



Embedded Systems Advanced Nano-Degree Embedded Software Design

Automotive Door Control System Design

Static Design

Ahmad Aladdin Tohamy

tuhami.10.8@gmail.com
July Cohort 2022

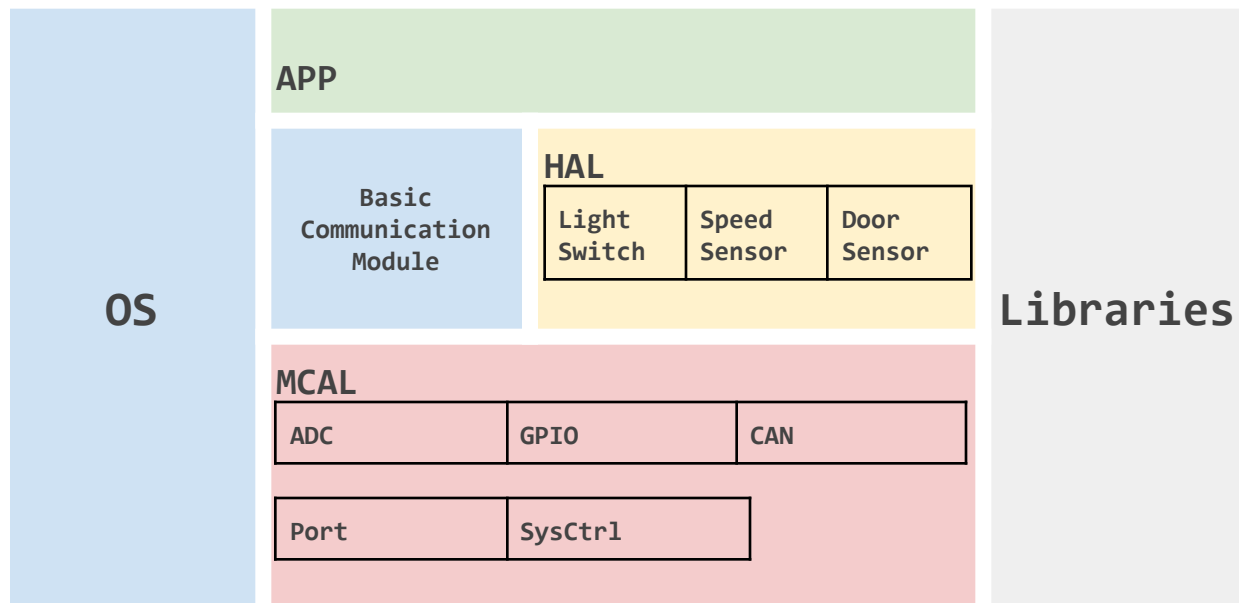
Table of Contents

Static Design Analysis	2
ECU 1	2
Layered Architecture	2
ECU 1 Components	2
ECU 1 Modules	2
APIs	3
Folder Structure	9
ECU 2	10
Layered Architecture	10
ECU 2 Components	10
ECU 2 Modules	10
APIs	11
Folder Structure	18

Static Design Analysis

1. ECU 1

- **Layered Architecture**



- **ECU 1 Components**

1) Door Sensor 2) Light Switch 3) Speed Sensor

- **ECU 1 Modules**

MCAL Layer	HAL Layer
1) General Purpose Input Output Module	1) Light Switch Module
2) Analog-to-Digital Converter	2) Speed Sensor Module
3) Controller Area Network Module	3) Door Sensor Module
4) Port Module	
5) System Control Module	
Service Layer	
1) Operating System	2) Basic Communication Module

- **APIs**

Port Module:

Function Name:		void PORT_Init (const Port_ConfigType * Port_ConfigArray)
Arguments :	Input:	Name : Port_ConfigArray
		Type : Pointer to Port_ConfigType Port_ConfigType is an unsigned char
		Range : Array size is hardware dependant as each element represents a pin Each element range is hardware dependant as well. We may assume 0-15 (the number of possible functionalities) as an example for illustration.
		Macros : which represent each pin possible functionalities according to data sheet For ex : PA0_DIO , PA1_GPT , PA2_ADC , PA3_CAN_TX , etc ..
		Description : Specifies each pin configuration
	Output:	None
Return :		None
Synchronous: Yes		Reentrant: Yes
Description:		This function sets Initializes each Pin with its desired functionality

SysCtrl Module:

Function Name:		void SysCtrl_MicrocontrollerInit (void)
Arguments :	Input:	Macros from SysCtrl_Configure.h header file
		Range : each configuration Macro has a range which is data sheet dependant
		Description : Specifies Microcontroller clock configuration
	Output:	None
Return :		None
Synchronous:		Yes
Reentrant:		Yes
Description:		This function Initializes necessary configurations for Microcontroller such as system clock , peripherals configurations

General Purpose Input Output Module:

Function Name:		GPIO_LevelType GPIO_ReadChannel (GPIO_ChannelType ChannelId);
Arguments :	Input:	Name : ChannelId
		Type : GPIO_ChannelType (An enum of microcontroller GPIO channels)
		Range : 0-Number of GPIO Channels (Hardware dependant)
		Variable / Macro : Macro
		Description : Indicates which GPIO channel to read from
	Output:	Name : -
		Type : GPIO_LevelType (An enum representing High/Low levels)
		Range : 0-1
		Variable / Macro : Variable
		Description : Indicates GPIO channel current level
Return :		GPIO_LevelType
Synchronous:		Yes
Reentrant:		No
Description:		This function receives input level from specified Pin Used typedefs GPIO_ChannelType : Specifies which channel to read from GPIO_LevelType : Specifies channel level (High/Low)

ADC Module:

Function Name:		void ADC_Init(void);
Arguments :	Input:	Macros from ADC_Configure.h header file
		Range : each configuration Macro has a range which is data sheet dependant
		Description : Specifies ADC configurations
	Output:	None
Return :		None
Synchronous:		Yes
Reentrant:		Yes
Description:		This function Initializes necessary configurations for Analog-to-Digital Converter Module

Function Name:		u8 ADC_StartConversion(ADC_ChannelType ChannelId);
Arguments :	Input:	Name : ChannelId
		Type : ADC_ChannelType
		Range : 0-Number of ADC channels (HW Dependant)
		Variable/Macro : Macro
		Description : Indicates which ADC channel to read from
	Output:	Name : -
		Type : unsigned char (u8)
		Range : 0-255
		Variable/Macro : Variable
		Description : Converted Digital Data
Return :		u8
Synchronous: Yes		Reentrant: No
Description:		<p>This function receives input level from specified Pin</p> <p>Used typedefs</p> <p>ADC_ChannelType : Specifies which channel to read signal from</p>

CAN Module:

Function Name:		void CAN1_Init(void);
Arguments :	Input:	Name : -
		Type : -
		Range : each configuration has a different range
		Variable / Macro : Macros
		Description : CAN1 Module Configurations
	Output:	None
Return :		None
Synchronous:		Yes
Reentrant:		Yes
Description:		This function Initializes necessary configurations for CAN Module

Function Name:		void CAN1_TransmitMessage(void);
Arguments :	Input:	Passed by writing over TxMailBox
		Type : unsigned char
		Range : 0-255
		Variable / Macro : Variable
		Description : Message content
	Output:	None
Return :		None
Synchronous:		Yes
Reentrant:		No
Description:		This function Transmits a message to CAN Transceiver

Light Switch Module:

Function Name:		LightSwitch_StateType LightSwitch_getState(void);
Arguments :	Input:	None
	Output:	Name : -
		Type : LightSwitch_StateType (High/Low)
		Range : 0-1
		Variable / Macro : Variable
		Description : Light Switch Current state
Return :		LightSwitch_StateType
Synchronous:		Yes
Reentrant:		Yes
Description:		This function gets the current light switch state Used Typedefs LightSwitch_StateType : Specifies switch level (ON/OFF)

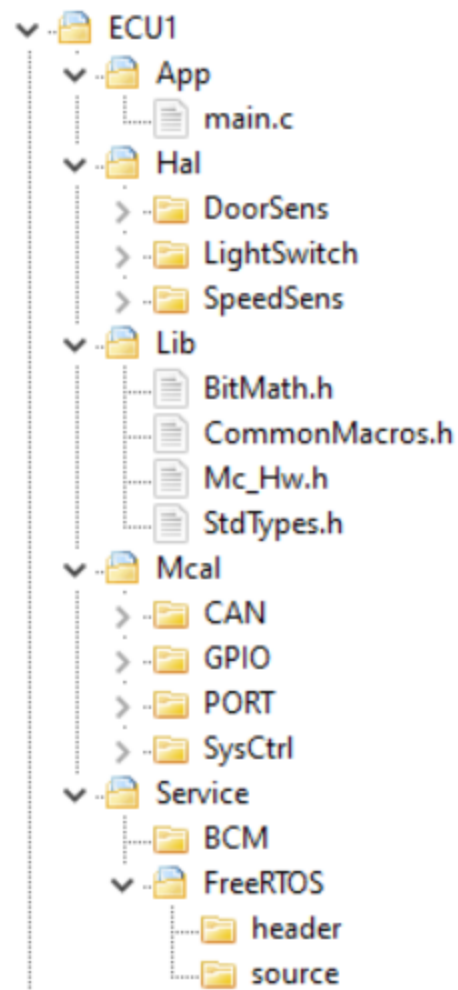
Speed Sensor Module:

Function Name:		u32 SpeedSens_getSpeed(void);
Arguments :	Input:	None
	Output:	Name : -
		Type : unsigned integer
		Range : 0-4294967295
		Variable / Macro : Variable
		Description : Speed Sensor Current value
Return :		u32
Synchronous:		Yes
Reentrant:		No
Description:		This function gets the digital form of a speed sensor

Door Sensor Module:

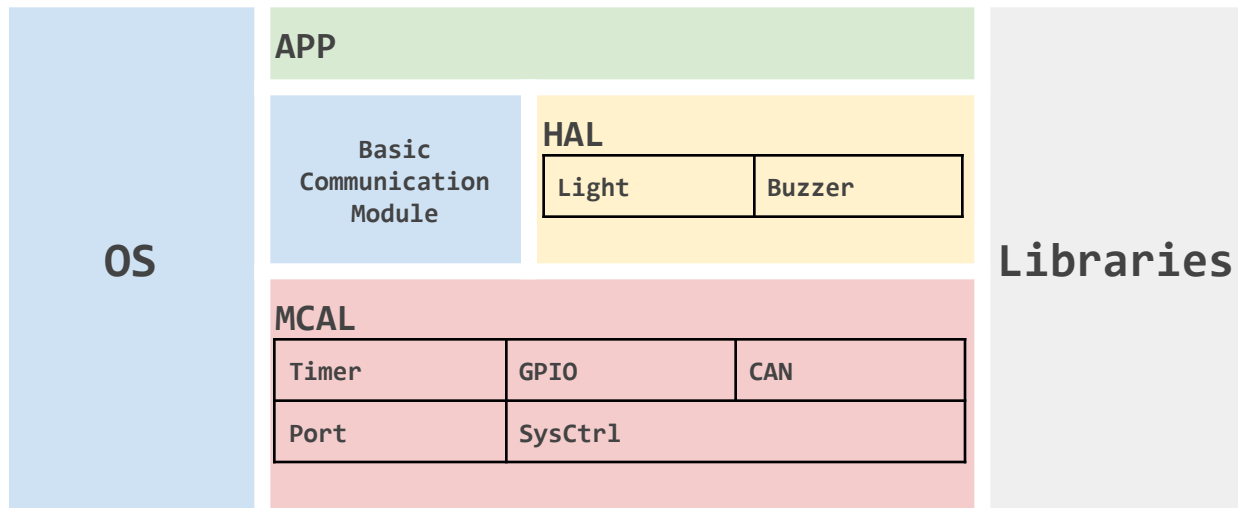
Function Name:		DoorSens_StateType DoorSens_getState(void);
Arguments :	Input:	None
	Output:	Name : -
		Type : DoorSens_StateType (Open/Closed)
		Range : 0-1
		Variable / Macro : Variable
		Description : Door Current state
Return :		DoorSens_StateType
Synchronous:		Yes
Reentrant:		No
Description:		This function gets the current light switch state Used Typedefs DoorSens_StateType : Specifies Door state (Open/Closed)

- **Folder Structure**



2. ECU 2

- **Layered Architecture**



- **ECU 2 Components**

- 1) Right Light
- 2) Left Light
- 3) Buzzer

- **ECU 2 Modules**

MCAL Layer	HAL Layer
1) General Purpose Input Output Module 2) General Purpose Timers Module 3) Controller Area Network Module 5) Port Module 6) System Control Module	1) Lights Module 2) Buzzer Module
Service Layer	
1) Operating System 2) Basic Communication Module	

- **APIs**

There are many common API between ECU1 and ECU2 such as:

Port Module: void PORT_Init (const u8 PinConfig)

SysCtrl Module: void SysCtrl_MicrocontrollerInit (void)

General Purpose Input Output Module:

GPIO_LevelType GPIO_ReadChannel (GPIO_ChannelType ChannelId);

CAN Module: void CAN1_Init(void)

General Purpose Timers Module:

Function Name:	void GPT_Init (Gpt_ConfigType * GPT_ConfigArray)	
Arguments :	Input:	Name : GPT_ConfigArray
		Type : Array of Gpt_ConfigType Gpt_ConfigType is a structure which represents each pin name and configurations
		Range : Array size is hardware dependant as each element represents a GPT channel .
		Macros : which represent each channel configurations
		Description : Specifies each GPT channel configuration
	Output:	None
Return :	Void	
Synchronous:	Yes	
Reentrant:	No	
Description:	<p>This function initializes the microcontroller timer with desired configurations</p> <p><u>Used typedefs</u></p> <p>Gpt_ConfigType : Contains configurations associated with timers such as (Channel Id , Channel Mode , Channel Tick Frequency , etc..)</p>	

Function Name:		void GPT_StartTimer(Gpt_ChannelType Channel, Gpt_ValueType Counts);
Arguments :	Input:	Name : Channel
		Type : Gpt_ChannelType
		Range : 0-Number of GPT Channels (HW dependant)
		Variable / Macro : Macro
		Description : Specifies which GPT channel to start
	Input:	Name: Ticks
		Type : Gpt_ValueType (unsigned integer)
		Range : 0-4294967295
		Variable / Macro : Variable
		Description : Specifies the number of ticks desired
	Output:	None
Return :		Void
Synchronous:		Yes
Reentrant:		No
Description:		<p>This function starts the specified timer with desired number of ticks</p> <p><u>Used typedefs</u></p> <p>Gpt_ChannelType : Contains all the channel IDs</p> <p>Gpt_ValueType : unsigned integer</p>

Function Name:		void GPT_StopTimer(Gpt_ChannelType Channel);
Arguments :	Input:	Name : Channel
		Type : Gpt_ChannelType
		Range : 0-Number of GPT Channels (HW dependant)
		Variable / Macro : Macro
		Description : Specifies which GPT channel to stop
	Output:	None
Return :		Void
Synchronous:		Yes
Reentrant:		No
Description:		<p>This function stops the specified timer with</p> <p><u>Used typedefs</u></p> <p>Gpt_ChannelType : Contains all the channel IDs</p>

General Purpose Input Output Module:

Function Name:		void GPIO_WriteChannel (GPIO_ChannelType ChannelId, GPIO_LevelType Level)
Arguments :	Input:	Name : ChannelId
		Type : GPIO_ChannelType
		Range : 0-Number of GPIO Channels (HW dependant)
		Variable / Macro : Macro
		Description : Specifies which GPIO channel to write over
	Input	Name: Level
		Type : GPIO_LevelType (High/Low)
		Range : 0-1
		Variable / Macro : Variable
		Description : Sets GPIO Channel level
Return :		None
Synchronous:		Yes
Reentrant:		Yes
Description:		This function sets specified Output Pin value as desired <u>Used typedefs</u> GPIO_ChannelType : Specifies which channel to write over GPIO_LevelType : Specifies desired level (High/Low)

CAN Module:

Function Name:		u8 CAN1_ReceiveMessage(void);
Arguments :	Input:	None
	Output:	Name : -
		Type : unsigned char (u8)
		Range : 0-255
		Variable/Macro : Variable
		Description : Received Data
Return :		u8
Synchronous:		Yes
Reentrant:		No
Description:		This function Receives a message from CAN Transceiver

Buzzer Module:

Function Name:		void Buzz_SetBuzzerON(void);
Arguments :	Input:	None
	Output:	None
Return :		None
Synchronous: Yes		Reentrant: No
Description:		This function Turns the buzzer on

Function Name:		void Buzz_SetBuzzerOFF(void);
Arguments :	Input:	None
	Output:	None
Return :		None
Synchronous: Yes		Reentrant: No
Description:		This function Turns the buzzer off

Lights Module:

Function Name:		void Lights_SetLightsON(void);
Arguments :	Input:	None
	Output:	None
Return :		None
Synchronous:		Yes
Reentrant:		No
Description:		This function Turns the Lights on

Function Name:		void Lights_SetLightsOFF(void);
Arguments :	Input:	None
	Output:	None
Return :		None
Synchronous:		Yes
Reentrant:		No
Description:		This function Turns the Lights Off

- **Folder Structure**

