UDACITY

Ministry of Communications and Information Technology

Automotive Door Control System

Static Design

Submitted by:

Ahmed Essam Salem

ahmed-medo1999@hotmail.com

Contents

List of Figures

System Schematic

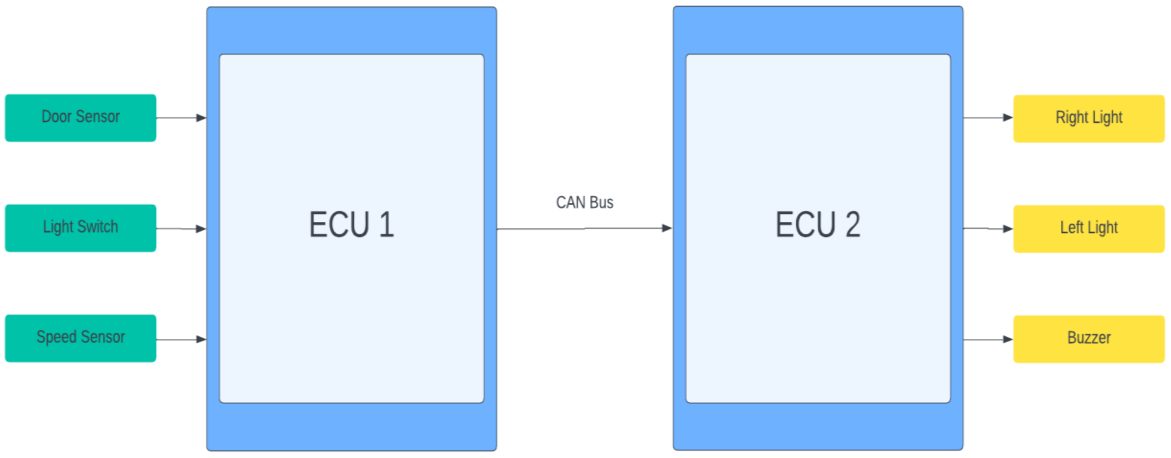

Figure 1 Block Diagram of System

ECU Layered Architecture

This section focuses on static views of a conceptual layered software architecture. The Layered Software Architecture describes the software architecture of AUTOSAR. It describes in an top-down approach the hierarchical structure of AUTOSAR software, maps the Basic Software Modules to software layers and shows their relationship.

ECU 1:

The first ECU features and general function are discussed in more detail in this section as an architecture overview

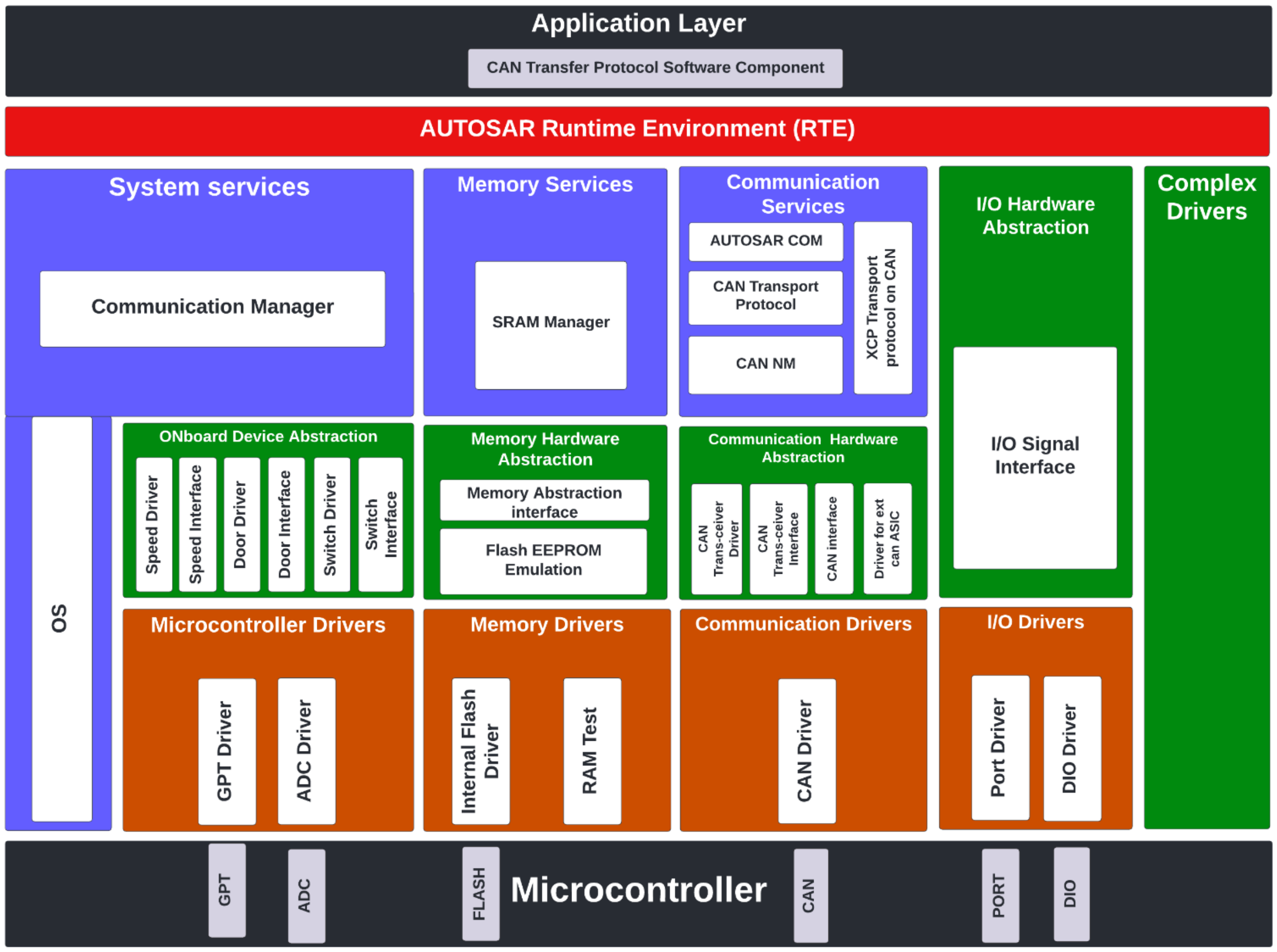

Figure 2 ECU 1 Layered Architecture

ECU 2:

The second ECU features and general function are discussed in more detail in this section as an architecture overview

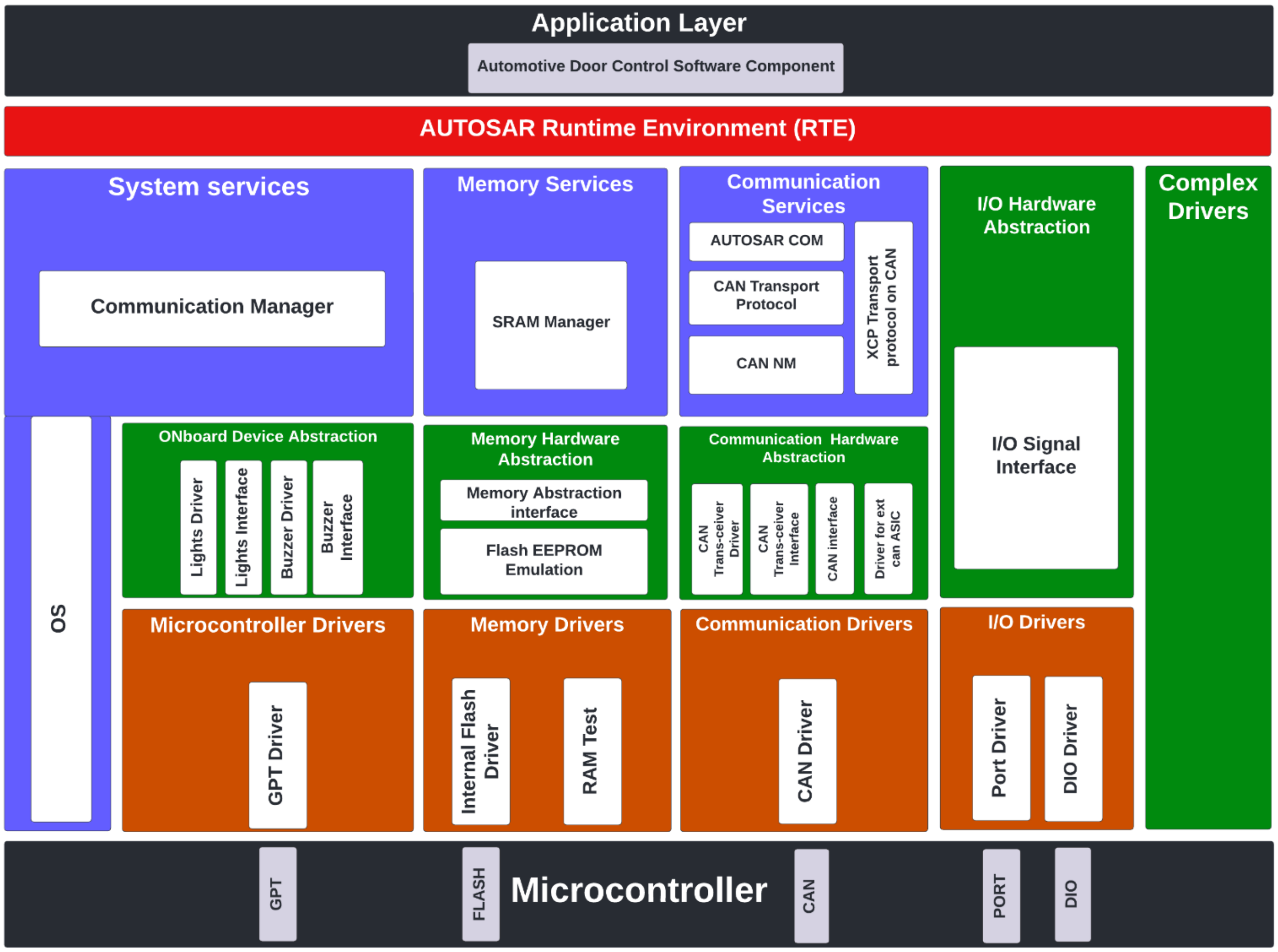

Figure 3 ECU 2 Layered Architecture

Modules API Specification and Type Definitions

ECU 1:

Port Driver:

Function Name:	Port_Init		
Arguments	Inputs	ConfigPtr	const Port_ConfigType*
		Pointer to configuration set.	
		-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the Port Driver module.		

Type definitions:

Name:	Port_ConfigType
Type:	Structure
Range:	The contents of the initialization data structure are specific to the microcontroller.
Description:	Type of the external data structure containing the initialization data for this module.

Name:	Port_PinType
Type:	uint
Range:	Shall cover all available port pins. The type should be chosen for the specific MCU platform (best performance).
Description:	Data type for the symbolic name of a port pin.

Name:	Port_PinDirectionType
Type:	Enumeration
Range:	PORT_PIN_IN: Sets port pin as input. PORT_PIN_OUT: Sets port pin as output.
Description:	Possible directions of a port pin.

Name:	Port_PinModeType
Type:	Enumeration
Range:	As several port pin modes shall be configurable on one pin, the range shall be determined by the implementation
Description:	Different port pin modes.

DIO Driver:

Function Name:	Dio_Init		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the DIO Driver module.		

Function Name:	Dio_ReadChannel		
Arguments	Inputs	ChannelId	Dio_ChannelType
		Description: ID of DIO Channel	
		-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	Dio_LevelType	STD_HIGH	The physical level of the corresponding Pin is STD_HIGH
		STD_LOW	The physical level of the corresponding Pin is STD_LOW
Description:	Returns the value of the specified DIO channel.		

Function Name:	Dio_WriteChannel		
Arguments	Inputs	ChannelId	Dio_ChannelType
		Description: ID of DIO Channel	
		Level	Dio_LevelType
		Description: Value to be written	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Service to set a level of a channel.		

Type definitions:

Name:	Dio_ChannelType
Type:	uint
Range:	Shall cover all available DIO channels
Description:	Numeric ID of a DIO channel.

Name:	Dio_LevelType
Type:	uint8
Range:	STD_LOW: 0x00 | Physical state 0V STD_HIGH: 0x01 | Physical state 3.3V or 5V
Description:	These are the possible levels a DIO channel can have (input or output)

CAN Driver:

Function Name:	CAN_Init		
Arguments	Inputs	Config	Can_ConfigType*
		Description: Pointer to driver configuration.	
		-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	This function initializes the CAN module.		

Function Name:	Can_SetBaudrate		
Arguments	Inputs	Controller	uint8
		Description: CAN controller, whose baud rate shall be set	
		BaudRateConfigID	uint16
		Description: references a baud rate configuration by ID	
	Output	None	None
	Input/Output	None	None
Return	E_OK	Service request accepted, setting of (new) baud rate started	
	E_NOK	Service request not accepted	
Description:	This service shall set the baud rate configuration of the CAN controller.Depending on necessary baud rate modifications the controller might have to reset		

Function Name:	CAN_Write		
Arguments	Inputs	Data	uint64
		Description: Data to be sent over CAN bus	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	Passes a CAN message to CanDrv for transmission.		

Function Name:	CAN_Read		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	uint8*		
Description:	Reads if a CAN message is sent on CanDrv for receive.		

Function Name:	CANIntHandler		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Reads the CAN interrupt controller status.		

Type definitions:

Name:	Can_ConfigType
Type:	Structure
Range:	
Description:	This is the type of the external data structure containing the overall initialization data for the CAN driver and SFR settings affecting all controllers. Furthermore it contains pointers to controller configuration structures. The contents of the initialization data structure are CAN hardware specific.

GPT Driver:

Function Name:	Gpt_Init		
Arguments	Inputs	ConfigPtr	Gpt_ConfigType*
		Description: Pointer to a selected configuration structure	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the GPT driver.		

Function Name:	Gpt_StartTimer		
Arguments	Inputs	Channel	Dio_ChannelType
		Description: Numeric identifier of the GPT channel.	
		Value	Dio_LevelType
		Description: Target time in number of ticks.	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Starts a timer channel.		

Function Name:	Gpt_StopTimer		
Arguments	Inputs	Channel	Gpt_ChannelType
		Description: Numeric identifier of the GPT channel.	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Stops a timer channel.		

Function Name:	TimerIntHandler		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Checks the timer call back pointer and calls function in upper layer.		

Function Name:	Timer_SetCallBack		
Arguments	Inputs	Ptr2Func	void
		Description: Address of Call Back Function	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	takes a pointer to function for upper layers function with the concept of call back.		

Type definitions:

Name:	Gpt_ConfigType
Type:	Structure
Range:	
Description:	This is the type of the data structure including the configuration set required for initializing the GPT timer unit.

Name:	Gpt_ChannelType
Type:	uint
Range:	Shall cover all available Gpt channels.
Description:	Numeric ID of a GPT channel.

Name:	Gpt_LevelType
Type:	uint8
Range:	Microcontroller dependent.
Description:	Type for reading and setting the timer values (in number of ticks)

ADC Driver:

Function Name:	ADC_Init		
Arguments	Inputs	ConfigPtr	ADC_ConfigType*
		Description: Pointer to a selected configuration structure	
		-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the ADC driver module.		

Function Name:	ADC_Read		
Arguments	Inputs	Channel	ADC_ChannelType
		Description: ID of ADC Channel	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	Returns the value of the specified ADC channel.		

Type definitions:

Name:	ADC_ConfigType
Type:	Structure
Range:	
Description:	This is the type of the data structure including the configuration set required for initializing the ADC module.

Name:	ADC_ChannelType
Type:	uint
Range:	Shall cover all available Gpt channels.
Description:	Numeric ID of an ADC channel.

Door Driver:

Function Name:	Door_Init		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the Door Sensor module.		

Function Name:	GetDoorStatus		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	DOOR_OPEN	0	
	DOOR_CLOSED	1	
Description:	Returns the value of the Door DIO channel.		

Light Switch Driver:

Function Name:	LightSwitch_Init		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the Light Switch module.		

Function Name:	GetSwitchStatus		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	SW_HIGH	0	
	SW_LOW	1	
Description:	Returns the value of the switch DIO channel.		

Speed Driver:

Function Name:	Speed_Init		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the Speed Sensor module.		

Function Name:	GetSpeedValue		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	Uint32		
Description:	Returns the value of the speed ADC channel.		

CAN TP Software Component:

Function Name:	CAN_TP_Init		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the CAN Transfer Protocol module.		

Function Name:	CAN_TP		
Arguments	Inputs	TickCounter	uint32
		Description: Counter of Timer channel ticks	
		DoorStatus	uint8
		Description: Value of DIO Channel 0	
		SwitchStatus	uint8
		Description: Value of DIO Channel 1	
		SpeedValue	uint16
		Description: Value of DIO Channel 2	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Send status messages periodically through the CAN protocol depending on required timing constraints.		

Function Name:	TimerTickCount		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Counts the timer ISR ticks.		

ECU 2:

Port Driver:

Function Name:	Port_Init		
Arguments	Inputs	ConfigPtr	const Port_ConfigType*
		Pointer to configuration set.	
		-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the Port Driver module.		

Type definitions:

Name:	Port_ConfigType
Type:	Structure
Range:	The contents of the initialization data structure are specific to the microcontroller.
Description:	Type of the external data structure containing the initialization data for this module.

Name:	Port_PinType
Type:	uint
Range:	Shall cover all available port pins. The type should be chosen for the specific MCU platform (best performance).
Description:	Data type for the symbolic name of a port pin.

Name:	Port_PinDirectionType
Type:	Enumeration
Range:	PORT_PIN_IN: Sets port pin as input. PORT_PIN_OUT: Sets port pin as output.
Description:	Possible directions of a port pin.

Name:	Port_PinModeType
Type:	Enumeration
Range:	As several port pin modes shall be configurable on one pin, the range shall be determined by the implementation
Description:	Different port pin modes.

DIO Driver:

Function Name:	Dio_Init		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the DIO Driver module.		

Function Name:	Dio_ReadChannel		
Arguments	Inputs	ChannelId	Dio_ChannelType
		Description: ID of DIO Channel	
		-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	Dio_LevelType	STD_HIGH	The physical level of the corresponding Pin is STD_HIGH
		STD_LOW	The physical level of the corresponding Pin is STD_LOW
Description:	Returns the value of the specified DIO channel.		

Function Name:	Dio_WriteChannel		
Arguments	Inputs	ChannelId	Dio_ChannelType
		Description: ID of DIO Channel	
		Level	Dio_LevelType
		Description: Value to be written	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Service to set a level of a channel.		

Type definitions:

Name:	Dio_ChannelType
Type:	uint
Range:	Shall cover all available DIO channels
Description:	Numeric ID of a DIO channel.

Name:	Dio_LevelType
Type:	uint8
Range:	STD_LOW: 0x00 | Physical state 0V STD_HIGH: 0x01 | Physical state 3.3V or 5V
Description:	These are the possible levels a DIO channel can have (input or output)

CAN Driver:

Function Name:	CAN_Init		
Arguments	Inputs	Config	Can_ConfigType*
		Description: Pointer to driver configuration.	
		-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	This function initializes the CAN module.		

Function Name:	Can_SetBaudrate		
Arguments	Inputs	Controller	uint8
		Description: CAN controller, whose baud rate shall be set	
		BaudRateConfigID	uint16
		Description: references a baud rate configuration by ID	
	Output	None	None
	Input/Output	None	None
Return	E_OK	Service request accepted, setting of (new) baud rate started	
	E_NOK	Service request not accepted	
Description:	This service shall set the baud rate configuration of the CAN controller.Depending on necessary baud rate modifications the controller might have to reset		

Function Name:	CAN_Write		
Arguments	Inputs	Data	uint64
		Description: Data to be sent over CAN bus	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	Passes a CAN message to CanDrv for transmission depending on input states.		

Function Name:	CAN_Read		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	uint8*		
Description:	Reads if a CAN message is sent on CanDrv for receive and returns its data.		

Function Name:	CANIntHandler		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Reads the CAN interrupt controller status and raises a flag for the received message.		

Function Name:	CAN_SetCallBack		
Arguments	Inputs	Ptr2Func	void
		Description: Address of Call Back Function	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	takes a pointer to function for upper layers function with the concept of call back.		

Type definitions:

Name:	Can_ConfigType
Type:	Structure
Range:	
Description:	This is the type of the external data structure containing the overall initialization data for the CAN driver and SFR settings affecting all controllers. Furthermore it contains pointers to controller configuration structures. The contents of the initialization data structure are CAN hardware specific.

GPT Driver:

Function Name:	Gpt_Init		
Arguments	Inputs	ConfigPtr	Gpt_ConfigType*
		Description: Pointer to a selected configuration structure	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the GPT driver.		

Function Name:	Gpt_StartTimer		
Arguments	Inputs	Channel	Dio_ChannelType
		Description: Numeric identifier of the GPT channel.	
		Value	Dio_LevelType
		Description: Target time in number of ticks.	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Starts a timer channel.		

Function Name:	Gpt_StopTimer		
Arguments	Inputs	Channel	Gpt_ChannelType
		Description: Numeric identifier of the GPT channel.	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Stops a timer channel.		

Function Name:	TimerIntHandler		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Checks the timer call back pointer and calls function in upper layer.		

Function Name:	Timer_SetCallBack		
Arguments	Inputs	Ptr2Func	void
		Description: Address of Call Back Function	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	takes a pointer to function for upper layers function with the concept of call back.		

Type definitions:

Name:	Gpt_ConfigType
Type:	Structure
Range:	
Description:	This is the type of the data structure including the configuration set required for initializing the GPT timer unit.

Name:	Gpt_ChannelType
Type:	uint
Range:	Shall cover all available Gpt channels.
Description:	Numeric ID of a GPT channel.

Name:	Gpt_LevelType
Type:	uint8
Range:	Microcontroller dependent.
Description:	Type for reading and setting the timer values (in number of ticks)

Right Light Driver:

Function Name:	RL_Init		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the car right lights module.		

Function Name:	SetRightLight_ON		
Arguments	Inputs	LightD	uint8
		Description: Numberic ID of Light Channel	
	Output	None	None
	Input/Output	None	None
Return	None		
Description:	Writes on Right Light Channel to set it on.		

Function Name:	SetRightLight_OFF		
Arguments	Inputs	LightD	uint8
		Description: Numberic ID of Light Channel	
	Output	None	None
	Input/Output	None	None
Return	None		
Description:	Writes on Right Light Channel to set it off.		

Left Light Driver:

Function Name:	LL_Init		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the car left lights module.		

Function Name:	SetLeftLight_ON		
Arguments	Inputs	LightD	uint8
		Description: Numberic ID of Light Channel	
	Output	None	None
	Input/Output	None	None
Return	None		
Description:	Writes on Left Light Channel to set it on.		

Function Name:	SetLeftLight_OFF		
Arguments	Inputs	LightD	uint8
		Description: Numberic ID of Light Channel	
	Output	None	None
	Input/Output	None	None
Return	None		
Description:	Writes on Left Light Channel to set it off.		

Buzzer Driver:

Function Name:	Buzzer_Init		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Initializes the Buzzer module.		

Function Name:	SetBuzzer_ON		
Arguments	Inputs	BuzzerID	uint8
		Description: Numberic ID of Buzzer Channel	
	Output	None	None
	Input/Output	None	None
Return	None		
Description:	Writes on Buzzer Channel to set it on.		

Function Name:	SetBuzzer_OFF		
Arguments	Inputs	BuzzerID	uint8
		Description: Numberic ID of Buzzer Channel	
	Output	None	None
	Input/Output	None	None
Return	None		
Description:	Writes on Buzzer Channel to set it off.		

Automotive Door Control Software Component:

Function Name:	Door_Control_Init		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	Initializes the Door Control module.		

Function Name:	Door_Control		
Arguments	Inputs	TickCounter	uint32
		Description: Counter of Timer channel ticks	
		DoorState	uint8
		Description: Value of door status message.	
		SwitchState	uint8
		Description: Value of switch status message.	
		SpeedValue	uint32
		Description: Value of speed message.	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Receives state messages sent through CAN protocol and controls buzzer and lights based on these states.		

Function Name:	DoorMessage_Store		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	Stores the door status messages sent over CAN bus.		

Function Name:	SwitchMessage_Store		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	Stores the switch status messages sent over CAN bus.		

Function Name:	SpeedMessage_Store		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	0	
	E_NOK	1	
Description:	Stores the speed value messages sent over CAN bus.		

Function Name:	TimerTickCount		
Arguments	Inputs	-	-
		-	
	Output	None	None
	Input/Output	None	None
Return	E_OK	None	
	E_NOK	None	
Description:	Counts the timer ISR ticks.		