Detailed API Description											
ECU name	**SWC name**	**API Description**	**Scope**	**Type**	**Reenterency**	**Syncronization**	**Recursion**	**Input Parameters**	**Parameter Description**	**Return Parameters**	**Parameter Description**
ECU 1	**Data_Handler**	main()	Private	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	void	NA	void	NA
		Data_Handler_Init()	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	void	NA	void	NA
		Data_Update()	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	void	NA	void	NA
		Data_Send()	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	void	NA	void	NA
	DR_Stat	DR_Stat_Init	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	* ConfigPtr	Pointer to Driver Pre-Compiler configured data	void	NA
		DR_UpdateStat	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	DR_Id	Identify required Door Id	void	NA
		DR_GetStat	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	DR_Id	Identify required Door Id	DR_StatType	return Job Status (OK in cause or success, NOK in case of Failure)
	SW_Stat	SW_Stat_Init	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	* ConfigPtr	Pointer to Driver Pre-Compiler configured data	void	NA
		SW_UpdateStat	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	SW_Id	Identify required SW Id	void	NA
		SW_GetStat	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	SW_Id	Identify required SW Id	SW_StatType	return Job Status (OK in cause or success, NOK in case of Failure)
	SPD_Stat	SPD_Stat_Init	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	* ConfigPtr	Pointer to Driver Pre-Compiler configured data	void	NA
		SPD_UpdateStat	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	SPD_Id	Identify required SPD sensor Id	void	NA
		SPD_GetStat	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	SPD_Id	Identify required SPD sensor Id	SPD_StatType	return Job Status (OK in cause or success, NOK in case of Failure)
		SPD_TimerHandler	public	**ISR**	Non-reentrant	**Syncronus**	Non-Recursion	void	NA	void	NA
		SPD_TickHandler	public	**ISR**	Non-reentrant	**Syncronus**	Non-Recursion	void	NA	void	NA
	TMR	TMR_Init	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	* ConfigPtr	Pointer to Driver Pre-Compiler configured data	void	NA
		TMR_StartTimer	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	channelId	Identify controlled TMR Id	void	NA
								TimerVale	Define Reload timer value before start		
		TMR_StopTimer	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	channelId	Identify controlled TMR Id	void	NA
ECU 2	**SW_StateHandler**	main	Private	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	void	NA	void	NA
		SW_StateHandler_Initt	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	void	NA	void	NA
		SW_StateHandler_StM	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	void	NA	void	NA
	LED_Ctrl	LED_Ctrl_Init	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	* ConfigPtr	Pointer to Driver Pre-Compiler configured data	void	NA
		LED_StatUpdate	public	**Function**	Non-reentrant	Syncronus	Non-Recursion	LED_Id	Identify controlled LED Id	void	NA
								state	Define required LED State	void	NA
	BZR_Ctrl	BZR_Stat_Init	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	* ConfigPtr	Pointer to Driver Pre-Compiler configured data	void	NA
		BZR_UpdateStat	public	**Function**	Non-reentrant	Syncronus	Non-Recursion	BZR_Id	Identify controlled BZR Id	void	NA
								BZR_Stat	Define required BZR State to be setted		
	PWM	PWM_Init	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	* ConfigPtr	Pointer to Driver Pre-Compiler configured data	void	NA
		PWM_Start	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	channelId	Identify controlled PWM channel Id	void	NA
		PWM_Stop	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	channelId	Identify controlled PWM channel Id	void	NA
	BCM_Handler	BCM_Handler_Init	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	void	NA	void	NA
		BCMH_Get_SW_State	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	*state	pointer to current SW State	Std_Type	return Job Status (OK in cause or success, NOK in case of Failure)
		BCMH_Get_DR_State	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	*state	pointer to current Door State	Std_Type	return Job Status (OK in cause or success, NOK in case of Failure)
		BCMH_Get_SPD_State	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	*state	pointer to current SPD State	Std_Type	return Job Status (OK in cause or success, NOK in case of Failure)
		BCMH_Update_SW_State	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	void	NA	Std_Type	return Job Status (OK in cause or success, NOK in case of Failure)
		BCMH_Update_DR_State	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	void	NA	Std_Type	return Job Status (OK in cause or success, NOK in case of Failure)
		BCMH_Update_SPD_State	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	void	NA	Std_Type	return Job Status (OK in cause or success, NOK in case of Failure)
	BCM	BCM_Init	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	* ConfigPtr	Pointer to Module Pre-Compiler configured data	void	NA
		BCM_Send	public	Function	Non-reentrant	**Syncronus**	Non-Recursion	* msgId	Pointer to Message ID to be sent	StdType	return Job Status (OK in cause or success, NOK in case of Failure)
								*HandleId	Message to Handelr moduile Id		
								*data	Pointer to data to be sent		
								* msgId	Pointer to Message ID to be received		return Job Status (OK in cause

Detailed API Description											
ECU name	**SWC name**	**API Description**	**Scope**	**Type**	**Reenterency**	**Syncronization**	**Recursion**	**Input Parameters**	**Parameter Description**	**Return Parameters**	**Parameter Description**
Common Components		BCM_Receive	public	Function	Reentrant	**Syncronus**	Non-Recursion	*HandleId	Message to Handelr upper Application moduile Id	StdType	return Job Status (OK in cause or success, NOK in case of Failure)
								*data	Pointer to Data to be received		
	CAN_Transv	CANTransv_Init	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	* ConfigPtr	Pointer to Driver Pre-Compiler configured data	void	NA
		CANTransv_Send	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	* data	Pointer to data to be sent	StdType	return Job Status (OK in cause or success, NOK in case of Failure)
		CANTransv_Receive	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	* data	Pointer to Data to be received	StdType	return Job Status (OK in cause or success, NOK in case of Failure)
	PORT	PORT_Init	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	* ConfigPtr	Pointer to Driver Pre-Compiler configured data	void	NA
	DIO	DIO_Init	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	* ConfigPtr	Pointer to Driver Pre-Compiler configured data	void	NA
		Dio_ReadChannel	public	**Function**	Reentrant	**Syncronus**	Non-Recursion	channelId	Identify controlled DIO channel Id	StdType	return Job Status (OK in cause or success, NOK in case of Failure)
		Dio_WriteChannel	public	**Function**	Non-reentrant	**Syncronus**	Non-Recursion	channelId	Identify controlled DIO channel Id	StdType	return Job Status (OK in cause or success, NOK in case of Failure)
								State	define intended write DIO State		