Embedded Systems Advanced Nano-Degree
Embedded Software Design

Automotive Door Control System Design

Static Design

Ahmad Aladdin Tohamy

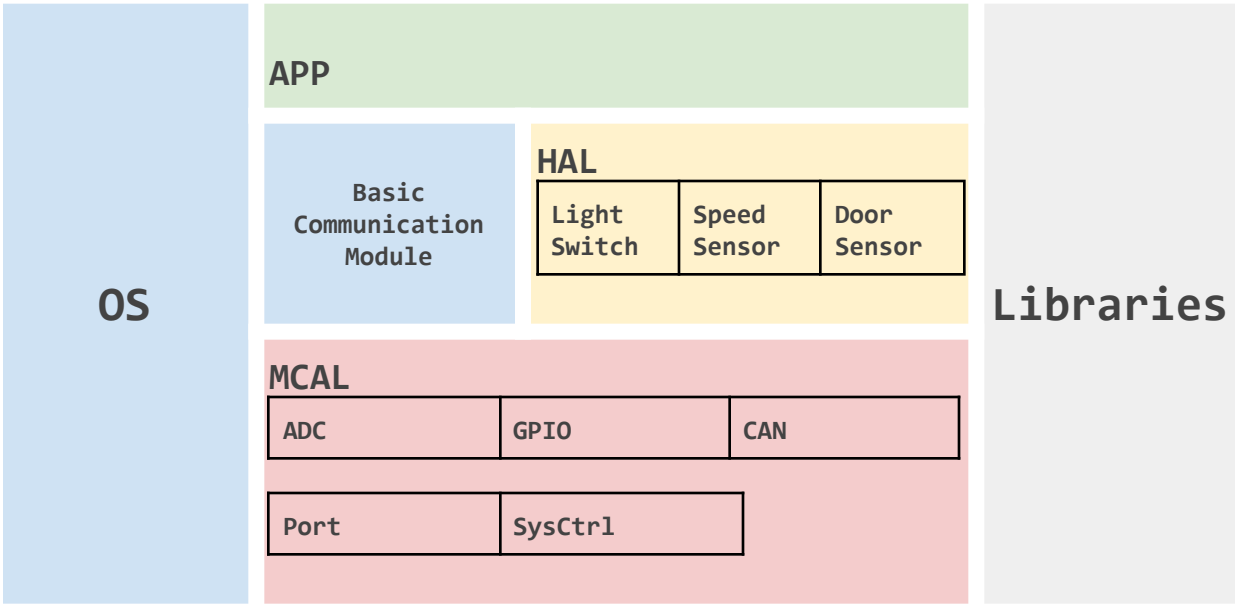
tuhami.10.8@gmail.com
July Cohort 2022

Table of Contents

Static Design Analysis

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

- **APIs**

Port Module:

Function Name:		void PORT_Init (const u8 PinConfig)
Arguments :	Input:	PinConfig : 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:	Hash definitions from SysCtrl_Configure.h header file
	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:	ChannelId : Indicates which pin to read from
	Output:	Channel 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:	Hash definitions from ADC_Configure.h header file
	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:	ChannelId : Indicates which pin to read from
	Output:	Converted Digital Data
Return :		u8
Synchronous:		Yes
Reentrant:		No
Description:		This function receives input level from specified Pin Used typedefs ADC_ChannelType : Specifies which channel to read signal from

---

CAN Module:

Function Name:		void CAN1_Init(void);
Arguments :	Input:	Hash definitions from CAN1_Configure.h header file
	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
	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:	Hash definition of Light Switch GPIO Channel Id
	Output:	LightSwitch 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)

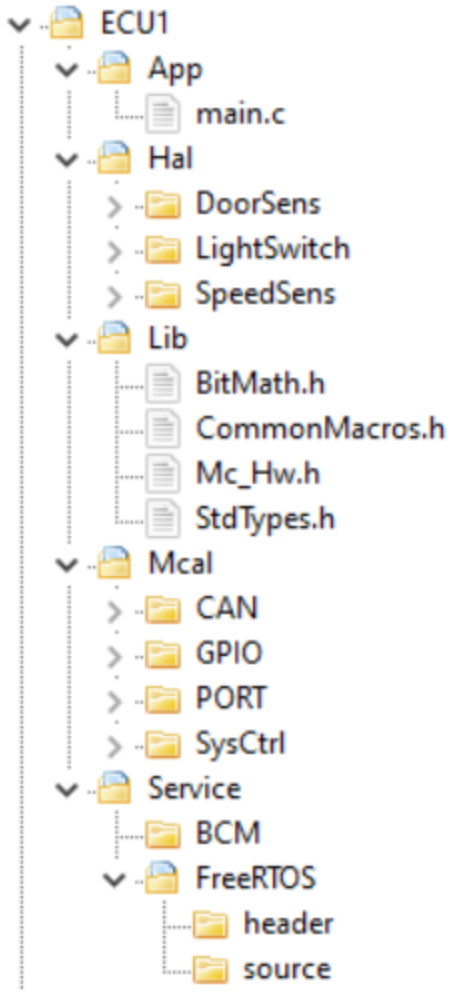
Speed Sensor Module:

Function Name:		u32 SpeedSens_getSpeed(void);
Arguments :	Input:	Hash definition of Speed Sensor Channel Id
	Output:	Current Speed
Return :		u32
Synchronous:		Yes
Reentrant:		No
Description:		This function gets the digital form of a speed sensor

Door Sensor Module:

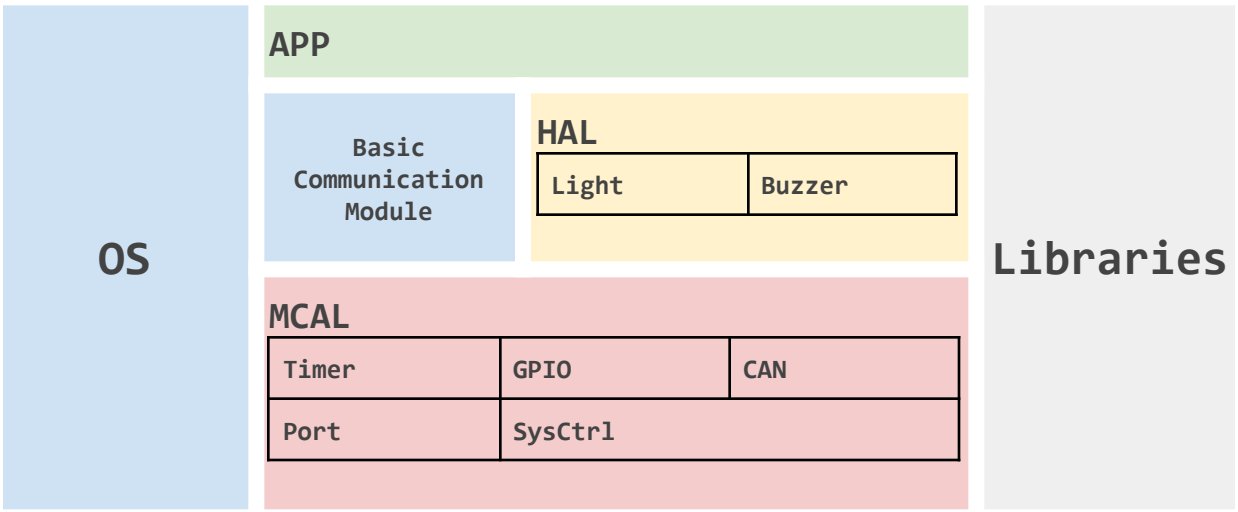
Function Name:		DoorSens_StateType DoorSens_getState(void);
Arguments :	Input:	Hash definition of Door Sensor GPIO Channel Id
	Output:	Door state
Return :		DoorSens_StateType
Synchronous:		Yes
Reentrant:		No
Description:		This function gets the current light switch state Used Typedefs DoorSens_StateType : 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:	GPT_ConfigArray (Array of desired configurations)
	Output:	None
Return :		Void
Synchronous:		Yes
Reentrant:		No
Description:		This function initializes the microcontroller timer with desired configurations Used typedefs Gpt_ConfigType : Contains configurations associated with timers such as (Channel Id , Channel Mode , Channel Tick Frequency , etc..)

Function Name:		void GPT_StartTimer(Gpt_ChannelType Channel, Gpt_ValueType Counts);
Arguments :	Input:	Channel : Specifies which timer to start Counts : Specifies the number of ticks desired
	Output:	None
Return :		Void
Synchronous:		Yes
Reentrant:		No
Description:		This function starts the specified timer with desired number of ticks Used typedefs Gpt_ChannelType : Contains all the channel IDs Gpt_ValueType : unsigned integer

Function Name:		void GPT_StopTimer(Gpt_ChannelType Channel);
Arguments :	Input:	Channel : Specifies which timer channel to stop
	Output:	None
Return :		Void
Synchronous:		Yes
Reentrant:		No
Description:		This function stops the specified timer with Used typedefs Gpt_ChannelType : Contains all the channel IDs

---

Function Name:		void GPT_nSecondsDelay(u32 time , Gpt_ChannelType Channel);
Arguments :	Input:	Channel : Specifies which timer to poll time : Specifies the number of seconds desired
	Output:	None
Return :		Void
Synchronous:		Yes
Reentrant:		No
Description:		This function is a busy wait implementation for the desired number of seconds Used typedefs Gpt_ChannelType : Contains all the channel IDs

General Purpose Input Output Module:

Function Name:		void GPIO_WriteChannel (GPIO_ChannelType ChannelId, GPIO_LevelType Level)
Arguments :	Input:	ChannelId : Indicates which pin to write over Level : Indicates the desired value to be written
Return :		None
Synchronous:		Yes
Reentrant:		Yes
Description:		This function sets specified Output Pin value as desired Used typedefs 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:	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**