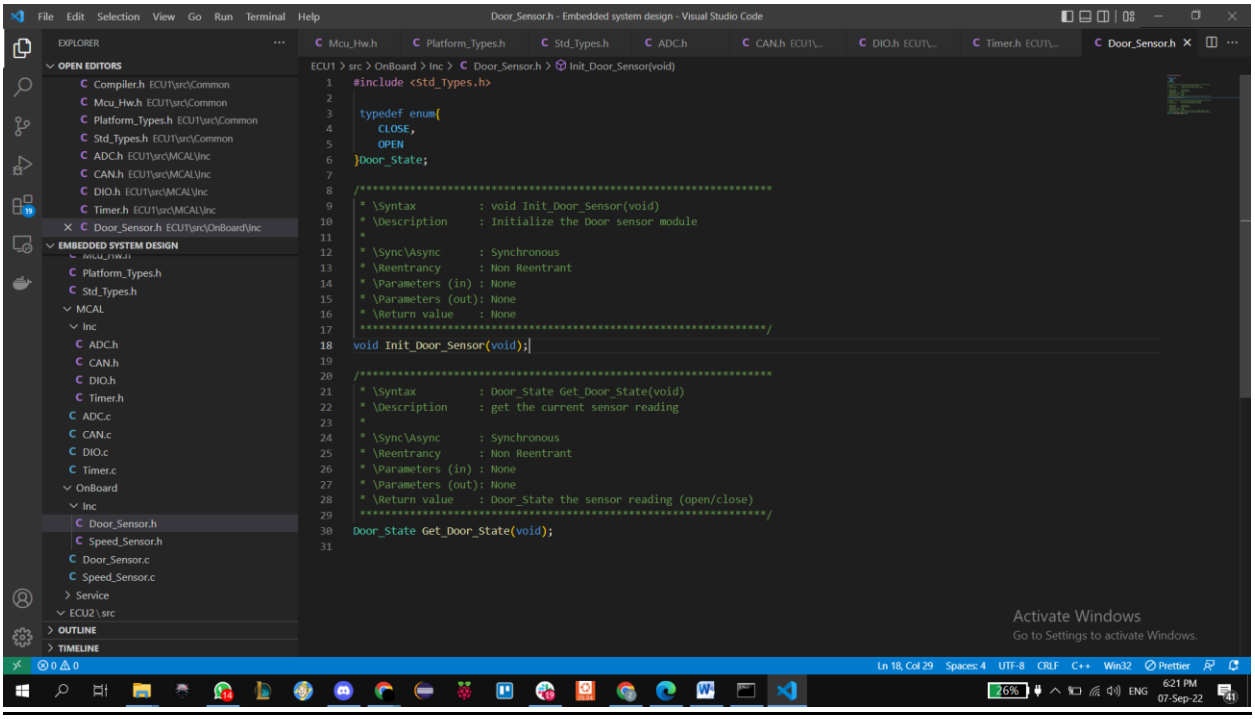

File Edit Selection View Go Run Terminal Help
Timer.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS
Switch_State.h ECU1\src\APP\Inc
Compiler.h ECU1\src\Common
Mcu_Hw.h ECU1\src\Common
Platform_Types.h ECU1\src\Common
Std_Types.h ECU1\src\Common
ADC.h ECU1\src\MCAL\Inc
CAN.h ECU1\src\MCAL\Inc
DIO.h ECU1\src\MCAL\Inc
Timer.h ECU1\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Car_Speed_State.c
Door_State.c
Switch_State.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
ADC.h
CAN.h
DIO.h
Timer.h
ADC.c
CAN.c
DIO.c
Timer.c
OnBoard
Inc
OUTLINE
TIMELINE
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
ADC.h
CAN.h ECU1\...
DIO.h ECU1\...
Timer.h ECU1\...
ECU1 > src > MCAL > Inc > Timer.h > ...
/***
* \Syntax : void Gpt_DisableNotification(Gpt_ChannelType Channel)
* \Description : disables the Interrupt on a given channel
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : Channel channel to disable its interrupts
* \Parameters (out): None
* \Return value : None
***/
void Gpt_DisableNotification(Gpt_ChannelType Channel);
/***
* \Syntax : void Gpt_EnableNotification(Gpt_ChannelType Channel)
* \Description : enables the Interrupt on a given channel
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : Channel channel to enable its interrupts
* \Parameters (out): None
* \Return value : None
***/
void Gpt_EnableNotification(Gpt_ChannelType Channel);
/***
* \Syntax : void Gpt_StartTimer(Gpt_ChannelType Channel,Gpt_ValueType Value)
* \Description : enables the timer TEN feild to start the timer
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : Channel the channel number to start
* Value the preload value to start the timer with
* \Parameters (out): None
* \Return value : None
Activate Windows
Go to Settings to activate Windows.
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
ENG 6:21 PM 07-Sep-22


File Edit Selection View Go Run Terminal Help
Timer.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS
Switch_State.h ECU1\src\APP\Inc
Compiler.h ECU1\src\Common
Mcu_Hw.h ECU1\src\Common
Platform_Types.h ECU1\src\Common
Std_Types.h ECU1\src\Common
ADC.h ECU1\src\MCAL\Inc
CAN.h ECU1\src\MCAL\Inc
DIO.h ECU1\src\MCAL\Inc
Timer.h ECU1\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Car_Speed_State.c
Door_State.c
Switch_State.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
ADC.h
CAN.h
DIO.h
Timer.h
ADC.c
CAN.c
DIO.c
Timer.c
OnBoard
Inc
OUTLINE
TIMELINE
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
ADC.h
CAN.h ECU1\...
DIO.h ECU1\...
Timer.h ECU1\...
ECU1 > src > MCAL > Inc > Timer.h > ...
/***
* \Syntax : void Gpt_StartTimer(Gpt_ChannelType Channel,Gpt_ValueType Value)
* \Description : enables the timer TEN feild to start the timer
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : Channel the channel number to start
* Value the preload value to start the timer with
* \Parameters (out): None
* \Return value : None
***/
void Gpt_StartTimer(Gpt_ChannelType Channel, Gpt_ValueType Value);
/***
* \Syntax : void Gpt_StopTimer(Gpt_ChannelType Channel)
* \Description : disables the timer TEN feild to stop the timer
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : Channel the channel number to stop
* \Parameters (out): None
* \Return value : None
***/
void Gpt_StopTimer(Gpt_ChannelType Channel);
/***
* \Syntax : Gpt_ValueType Gpt_GetTimeElapsed(Gpt_ChannelType Channel)
* \Description : returns the number of counted ticks from the start of the timer
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : Channel the channel id to get its elapsed tick value
* \Parameters (out): Gpt_ValueType the elapsed tick value
* \Return value : None
Activate Windows
Go to Settings to activate Windows.
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
ENG 6:21 PM 07-Sep-22

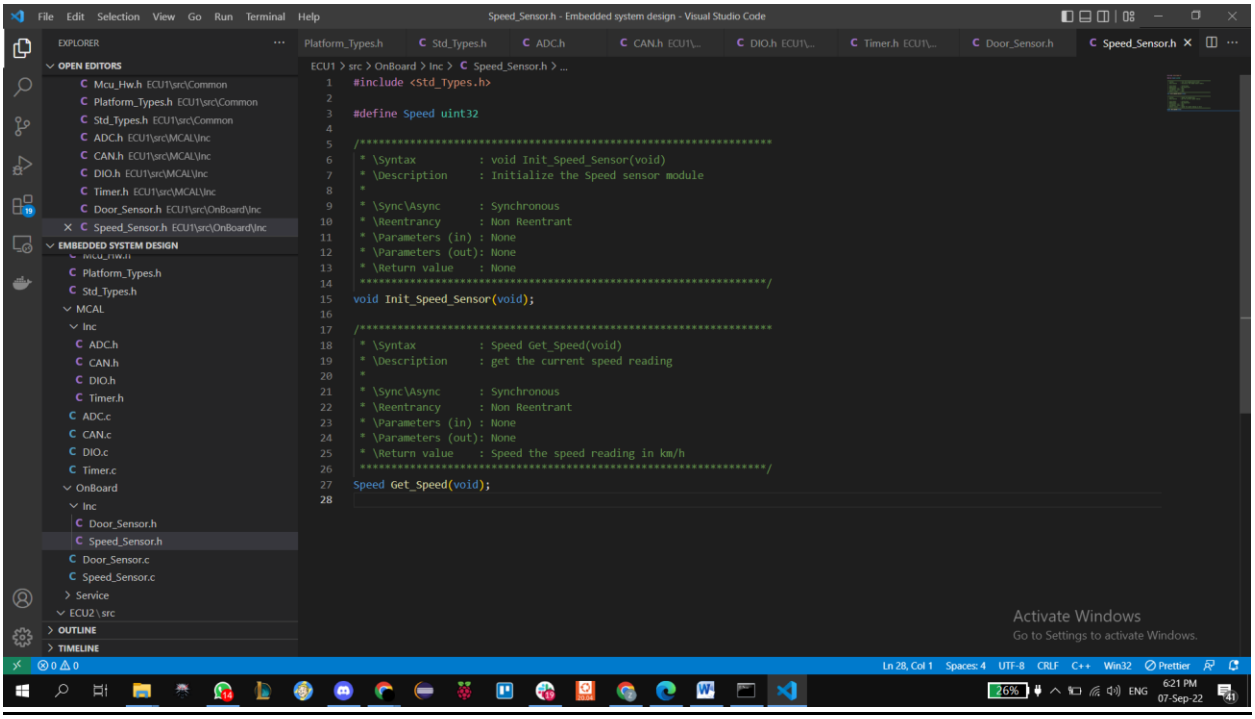

File Edit Selection View Go Run Terminal Help
Timer.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS
Switch_State.h ECU1\src\APP\Inc
Compiler.h ECU1\src\Common
Mcu_Hw.h ECU1\src\Common
Platform_Types.h ECU1\src\Common
Std_Types.h ECU1\src\Common
ADC.h ECU1\src\MCAL\Inc
CAN.h ECU1\src\MCAL\Inc
DIO.h ECU1\src\MCAL\Inc
Timer.h ECU1\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Car_Speed_State.c
Door_State.c
Switch_State.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
ADC.h
CAN.h
DIO.h
Timer.h
ADC.c
CAN.c
DIO.c
Timer.c
OnBoard
Inc
OUTLINE
TIMELINE
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
ADC.h
CAN.h ECU1\...
DIO.h ECU1\...
Timer.h ECU1\...
ECU1 > src > MCAL > Inc > Timer.h > ...
175 **/
176
177 void Gpt_StopTimer(Gpt_ChannelType Channel);
178
179 /**
180 * \Syntax : Gpt_ValueType Gpt_GetTimeElapsed(Gpt_ChannelType Channel)
181 * \Description : returns the number of counted ticks from the start of the timer
182 *
183 * \Sync\Async : Synchronous
184 * \Reentrancy : Non Reentrant
185 * \Parameters (in) : Channel the channel id to get its elapsed tick value
186 * \Parameters (out): Gpt_ValueType the elapsed tick value
187 * \Return value : None
188 **/
189
190 Gpt_ValueType Gpt_GetTimeElapsed(Gpt_ChannelType Channel);
191
192 /**
193 * \Syntax : Gpt_ValueType Gpt_GetTimeRemaining(Gpt_ChannelType Channel)
194 * \Description : returns the number of ticks still remaining for the timer to reset
195 *
196 * \Sync\Async : Synchronous
197 * \Reentrancy : Non Reentrant
198 * \Parameters (in) : Channel the channel id to get its remaining tick value
199 * \Parameters (out): Gpt_ValueType the remaining tick value
200 * \Return value : None
201 **/
202
203 Gpt_ValueType Gpt_GetTimeRemaining(Gpt_ChannelType Channel);
204
205 /**
206 * GLOBAL DATA PROTOTYPES
207 **/
208 extern const Gpt_ConfigType Gpt_Cfg[];
209 extern const Gpt_Driver_Cfg_Type Gpt_Driver_Cfg[];
210 extern boolean Timer0Flag;
211
Activate Windows
Go to Settings to activate Windows.
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
26% ENG 6:21 PM 07-Sep-22


File Edit Selection View Go Run Terminal Help
Timer.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS
Switch_State.h ECU1\src\APP\Inc
Compiler.h ECU1\src\Common
Mcu_Hw.h ECU1\src\Common
Platform_Types.h ECU1\src\Common
Std_Types.h ECU1\src\Common
ADC.h ECU1\src\MCAL\Inc
CAN.h ECU1\src\MCAL\Inc
DIO.h ECU1\src\MCAL\Inc
Timer.h ECU1\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Car_Speed_State.c
Door_State.c
Switch_State.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
ADC.h
CAN.h
DIO.h
Timer.h
ADC.c
CAN.c
DIO.c
Timer.c
OnBoard
Inc
OUTLINE
TIMELINE
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
ADC.h
CAN.h ECU1\...
DIO.h ECU1\...
Timer.h ECU1\...
ECU1 > src > MCAL > Inc > Timer.h > ...
185 * \Parameters (in) : Channel the channel id to get its elapsed tick value
186 * \Parameters (out): Gpt_ValueType the elapsed tick value
187 * \Return value : None
188 **/
189
190 Gpt_ValueType Gpt_GetTimeElapsed(Gpt_ChannelType Channel);
191
192 /**
193 * \Syntax : Gpt_ValueType Gpt_GetTimeRemaining(Gpt_ChannelType Channel)
194 * \Description : returns the number of ticks still remaining for the timer to reset
195 *
196 * \Sync\Async : Synchronous
197 * \Reentrancy : Non Reentrant
198 * \Parameters (in) : Channel the channel id to get its remaining tick value
199 * \Parameters (out): Gpt_ValueType the remaining tick value
200 * \Return value : None
201 **/
202
203 Gpt_ValueType Gpt_GetTimeRemaining(Gpt_ChannelType Channel);
204
205 /**
206 * GLOBAL DATA PROTOTYPES
207 **/
208 extern const Gpt_ConfigType Gpt_Cfg[];
209 extern const Gpt_Driver_Cfg_Type Gpt_Driver_Cfg[];
210 extern boolean Timer0Flag;
211
212 #endif /*GPT_H*/
213
214 /**
215 * END OF FILE: Gpt.h
216 **/
217
Activate Windows
Go to Settings to activate Windows.
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
26% ENG 6:21 PM 07-Sep-22

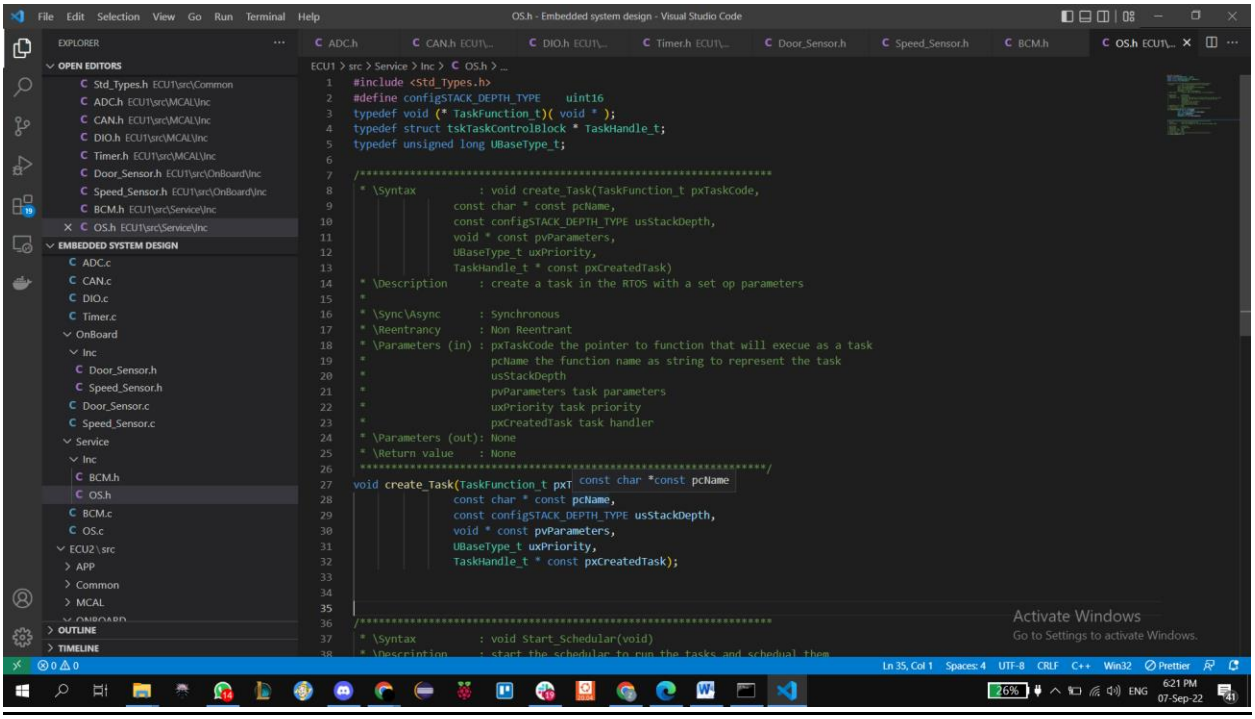
9.Door_Sensor_Module_APIs:



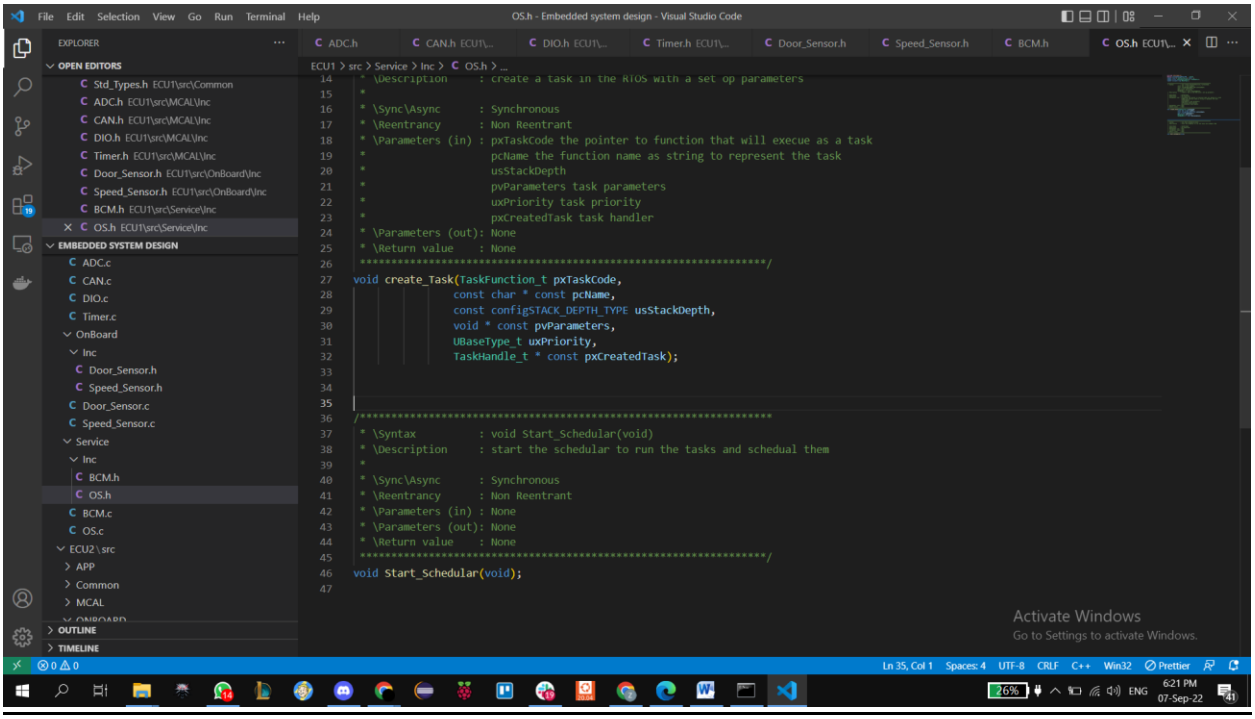
10.Speed_Sensor_Module_APIs:



11.BCM_Module_APIs:

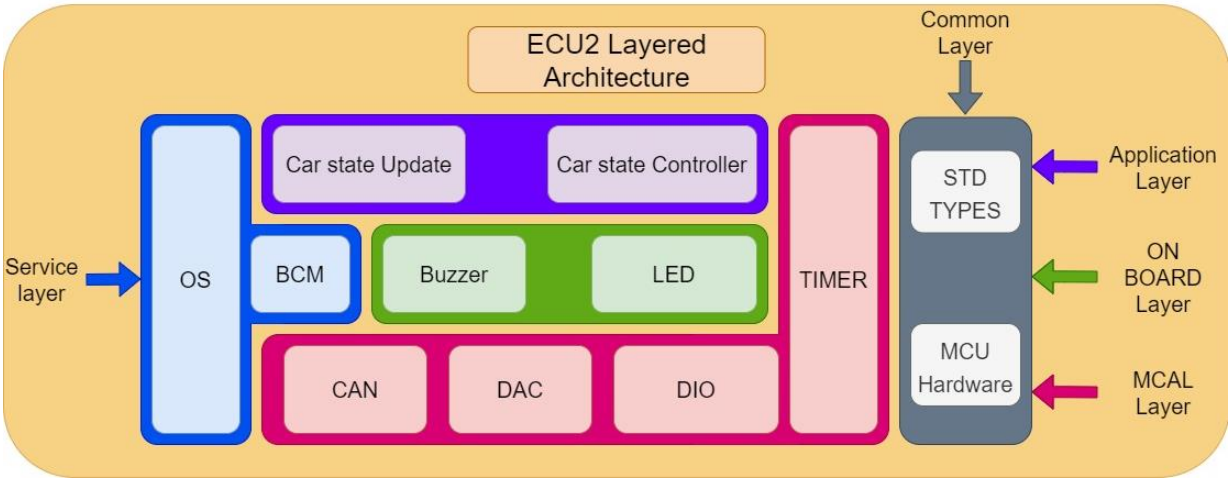


12.OS_APIs:



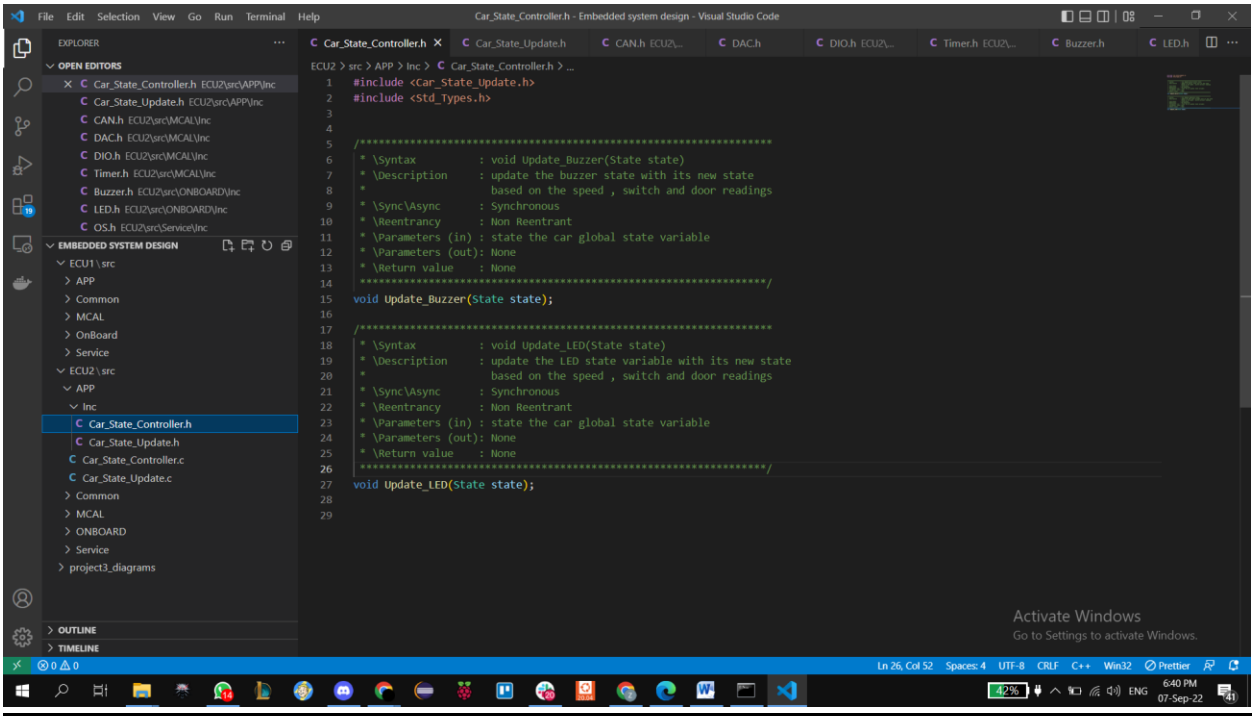
ECU2:

layered architecture:

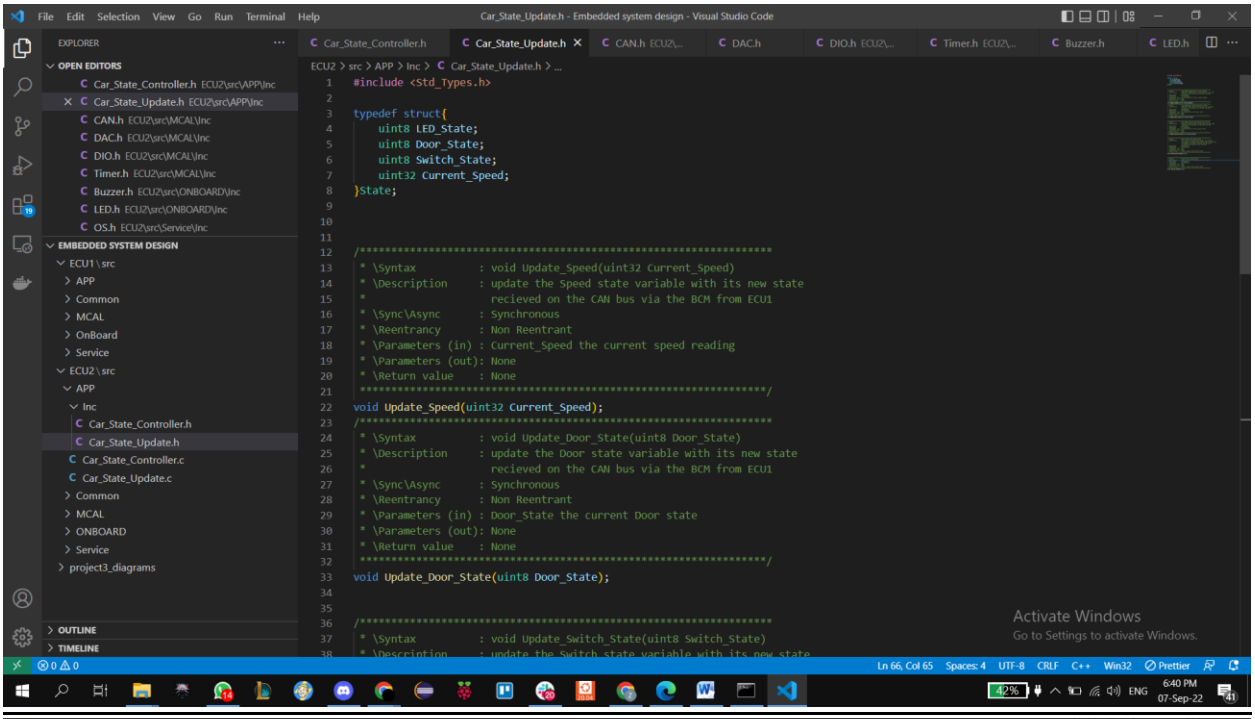


Full API details and File structure:

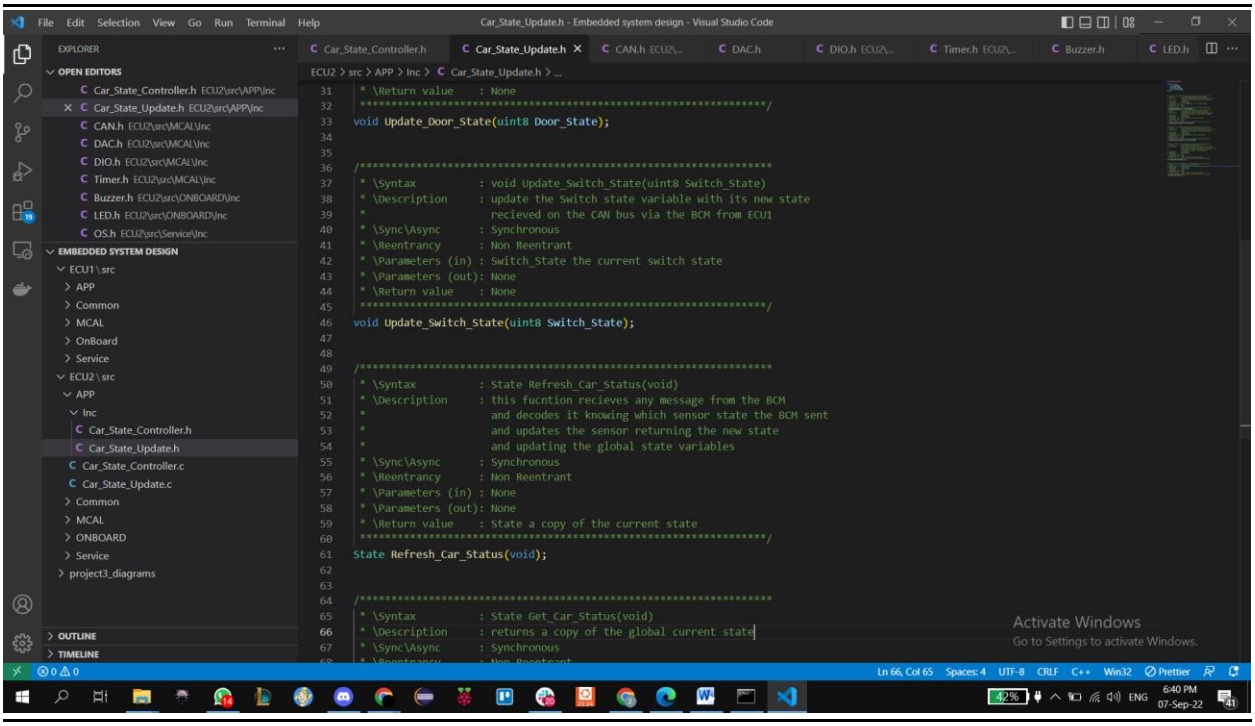
1.Car_Speed_Controller_Module_APIs:



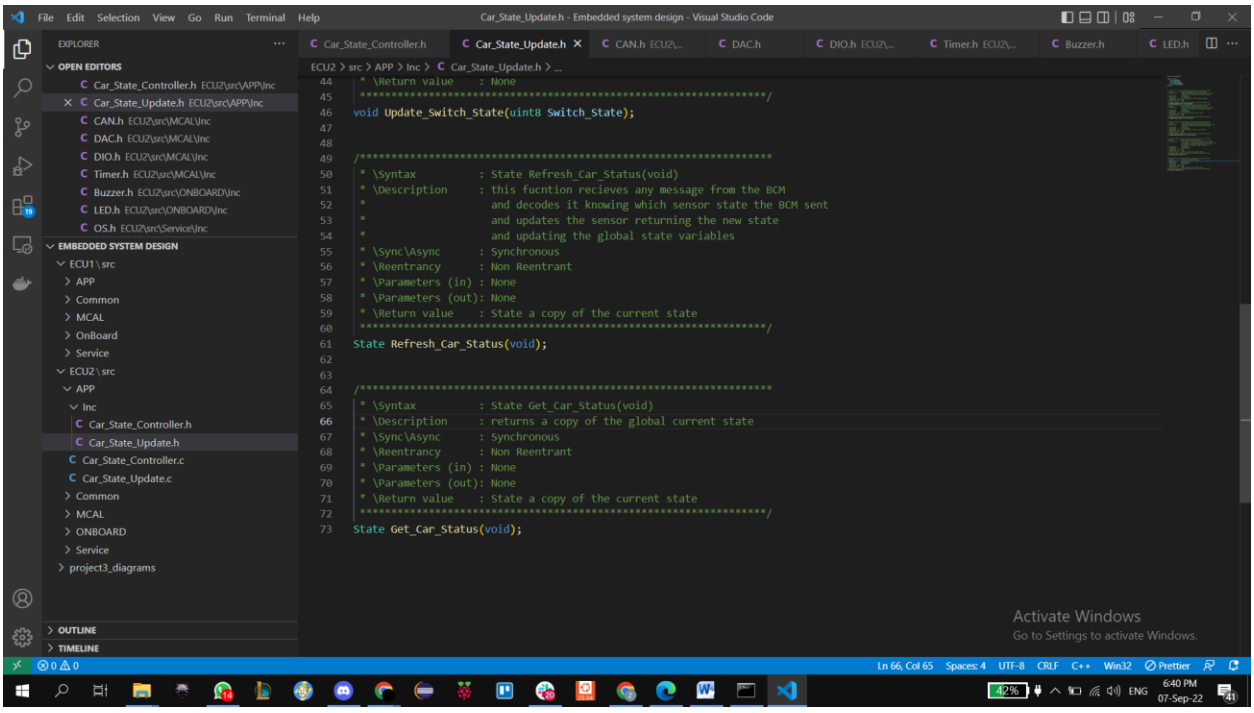
2. Car_Status_Update_Module_APIs:



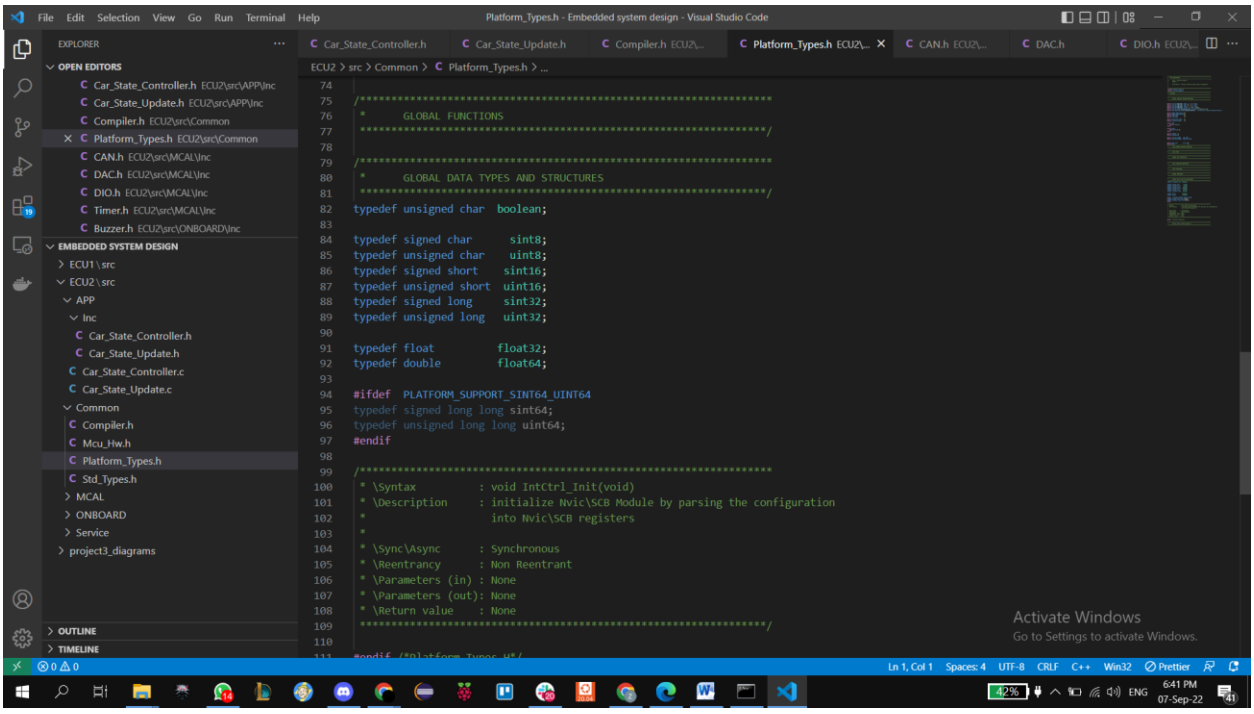
```
#include <Std_Types.h>

typedef struct{
    uint8 LED_State;
    uint8 Door_State;
    uint8 Switch_State;
    uint32 Current_Speed;
}State;



/*******************************************************************
 * \Syntax          : void Update_Speed(uint32 Current_Speed)
 * \Description     : update the Speed state variable with its new state
 *                    recieved on the CAN bus via the BCM from ECU1
 * \Sync\Async      : Synchronous
 * \Reentrancy      : Non Reentrant
 * \Parameters (in) : Current_Speed the current speed reading
 * \Parameters (out): None
 * \Return value    : None
 *******************************************************************/
void Update_Speed(uint32 Current_Speed);
/*******************************************************************
 * \Syntax          : void Update_Door_State(uint8 Door_State)
 * \Description     : update the Door state variable with its new state
 *                    recieved on the CAN bus via the BCM from ECU1
 * \Sync\Async      : Synchronous
 * \Reentrancy      : Non Reentrant
 * \Parameters (in) : Door_State the current Door state
 * \Parameters (out): None
 * \Return value    : None
 *******************************************************************/
void Update_Door_State(uint8 Door_State);


/*******************************************************************
 * \Syntax          : void Update_Switch_State(uint8 Switch_State)
 * \Description     : update the Switch state variable with its new state
```



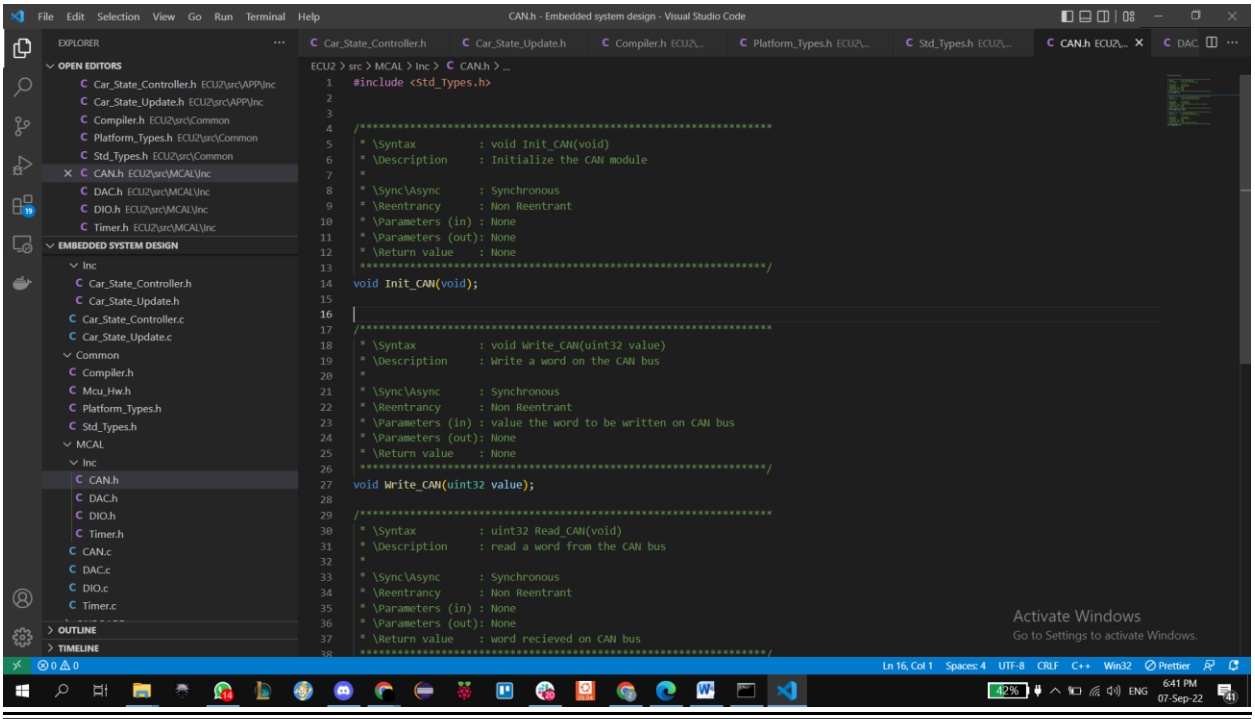
```
 * \Return value    : None
 *******************************************************************/
void Update_Door_State(uint8 Door_State);


/*******************************************************************
 * \Syntax          : void Update_Switch_State(uint8 Switch_State)
 * \Description     : update the Switch state variable with its new state
 *                    recieved on the CAN bus via the BCM from ECU1
 * \Sync\Async      : Synchronous
 * \Reentrancy      : Non Reentrant
 * \Parameters (in) : Switch_State the current switch state
 * \Parameters (out): None
 * \Return value    : None
 *******************************************************************/
void Update_Switch_State(uint8 Switch_State);


/*******************************************************************
 * \Syntax          : State Refresh_Car_Status(void)
 * \Description     : this fucntion recieves any message from the BCM
 *                    and decodes it knowing which sensor state the BCM sent
 *                    and updates the sensor returning the new state
 *                    and updating the global state variables
 * \Sync\Async      : Synchronous
 * \Reentrancy      : Non Reentrant
 * \Parameters (in) : None
 * \Parameters (out): None
 * \Return value    : State a copy of the current state
 *******************************************************************/
State Refresh_Car_Status(void);


/*******************************************************************
 * \Syntax          : State Get_Car_Status(void)
 * \Description     : returns a copy of the global current state
 * \Sync\Async      : Synchronous
```

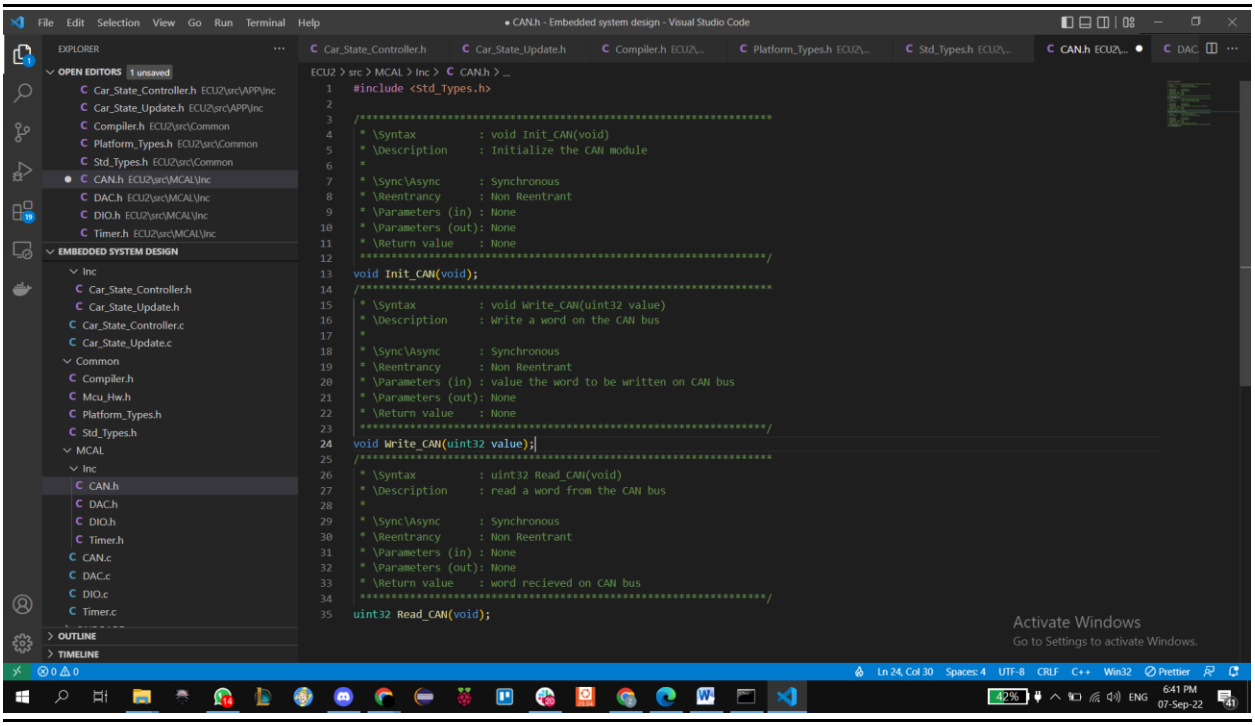


3. Std_Stypes_Module_APIs:

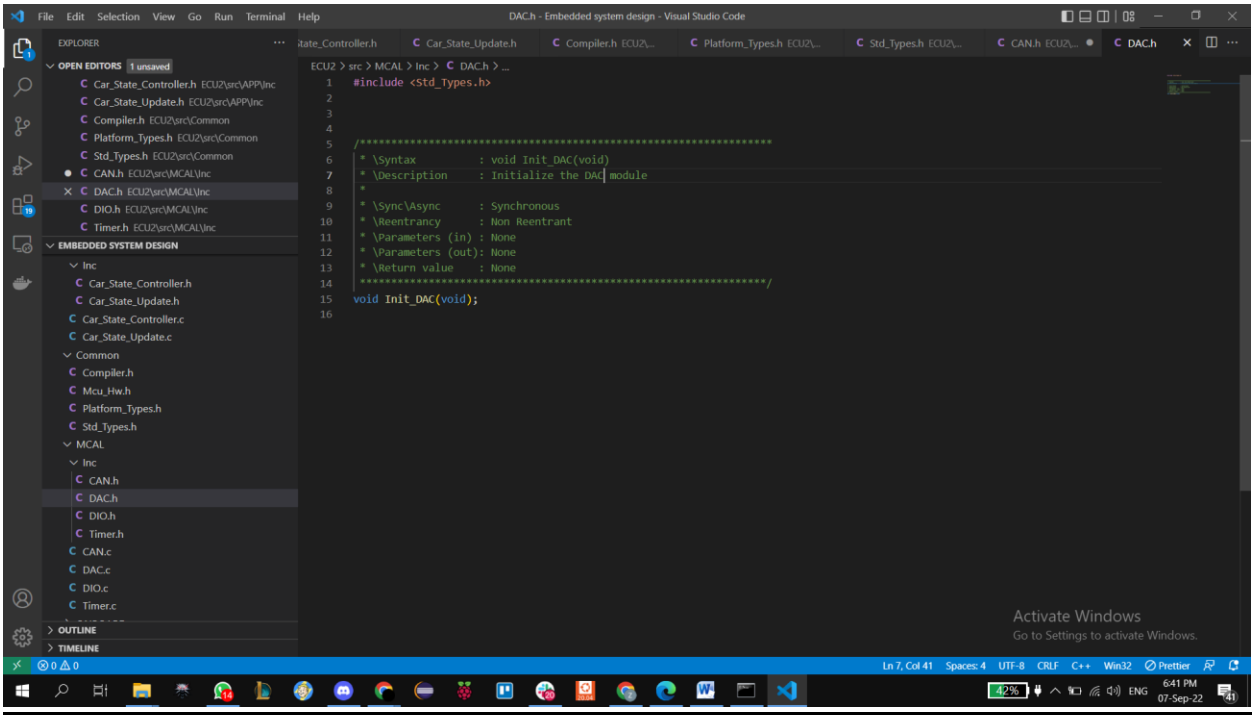


4. CAN_Module_APIs:

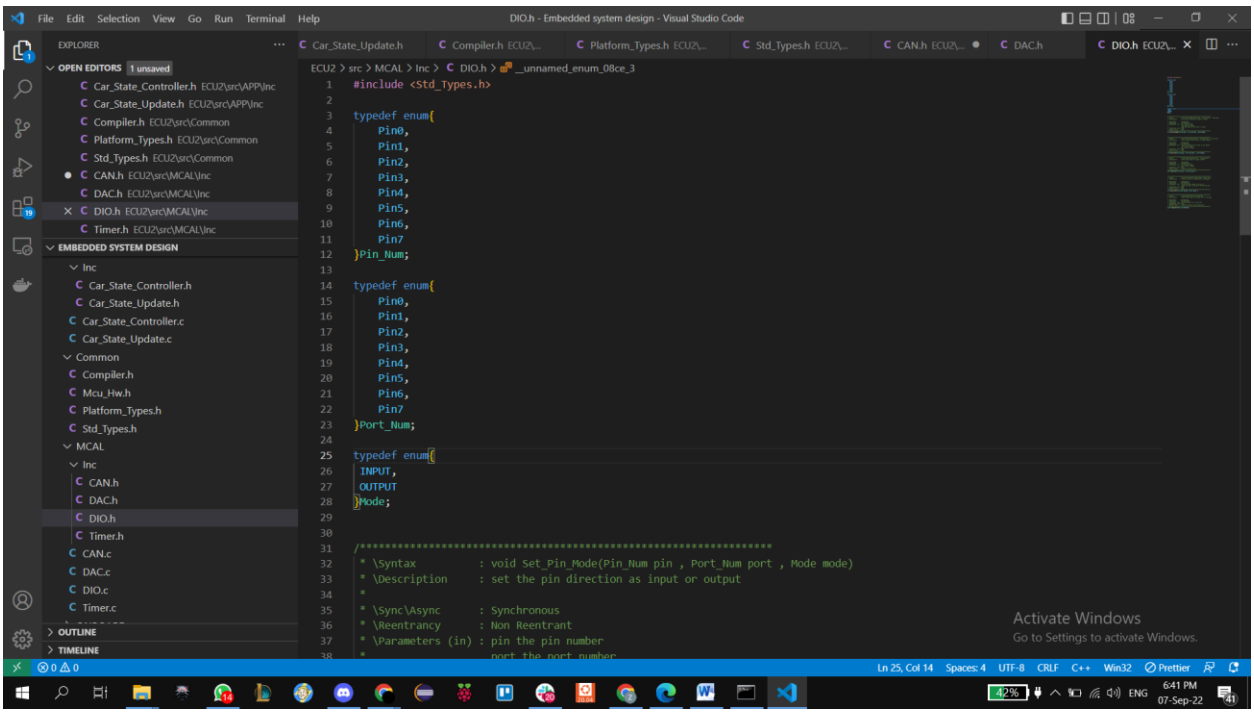


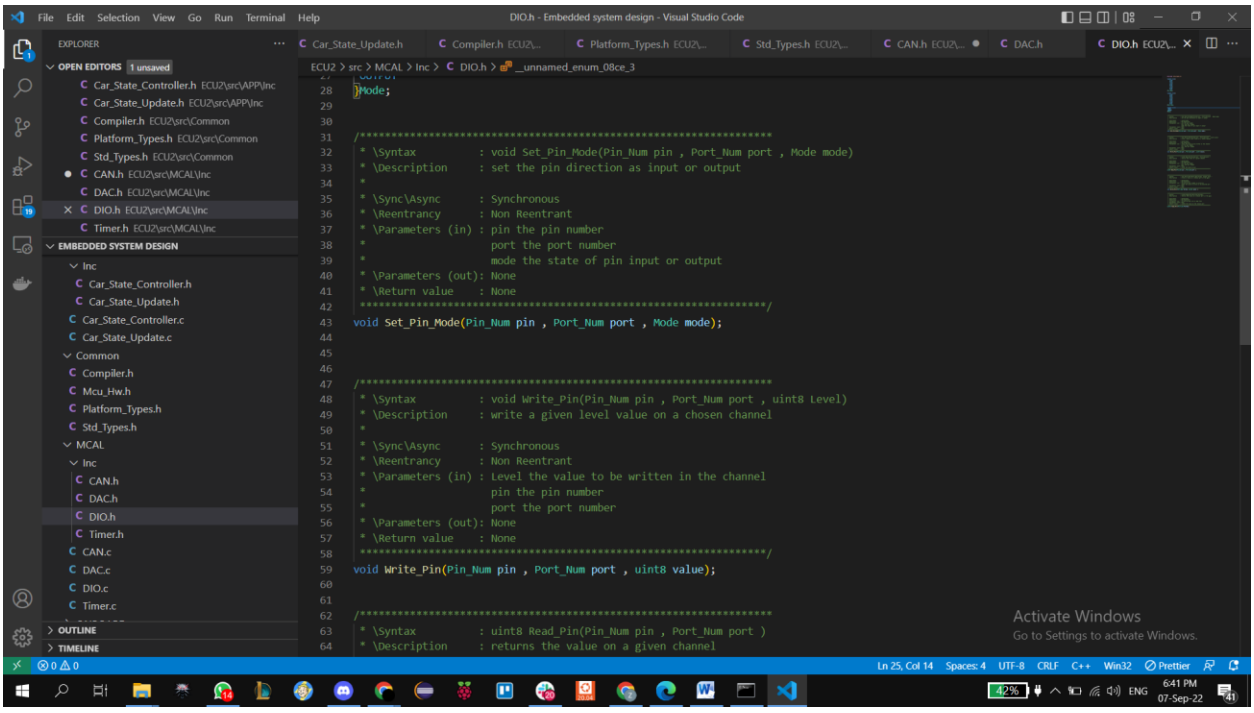


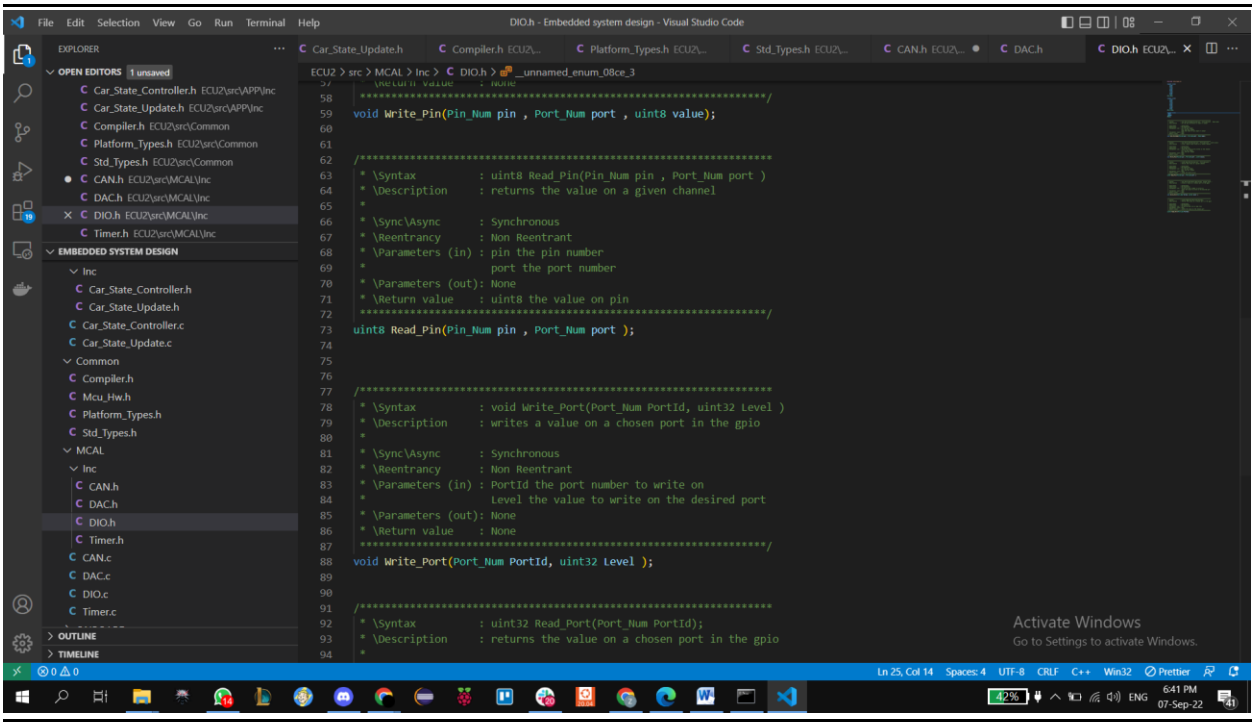
5. DAC_Module_APIs:

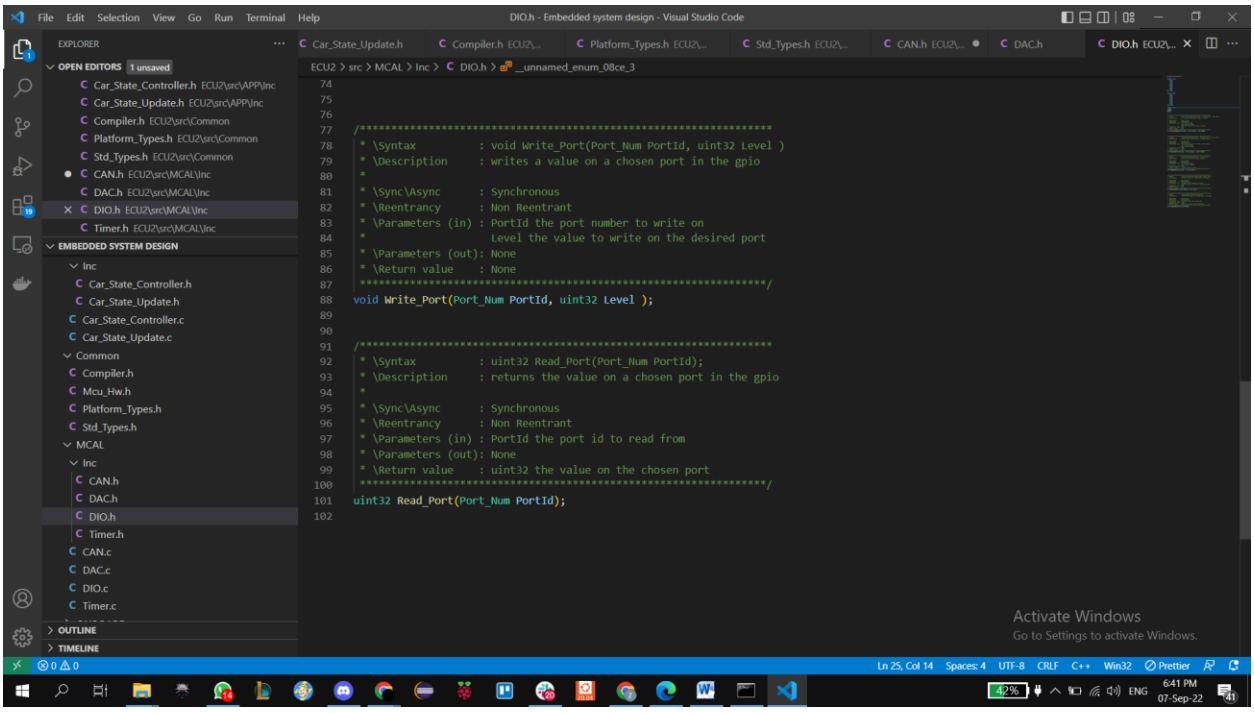


6. DIO_Module_APIs:

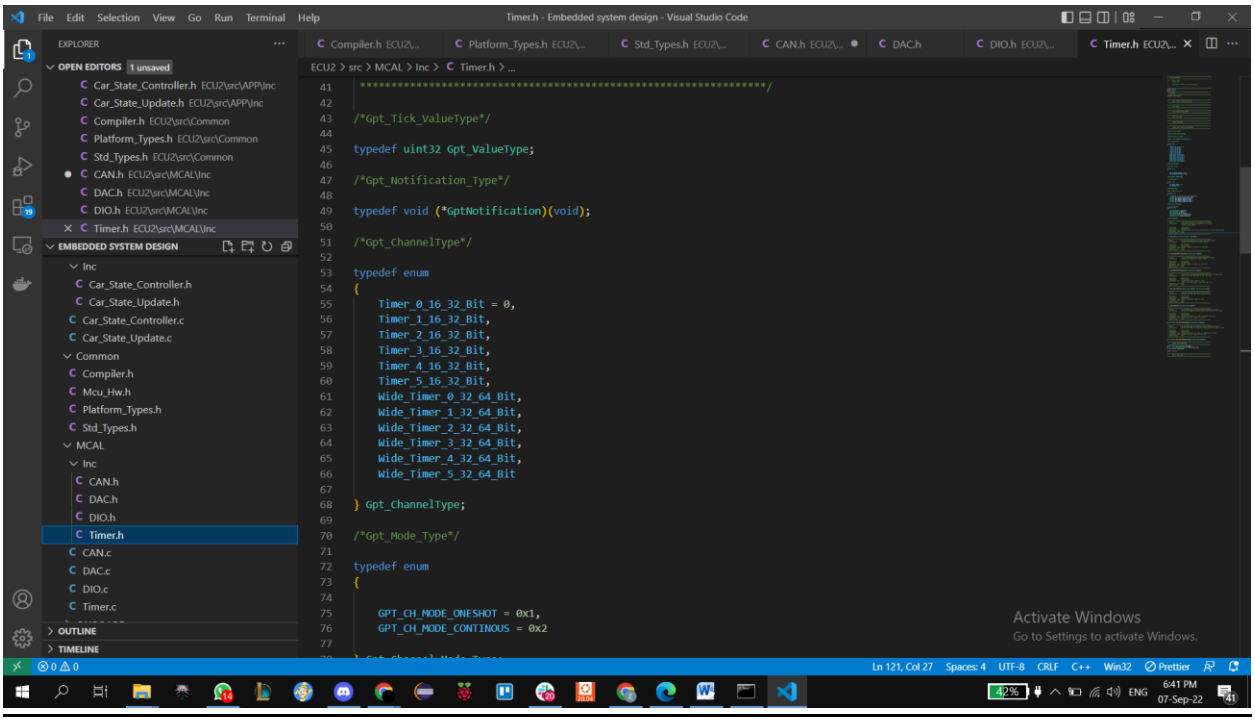


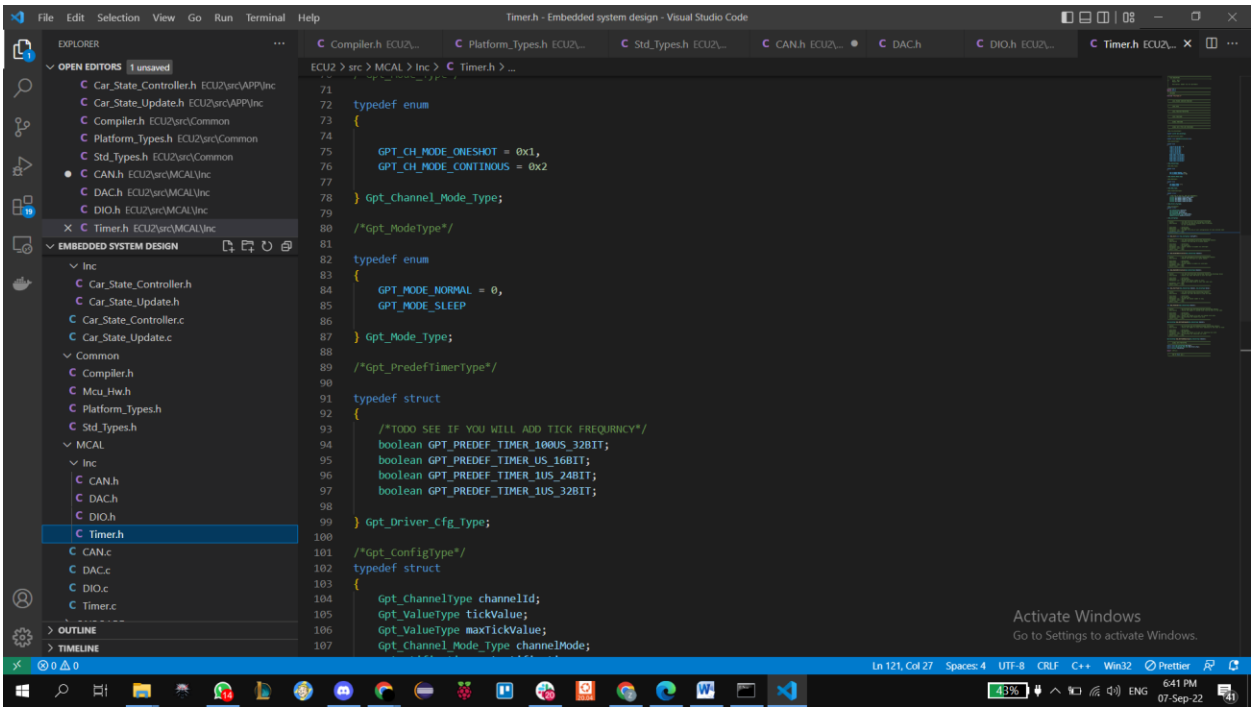

File Edit Selection View Go Run Terminal Help
DIO.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS 1 unsaved
Car_State_Controller.h ECU2\src\APP\Inc
Car_State_Update.h ECU2\src\APP\Inc
Compiler.h ECU2\src\Common
Platform_Types.h ECU2\src\Common
Std_Types.h ECU2\src\Common
CAN.h ECU2\src\MCAL\Inc
DAC.h ECU2\src\MCAL\Inc
DIO.h ECU2\src\MCAL\Inc
Timer.h ECU2\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Inc
Car_State_Controller.h
Car_State_Update.h
Car_State_Controller.c
Car_State_Update.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
CAN.h
DAC.h
DIO.h
Timer.h
CAN.c
DAC.c
DIO.c
Timer.c
OUTLINE
TIMELINE
ECU2 > src > MCAL > Inc > DIO.h > __unnamed_enum_08ce_3
}Mode;
/***
* \Syntax : void Set_Pin_Mode(Pin_Num pin , Port_Num port , Mode mode)
* \Description : set the pin direction as input or output
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : pin the pin number
* port the port number
* mode the state of pin input or output
* \Parameters (out): None
* \Return value : None
***/
void Set_Pin_Mode(Pin_Num pin , Port_Num port , Mode mode);
/***
* \Syntax : void Write_Pin(Pin_Num pin , Port_Num port , uint8 Level)
* \Description : write a given level value on a chosen channel
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : Level the value to be written in the channel
* pin the pin number
* port the port number
* \Parameters (out): None
* \Return value : None
***/
void Write_Pin(Pin_Num pin , Port_Num port , uint8 value);
/***
* \Syntax : uint8 Read_Pin(Pin_Num pin , Port_Num port)
* \Description : returns the value on a given channel
Activate Windows
Go to Settings to activate Windows.
Ln 25, Col 14 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
ENG 6:41 PM 07-Sep-22

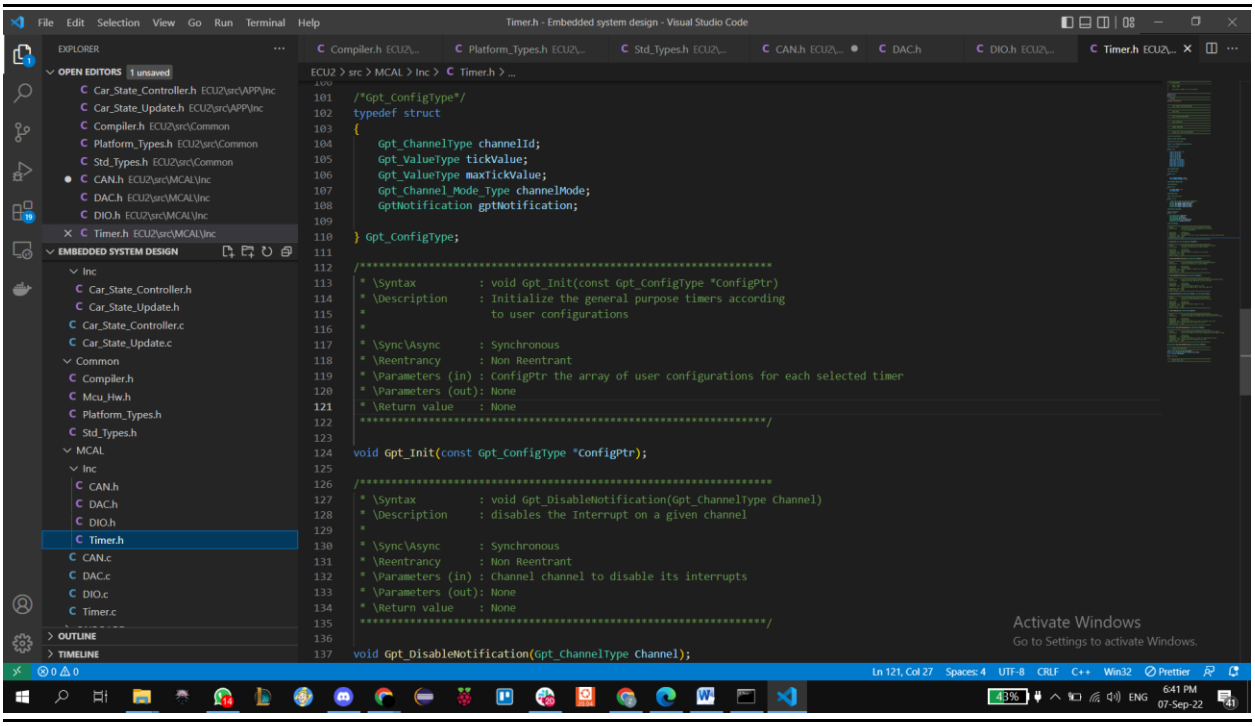

File Edit Selection View Go Run Terminal Help
DIO.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS 1 unsaved
Car_State_Controller.h ECU2\src\APP\Inc
Car_State_Update.h ECU2\src\APP\Inc
Compiler.h ECU2\src\Common
Platform_Types.h ECU2\src\Common
Std_Types.h ECU2\src\Common
CAN.h ECU2\src\MCAL\Inc
DAC.h ECU2\src\MCAL\Inc
DIO.h ECU2\src\MCAL\Inc
Timer.h ECU2\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Inc
Car_State_Controller.h
Car_State_Update.h
Car_State_Controller.c
Car_State_Update.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
CAN.h
DAC.h
DIO.h
Timer.h
CAN.c
DAC.c
DIO.c
Timer.c
OUTLINE
TIMELINE
ECU2 > src > MCAL > Inc > DIO.h > __unnamed_enum_08ce_3
* \Return value : None
***/
void Write_Pin(Pin_Num pin , Port_Num port , uint8 value);
/***
* \Syntax : uint8 Read_Pin(Pin_Num pin , Port_Num port)
* \Description : returns the value on a given channel
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : pin the pin number
* port the port number
* \Parameters (out): None
* \Return value : uint8 the value on pin
***/
uint8 Read_Pin(Pin_Num pin , Port_Num port);
/***
* \Syntax : void Write_Port(Port_Num PortId, uint32 Level)
* \Description : writes a value on a chosen port in the gpio
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : PortId the port number to write on
* Level the value to write on the desired port
* \Parameters (out): None
* \Return value : None
***/
void Write_Port(Port_Num PortId, uint32 Level);
/***
* \Syntax : uint32 Read_Port(Port_Num PortId);
* \Description : returns the value on a chosen port in the gpio
*
Activate Windows
Go to Settings to activate Windows.
Ln 25, Col 14 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
ENG 6:41 PM 07-Sep-22

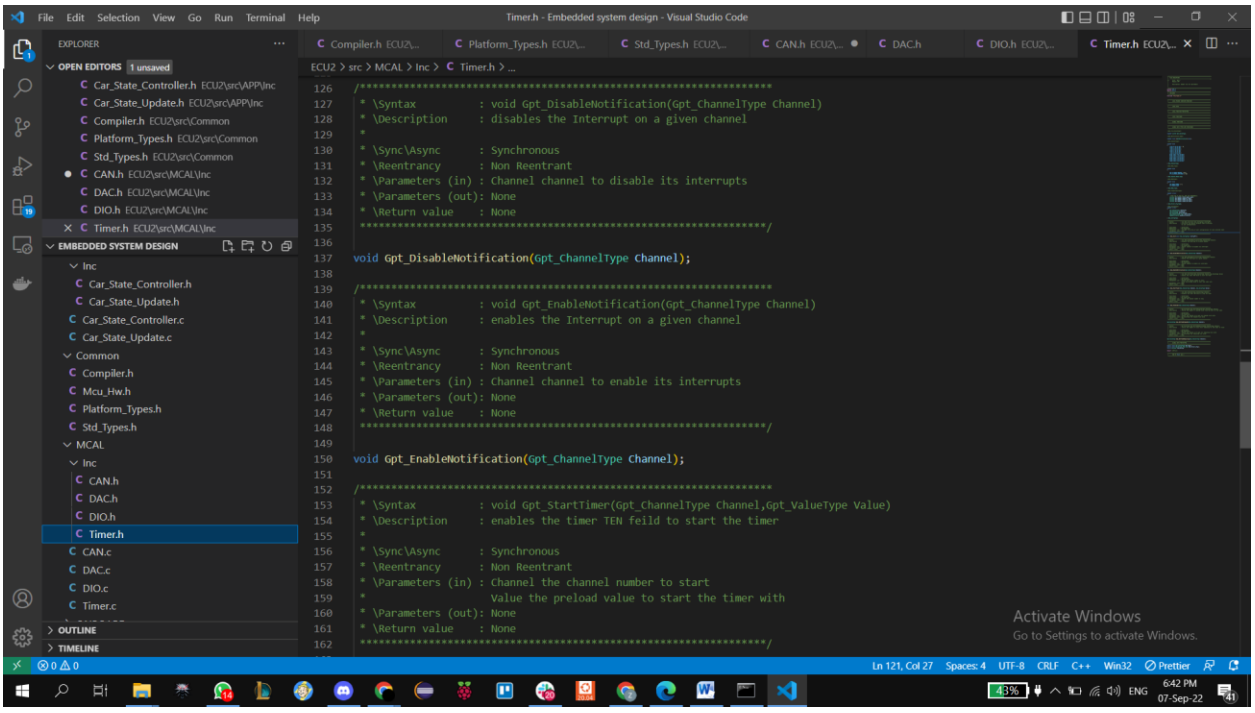


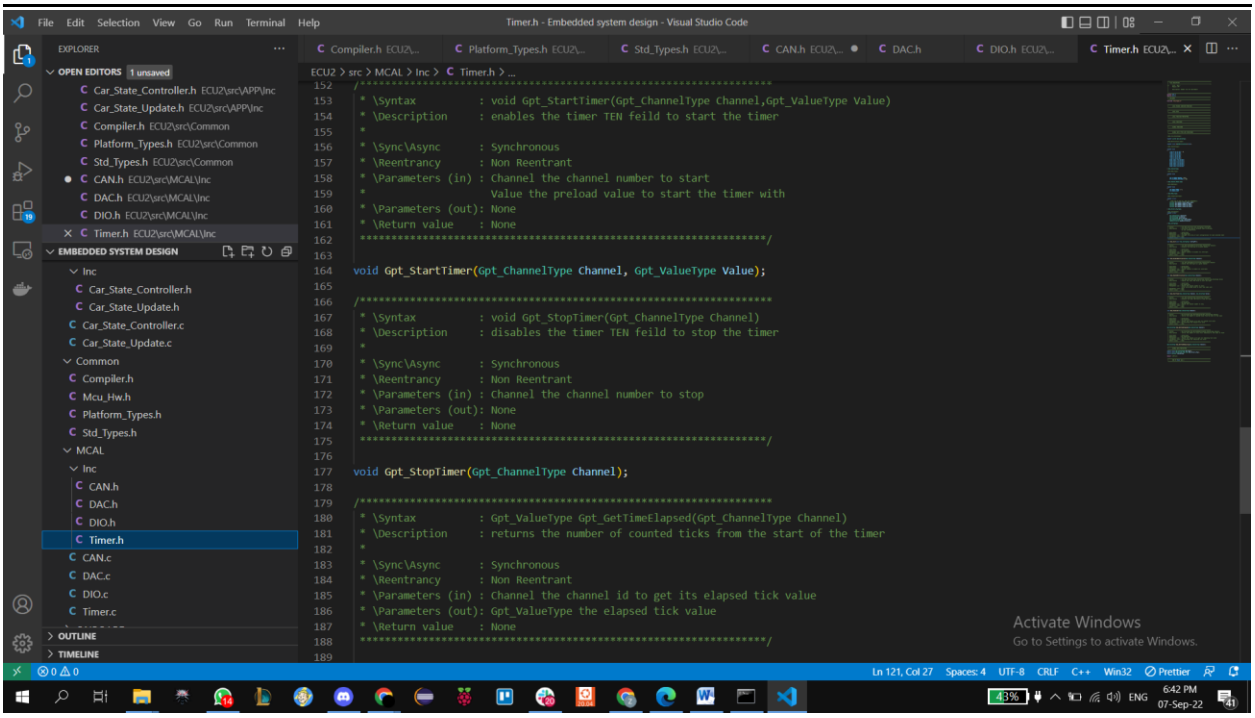
7. Timer_Module_APIs:

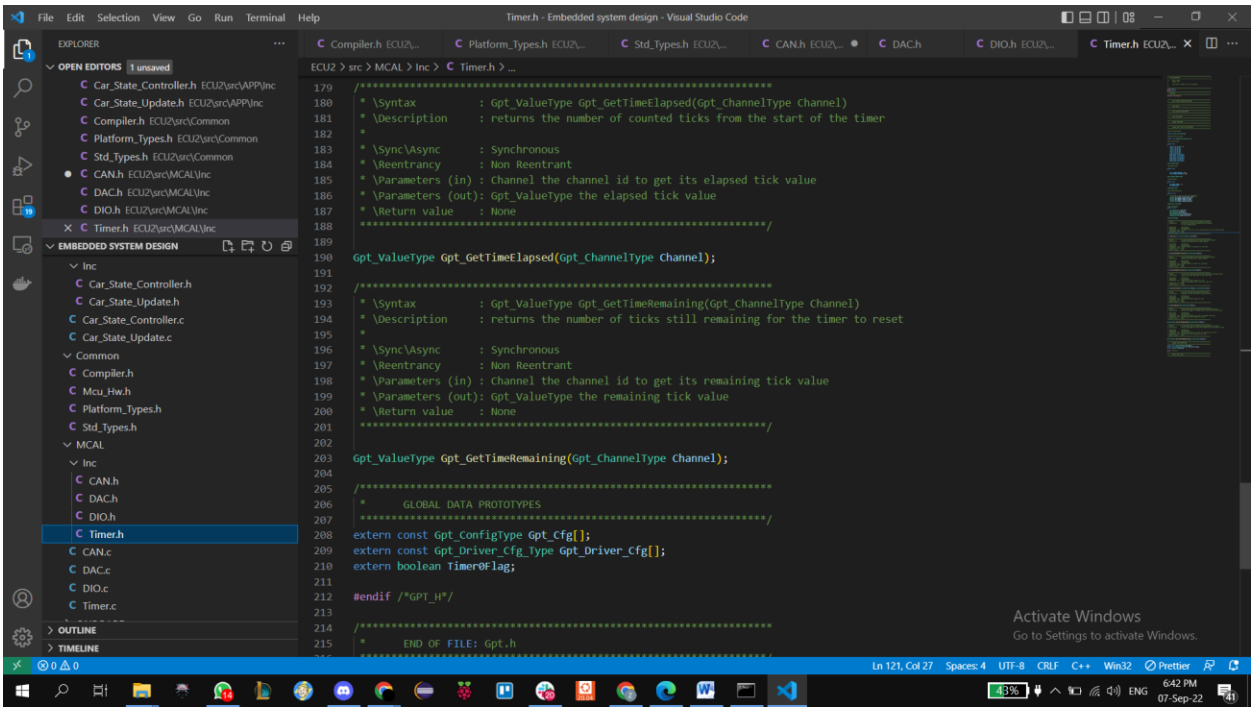


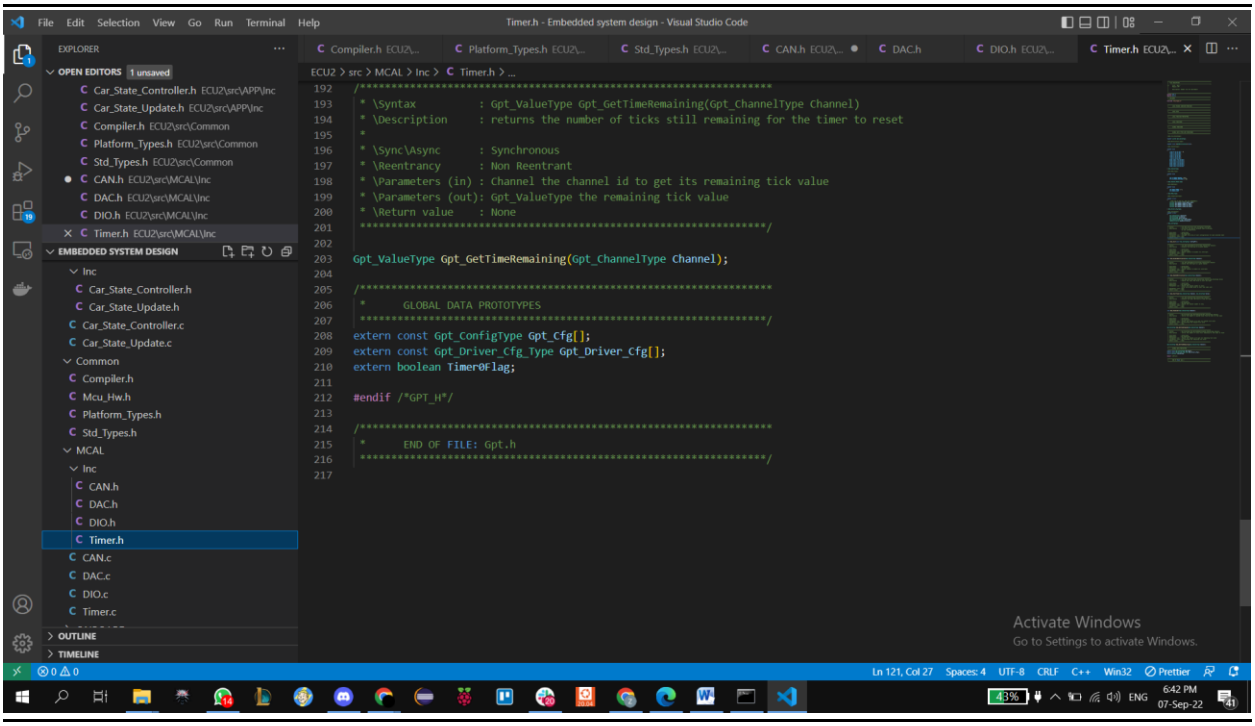

File Edit Selection View Go Run Terminal Help
Timer.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS 1 unsaved
Car_State_Controller.h ECU2\src\APP\Inc
Car_State_Update.h ECU2\src\APP\Inc
Compiler.h ECU2\src\Common
Platform_Types.h ECU2\src\Common
Std_Types.h ECU2\src\Common
CAN.h ECU2\src\MCAL\Inc
DAC.h ECU2\src\MCAL\Inc
DIO.h ECU2\src\MCAL\Inc
Timer.h ECU2\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Inc
Car_State_Controller.h
Car_State_Update.h
Car_State_Controller.c
Car_State_Update.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
CAN.h
DAC.h
DIO.h
Timer.h
CAN.c
DAC.c
DIO.c
Timer.c
OUTLINE
TIMELINE
ECU2 > src > MCAL > Inc > Timer.h > ...
typedef enum
{
GPT_CH_MODE_ONESHOT = 0x1,
GPT_CH_MODE_CONTINOUS = 0x2
} Gpt_Channel_Mode_Type;
/*Gpt_ModeType*/
typedef enum
{
GPT_MODE_NORMAL = 0,
GPT_MODE_SLEEP
} Gpt_Mode_Type;
/*Gpt_PredefTimerType*/
typedef struct
{
/*TODO SEE IF YOU WILL ADD TICK FREQURNCY*/
boolean GPT_PREDEF_TIMER_100US_32BIT;
boolean GPT_PREDEF_TIMER_US_16BIT;
boolean GPT_PREDEF_TIMER_1US_24BIT;
boolean GPT_PREDEF_TIMER_1US_32BIT;
} Gpt_Driver_Cfg_Type;
/*Gpt_ConfigType*/
typedef struct
{
Gpt_ChannelType channelId;
Gpt_ValueType tickValue;
Gpt_ValueType maxTickValue;
Gpt_Channel_Mode_Type channelMode;
Activate Windows
Go to Settings to activate Windows.
Ln 121, Col 27 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
ENG 6:41 PM 07-Sep-22


File Edit Selection View Go Run Terminal Help
Timer.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS 1 unsaved
Car_State_Controller.h ECU2\src\APP\Inc
Car_State_Update.h ECU2\src\APP\Inc
Compiler.h ECU2\src\Common
Platform_Types.h ECU2\src\Common
Std_Types.h ECU2\src\Common
CAN.h ECU2\src\MCAL\Inc
DAC.h ECU2\src\MCAL\Inc
DIO.h ECU2\src\MCAL\Inc
Timer.h ECU2\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Inc
Car_State_Controller.h
Car_State_Update.h
Car_State_Controller.c
Car_State_Update.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
CAN.h
DAC.h
DIO.h
Timer.h
CAN.c
DAC.c
DIO.c
Timer.c
OUTLINE
TIMELINE
ECU2 > src > MCAL > Inc > Timer.h > ...
/*Gpt_ConfigType*/
typedef struct
{
Gpt_ChannelType channelId;
Gpt_ValueType tickValue;
Gpt_ValueType maxTickValue;
Gpt_Channel_Mode_Type channelMode;
GptNotification gptNotification;
} Gpt_ConfigType;
/**********
* \Syntax : void Gpt_Init(const Gpt_ConfigType *ConfigPtr)
* \Description : Initialize the general purpose timers according
to user configurations
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : ConfigPtr the array of user configurations for each selected timer
* \Parameters (out): None
* \Return value : None
**********/
void Gpt_Init(const Gpt_ConfigType *ConfigPtr);
/**********
* \Syntax : void Gpt_DisableNotification(Gpt_ChannelType Channel)
* \Description : disables the Interrupt on a given channel
*
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : Channel channel to disable its interrupts
* \Parameters (out): None
* \Return value : None
**********/
void Gpt_DisableNotification(Gpt_ChannelType Channel);
Activate Windows
Go to Settings to activate Windows.
Ln 121, Col 27 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
ENG 6:41 PM 07-Sep-22

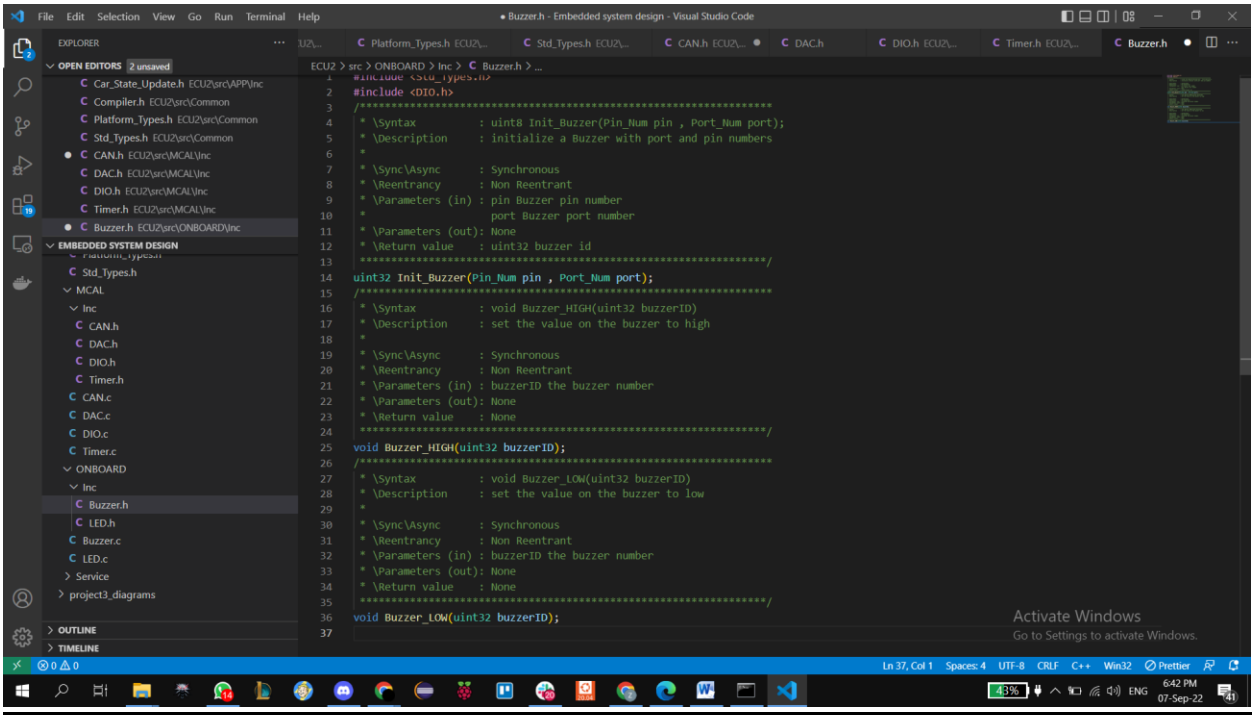

File Edit Selection View Go Run Terminal Help
Timer.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS 1 unsaved
Car_State_Controller.h ECU2\src\APP\Inc
Car_State_Update.h ECU2\src\APP\Inc
Compiler.h ECU2\src\Common
Platform_Types.h ECU2\src\Common
Std_Types.h ECU2\src\Common
CAN.h ECU2\src\MCAL\Inc
DAC.h ECU2\src\MCAL\Inc
DIO.h ECU2\src\MCAL\Inc
Timer.h ECU2\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Inc
Car_State_Controller.h
Car_State_Update.h
Car_State_Controller.c
Car_State_Update.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
CAN.h
DAC.h
DIO.h
Timer.h
CAN.c
DAC.c
DIO.c
Timer.c
OUTLINE
TIMELINE
Compiler.h ECU2\...
Platform_Types.h ECU2\...
Std_Types.h ECU2\...
CAN.h ECU2\...
DAC.h
DIO.h ECU2\...
Timer.h ECU2\...
ECU2 > src > MCAL > Inc > Timer.h > ...
126 /**
127 * \Syntax : void Gpt_DisableNotification(Gpt_ChannelType Channel)
128 * \Description : disables the Interrupt on a given channel
129 *
130 * \Sync\Async : Synchronous
131 * \Reentrancy : Non Reentrant
132 * \Parameters (in) : Channel channel to disable its interrupts
133 * \Parameters (out): None
134 * \Return value : None
135 **/
136
137 void Gpt_DisableNotification(Gpt_ChannelType Channel);
138
139 /**
140 * \Syntax : void Gpt_EnableNotification(Gpt_ChannelType Channel)
141 * \Description : enables the Interrupt on a given channel
142 *
143 * \Sync\Async : Synchronous
144 * \Reentrancy : Non Reentrant
145 * \Parameters (in) : Channel channel to enable its interrupts
146 * \Parameters (out): None
147 * \Return value : None
148 **/
149
150 void Gpt_EnableNotification(Gpt_ChannelType Channel);
151
152 /**
153 * \Syntax : void Gpt_StartTimer(Gpt_ChannelType Channel,Gpt_ValueType Value)
154 * \Description : enables the timer TEN feild to start the timer
155 *
156 * \Sync\Async : Synchronous
157 * \Reentrancy : Non Reentrant
158 * \Parameters (in) : Channel the channel number to start
159 * Value the preload value to start the timer with
160 * \Parameters (out): None
161 * \Return value : None
162 **/
Activate Windows
Go to Settings to activate Windows.
Ln 121, Col 27 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
43% ENG 6:42 PM 07-Sep-22


File Edit Selection View Go Run Terminal Help
Timer.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS 1 unsaved
Car_State_Controller.h ECU2\src\APP\Inc
Car_State_Update.h ECU2\src\APP\Inc
Compiler.h ECU2\src\Common
Platform_Types.h ECU2\src\Common
Std_Types.h ECU2\src\Common
CAN.h ECU2\src\MCAL\Inc
DAC.h ECU2\src\MCAL\Inc
DIO.h ECU2\src\MCAL\Inc
Timer.h ECU2\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Inc
Car_State_Controller.h
Car_State_Update.h
Car_State_Controller.c
Car_State_Update.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
CAN.h
DAC.h
DIO.h
Timer.h
CAN.c
DAC.c
DIO.c
Timer.c
OUTLINE
TIMELINE
Compiler.h ECU2\...
Platform_Types.h ECU2\...
Std_Types.h ECU2\...
CAN.h ECU2\...
DAC.h
DIO.h ECU2\...
Timer.h ECU2\...
ECU2 > src > MCAL > Inc > Timer.h > ...
153 * \Syntax : void Gpt_StartTimer(Gpt_ChannelType Channel,Gpt_ValueType Value)
154 * \Description : enables the timer TEN feild to start the timer
155 *
156 * \Sync\Async : Synchronous
157 * \Reentrancy : Non Reentrant
158 * \Parameters (in) : Channel the channel number to start
159 * Value the preload value to start the timer with
160 * \Parameters (out): None
161 * \Return value : None
162 **/
163
164 void Gpt_StartTimer(Gpt_ChannelType Channel, Gpt_ValueType Value);
165
166 /**
167 * \Syntax : void Gpt_StopTimer(Gpt_ChannelType Channel)
168 * \Description : disables the timer TEN feild to stop the timer
169 *
170 * \Sync\Async : Synchronous
171 * \Reentrancy : Non Reentrant
172 * \Parameters (in) : Channel the channel number to stop
173 * \Parameters (out): None
174 * \Return value : None
175 **/
176
177 void Gpt_StopTimer(Gpt_ChannelType Channel);
178
179 /**
180 * \Syntax : Gpt_ValueType Gpt_GetTimeElapsed(Gpt_ChannelType Channel)
181 * \Description : returns the number of counted ticks from the start of the timer
182 *
183 * \Sync\Async : Synchronous
184 * \Reentrancy : Non Reentrant
185 * \Parameters (in) : Channel the channel id to get its elapsed tick value
186 * \Parameters (out): Gpt_ValueType the elapsed tick value
187 * \Return value : None
188 **/
189
Activate Windows
Go to Settings to activate Windows.
Ln 121, Col 27 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
43% ENG 6:42 PM 07-Sep-22

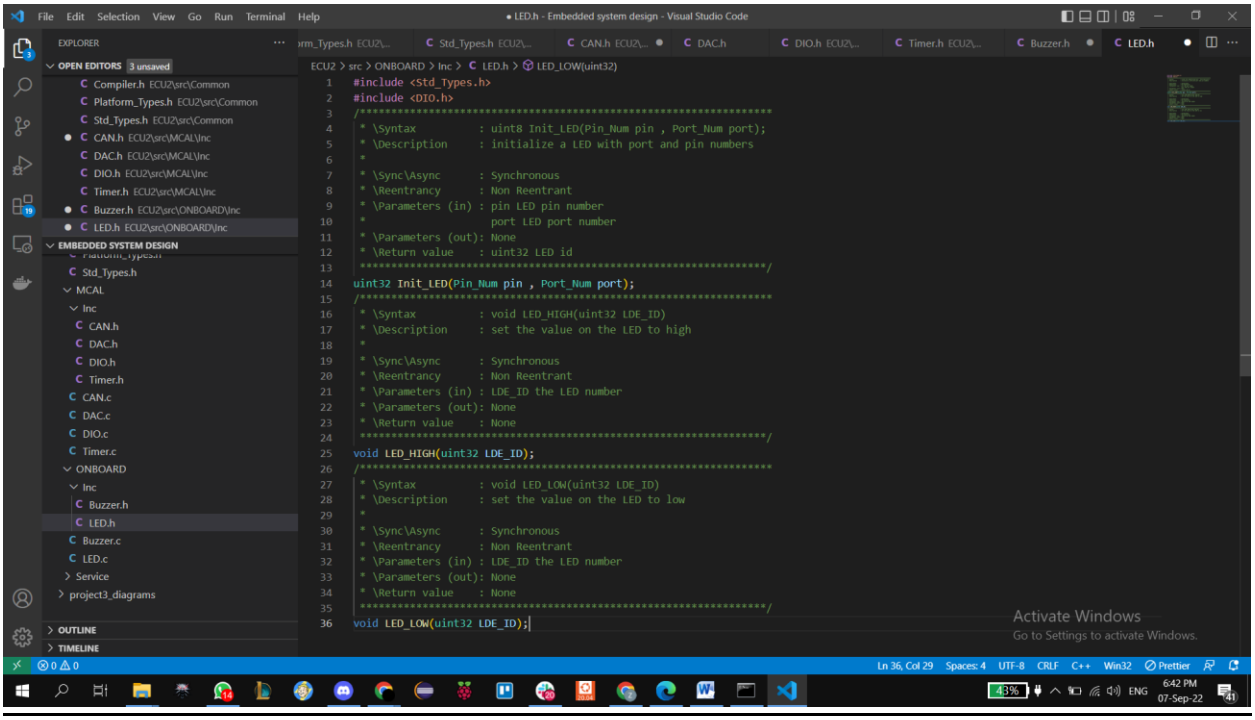

File Edit Selection View Go Run Terminal Help
Timer.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS 1 unsaved
Car_State_Controller.h ECU2\src\APP\Inc
Car_State_Update.h ECU2\src\APP\Inc
Compiler.h ECU2\src\Common
Platform_Types.h ECU2\src\Common
Std_Types.h ECU2\src\Common
CAN.h ECU2\src\MCAL\Inc
DAC.h ECU2\src\MCAL\Inc
DIO.h ECU2\src\MCAL\Inc
Timer.h ECU2\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Inc
Car_State_Controller.h
Car_State_Update.h
Car_State_Controller.c
Car_State_Update.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
CAN.h
DAC.h
DIO.h
Timer.h
CAN.c
DAC.c
DIO.c
Timer.c
OUTLINE
TIMELINE
ECU2 > src > MCAL > Inc > Timer.h > ...
* \Syntax : Gpt_ValueType Gpt_GetTimeElapsed(Gpt_ChannelType Channel)
* \Description : returns the number of counted ticks from the start of the timer
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : Channel the channel id to get its elapsed tick value
* \Parameters (out): Gpt_ValueType the elapsed tick value
* \Return value : None
Gpt_ValueType Gpt_GetTimeElapsed(Gpt_ChannelType Channel);
* \Syntax : Gpt_ValueType Gpt_GetTimeRemaining(Gpt_ChannelType Channel)
* \Description : returns the number of ticks still remaining for the timer to reset
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : Channel the channel id to get its remaining tick value
* \Parameters (out): Gpt_ValueType the remaining tick value
* \Return value : None
Gpt_ValueType Gpt_GetTimeRemaining(Gpt_ChannelType Channel);
* GLOBAL DATA PROTOTYPES
extern const Gpt_ConfigType Gpt_Cfg[];
extern const Gpt_Driver_Cfg_Type Gpt_Driver_Cfg[];
extern boolean Timer0Flag;
#endif /*GPT_H*/
* END OF FILE: Gpt.h
Activate Windows
Go to Settings to activate Windows.
Ln 121, Col 27 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
43% ENG 6:42 PM 07-Sep-22


File Edit Selection View Go Run Terminal Help
Timer.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS 1 unsaved
Car_State_Controller.h ECU2\src\APP\Inc
Car_State_Update.h ECU2\src\APP\Inc
Compiler.h ECU2\src\Common
Platform_Types.h ECU2\src\Common
Std_Types.h ECU2\src\Common
CAN.h ECU2\src\MCAL\Inc
DAC.h ECU2\src\MCAL\Inc
DIO.h ECU2\src\MCAL\Inc
Timer.h ECU2\src\MCAL\Inc
EMBEDDED SYSTEM DESIGN
Inc
Car_State_Controller.h
Car_State_Update.h
Car_State_Controller.c
Car_State_Update.c
Common
Compiler.h
Mcu_Hw.h
Platform_Types.h
Std_Types.h
MCAL
Inc
CAN.h
DAC.h
DIO.h
Timer.h
CAN.c
DAC.c
DIO.c
Timer.c
OUTLINE
TIMELINE
ECU2 > src > MCAL > Inc > Timer.h > ...
* \Syntax : Gpt_ValueType Gpt_GetTimeRemaining(Gpt_ChannelType Channel)
* \Description : returns the number of ticks still remaining for the timer to reset
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : Channel the channel id to get its remaining tick value
* \Parameters (out): Gpt_ValueType the remaining tick value
* \Return value : None
Gpt_ValueType Gpt_GetTimeRemaining(Gpt_ChannelType Channel);
* GLOBAL DATA PROTOTYPES
extern const Gpt_ConfigType Gpt_Cfg[];
extern const Gpt_Driver_Cfg_Type Gpt_Driver_Cfg[];
extern boolean Timer0Flag;
#endif /*GPT_H*/
* END OF FILE: Gpt.h
Activate Windows
Go to Settings to activate Windows.
Ln 121, Col 27 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
43% ENG 6:42 PM 07-Sep-22

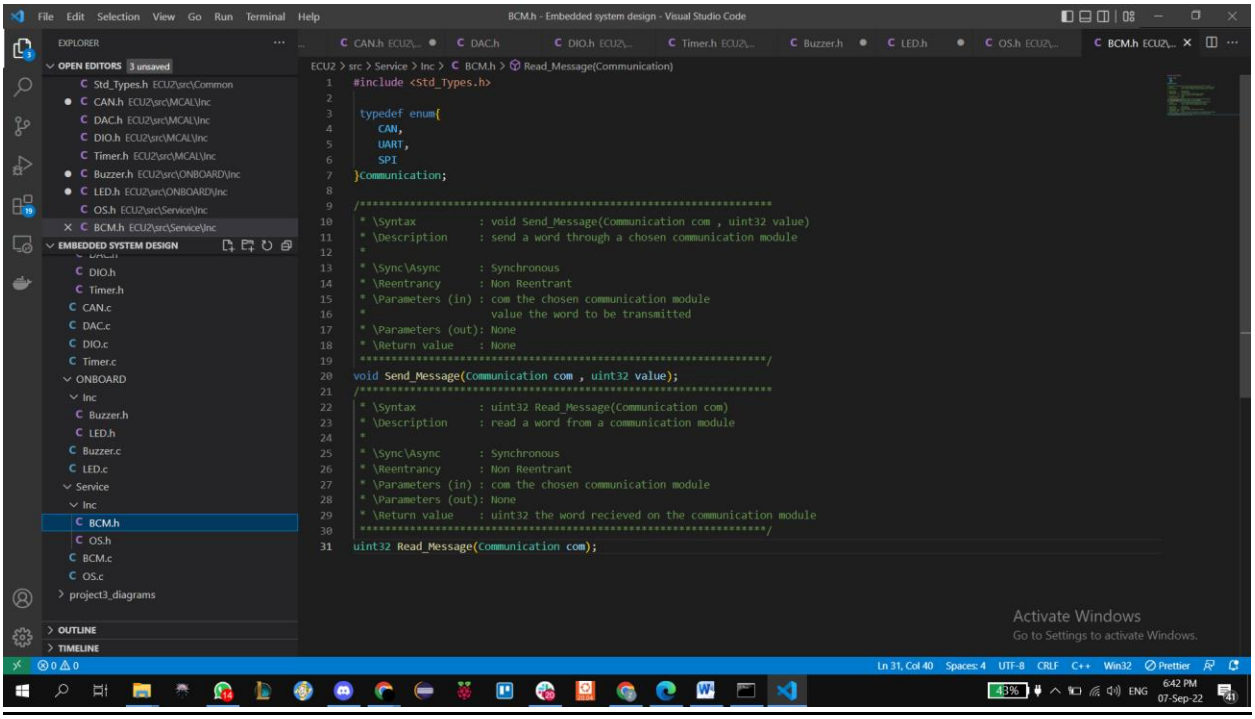
8. Buzzer_Module_APIs:



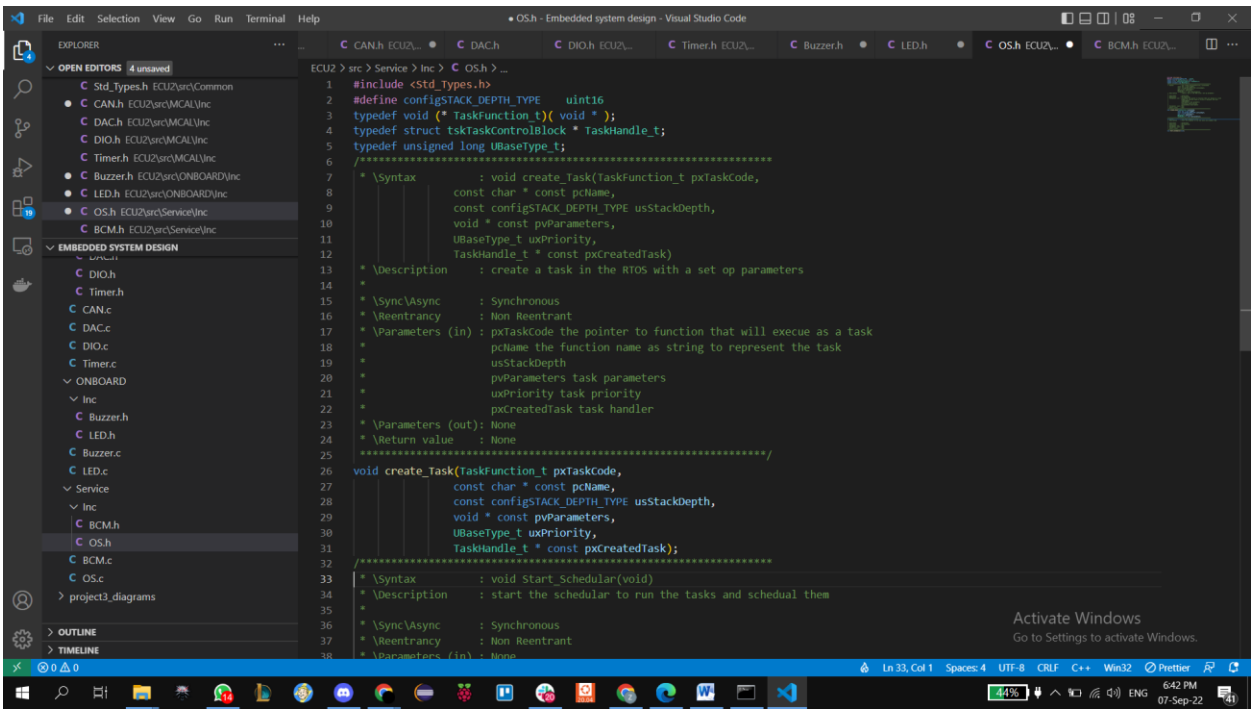
9. LED_Module_APIs:

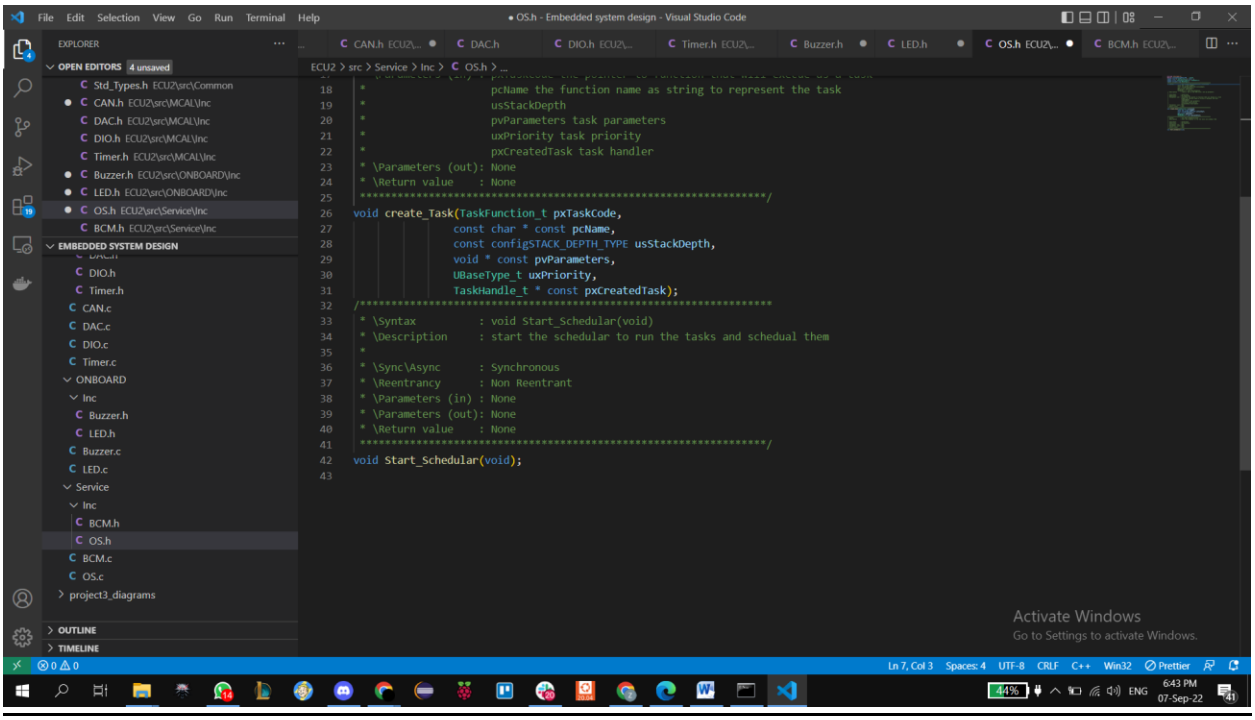


10. BCM_Module_APIs:



11. OS_Module_APIs:




File Edit Selection View Go Run Terminal Help
• OS.h - Embedded system design - Visual Studio Code
EXPLORER
OPEN EDITORS 4 unsaved
Std_Types.h ECU2\src\Common
CAN.h ECU2\src\MCAL\Inc
DAC.h ECU2\src\MCAL\Inc
DIO.h ECU2\src\MCAL\Inc
Timer.h ECU2\src\MCAL\Inc
Buzzer.h ECU2\src\ONBOARD\Inc
LED.h ECU2\src\ONBOARD\Inc
OS.h ECU2\src\Service\Inc
BCM.h ECU2\src\Service\Inc
EMBEDDED SYSTEM DESIGN
DIO.h
Timer.h
CAN.c
DAC.c
DIO.c
Timer.c
ONBOARD
Inc
Buzzer.h
LED.h
Buzzer.c
LED.c
Service
Inc
BCM.h
OS.h
BCM.c
OS.c
project3_diagrams
OUTLINE
TIMELINE
CAN.h ECU2\... DAC.h DIO.h ECU2\... Timer.h ECU2\... Buzzer.h LED.h OS.h ECU2\... BCM.h ECU2\...
ECU2 > src > Service > Inc > OS.h > ...
* pcName the function name as string to represent the task
* usStackDepth
* pvParameters task parameters
* uxPriority task priority
* pxCreatedTask task handler
* \Parameters (out): None
* \Return value : None
void create_Task(TaskFunction_t pxTaskCode,
const char * const pcName,
const configSTACK_DEPTH_TYPE usStackDepth,
void * const pvParameters,
UBaseType_t uxPriority,
TaskHandle_t * const pxCreatedTask);
* \Syntax : void Start_Schedular(void)
* \Description : start the schedular to run the tasks and schedual them
* \Sync\Async : Synchronous
* \Reentrancy : Non Reentrant
* \Parameters (in) : None
* \Parameters (out): None
* \Return value : None
void Start_Schedular(void);
Activate Windows
Go to Settings to activate Windows.
Ln 7, Col 3 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier
44% ENG 6:43 PM 07-Sep-22