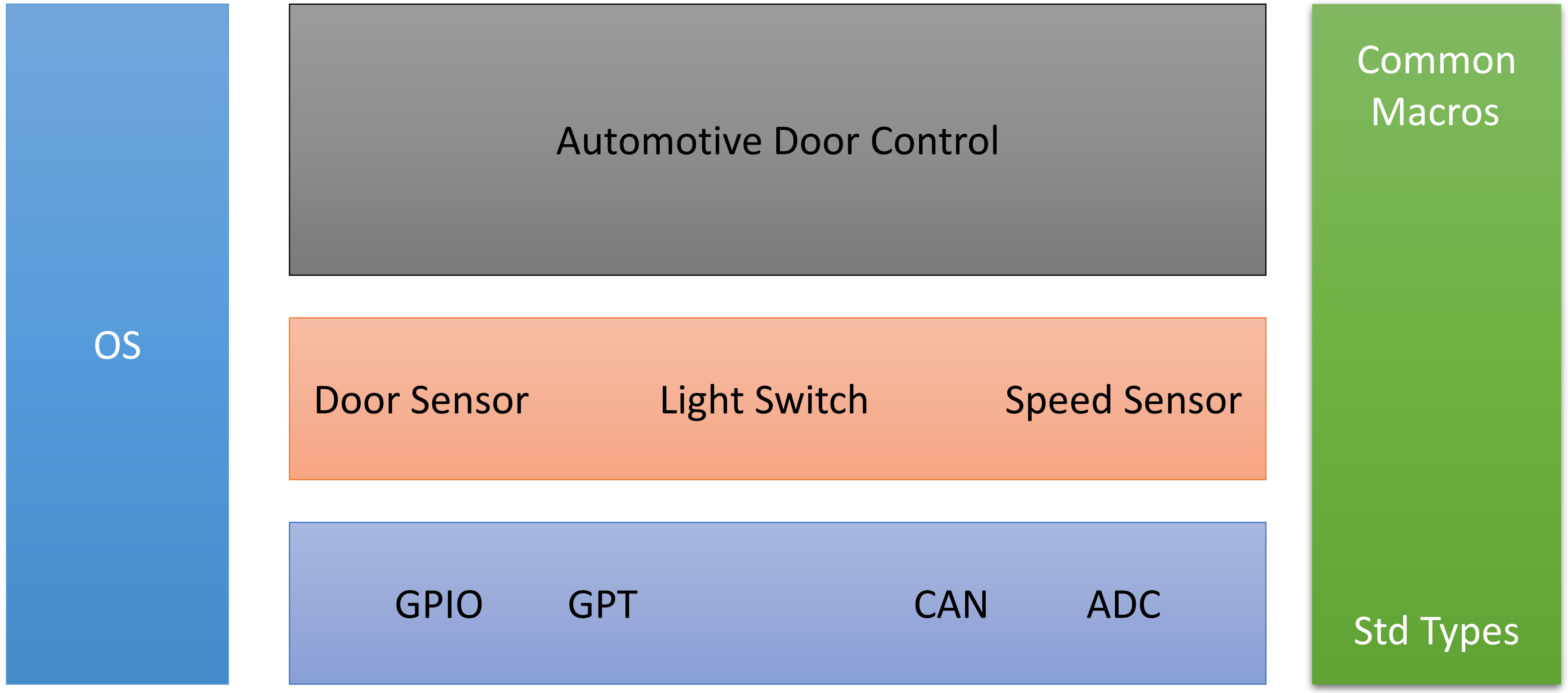
ECU 1


OS
Automotive Door Control
Door Sensor
Light Switch
Speed Sensor
GPIO
GPT
CAN
ADC
Common Macros
Std Types

GPIO

API - Types
Gpio_ChannelType
Gpio_PortIDType
Gpio_LevelType
Gpio_PortLevelType

API - Functions
void GPIO_Init(const Gpio_ConfigType * ConfigPtr)
Gpio_LevelType GPIO_ReadChannel(Gpio_ChannelType ChannelID)
void GPIO_WriteChannel(Gpio_ChannelType ChannelID, Gpio_LevelType Level)
Gpio_LevelType GPIO_FlipChannel(Gpio_ChannelType ChannelID)
Gpio_PortLevelType GPIO_ReadPort(Gpio_PortIDType PortID)
void GPIO_WritePort(Gpio_PortType PortID, Gpio_PortLevelType Level)

Configurations

- PortPinMode
- PortPinLevelValue
- PortPinDirection
- PortPinInternalAttach

Name	Channel ID	
Type	uint8	
Range		
Description	Numeric ID of Dio Pins.	

Name	Gpio_LevelType	
Type	uint8	
Range	0	Physical state 0V
	1	Physical state 5V or 3.3V
Description	These are the possible levels a DIO channel can have (input or output).	

Name	Gpio_ConfigType
Type	Structure
Range	uint8
Description	This structure contains all post-build configurable parameters of the DIO driver. A pointer to this structure is passed to the DIO driver initialization function for configuration.

Name	Gpio_PortIDType	
Type	uint8	
Range		
Description	Numeric ID of Port Numbers.	

Name	Gpio_PortLevelType	
Type	uint8	
Range	0	Physical state 0V
	1	Physical state 5V or 3.3V
Description	These are the possible levels a Port can have (input or output).	

Function Name	GPIO_Init		
Arguments	INPUTS	* ConfigPtr	Gpio_ConfigType
		Pointer to post-build configuration data	
	Output	None	
	Input/Output	None	
Return	void		
Description	Initializes the GPIO module.		

Function Name	GPIO_ReadChannel		
Arguments	INPUTS	ChannelID	Gpio_ChannelType
		ID of DIO pin	
	Output	GPIO_LevelType	
	Input/Output	None	
Return	E_OK	1	
	E_NOT_OK	0	
Description	Returns the value of the specified DIO pin.		

Function Name	GPIO_WriteChannel		
Arguments	INPUTS	ChannelID	Gpio_ChannelType
		ID of DIO pin	
		Level	Gpio_LevelType
		Value to be written	
	Output	None	
	Input/Output	None	
Return	void		
Description	Set level of the specified DIO pin.		

Function Name	GPIO_ReadPort		
Arguments	INPUTS	PortID	Gpio_PortIDType
		ID of DIO port	
	Output	GPIO_PortLevelType	
	Input/Output	None	
Return	E_OK	1	
	E_NOT_OK	0	
Description	Returns the level of the specified DIO port.		

Function Name	GPIO_WritePort		
Arguments	INPUTS	PortID	Gpio_PortIDType
		ID of DIO port	
		Level	Gpio_LevelType
		Value to be written	
	Output	None	
	Input/Output	None	
Return	void		
Description	Set level of the specified port.		

CAN

API - Types
CAN_ConfigType “Structure”

API - Functions
void CAN_Init(const CAN_ConfigType* Config)
Std_ReturnType CAN_SetBaudrate(uint8 Controller, uint16 BaudRateConfigID)
void CAN_Write(uint32 Data)
uint32 CAN_Read(void)

Function Name	CAN_Init		
Arguments	INPUTS	* ConfigPtr	CAN_ConfigType
		Pointer to a selected configurations	
	Output	None	
	Input/Output	None	
Return	void		
Description	Initializes the CAN module.		

Function Name	CAN_SetBaudRate		
Arguments	INPUTS	Controller	uint8
		CAN controller whose baud rate will be set	
		BaudRate	uint16
		Requested baud rate in kbps	
	Output	None	
	Input/Output	None	
Return	E_OK	1	
	E_NOK	0	
Description	Set level of the specified port.		

Function Name	CAN_Write		
Arguments	INPUTS	Data	uint32
		Data to be sent	
	Output	None	
	Input/Output	None	
Return	void		
Description	Send data from CAN controller.		

Function Name	CAN_Read	
Arguments	INPUTS	None
	Output	uint32
	Input/Output	None
Return	uint32	
Description	Receive data from CAN controller.	

ADC

API - Types
ADC_ConfigType “Structure”
ADC_Prescalar
ADC_RefVolatge

API - Functions
void ADC_Init(const ADC_ConfigType * Config_Ptr)
uint32 ADC_readChannel(uint8 CH_num)

Function Name	ADC_Init		
Arguments	INPUTS	* ConfigPtr	ADC_ConfigType
		Pointer to a selected configurations	
	Output	None	
	Input/Output	None	
Return	void		
Description	Initializes the ADC module.		

Function Name	ADC_readChannel		
Arguments	INPUTS	Ch_Num	uint8
		ID of ADC channel	
	Output	uint32	
	Input/Output	None	
Return	uint32		
Description	Return value of the specified ADC channel		

GPT

API - Types
GPT_ConfigType “Structure”
GPT_ValueType uint8

API - Functions
void Timer_Init(const GPT_ConfigType * Config_Ptr)
void Timer_Start(GPT_ValueType Value)
void Timer_Stop(void)

Name	Timer_ValueType
Type	uint8
Range	The range of this type is µC dependent (width of the timer register) and has to be described by the supplier.
Description	Type for reading and setting the timer values (in number of ticks).

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

Function Name	Timer_Init		
Arguments	INPUTS	* ConfigPtr	Timer_ConfigType
		Pointer to a selected configurations	
	Output	None	
	Input/Output	None	
Return	void		
Description	Initializes the Timer module.		

Function Name	Timer_Start		
Arguments	INPUTS	Value	GPT_ValueType
		Target time in ticks	
	Output	None	
	Input/Output	None	
Return	void		
Description	Start the timer module.		

Function Name	Timer_Stop	
Arguments	INPUTS	None
	Output	None
	Input/Output	None
Return	void	
Description	Stop the timer module.	

Door Sensor

Must include the “GPIO driver”

API - Functions
void D_Init(void)
uint8 D_ReadValue(Gpio_ChannelType ChannelID)

Function Name	D_Init	
Arguments	INPUTS	None
	Output	None
	Input/Output	None
Return	void	
Description	Initializes the door sensor.	

Function Name	D_ReadValue	
Arguments	INPUTS	ChannelID
	Output	uint8
	Input/Output	None
Return	uint8	
Description	Get the state of the door sensor.	

Light Switch

Must include the “GPIO driver”

API - Functions
void L_Init(void)
uint8 L_GetState(Gpio_ChannelType ChannelID)

Function Name	L_Init	
Arguments	INPUTS	None
	Output	None
	Input/Output	None
Return	void	
Description	Initializes the Light switch.	

Function Name	L_GetState	
Arguments	INPUTS	ChannelID
	Output	uint8
	Input/Output	None
Return	uint8	
Description	Get the state of the light sensor.	

Speed Sensor

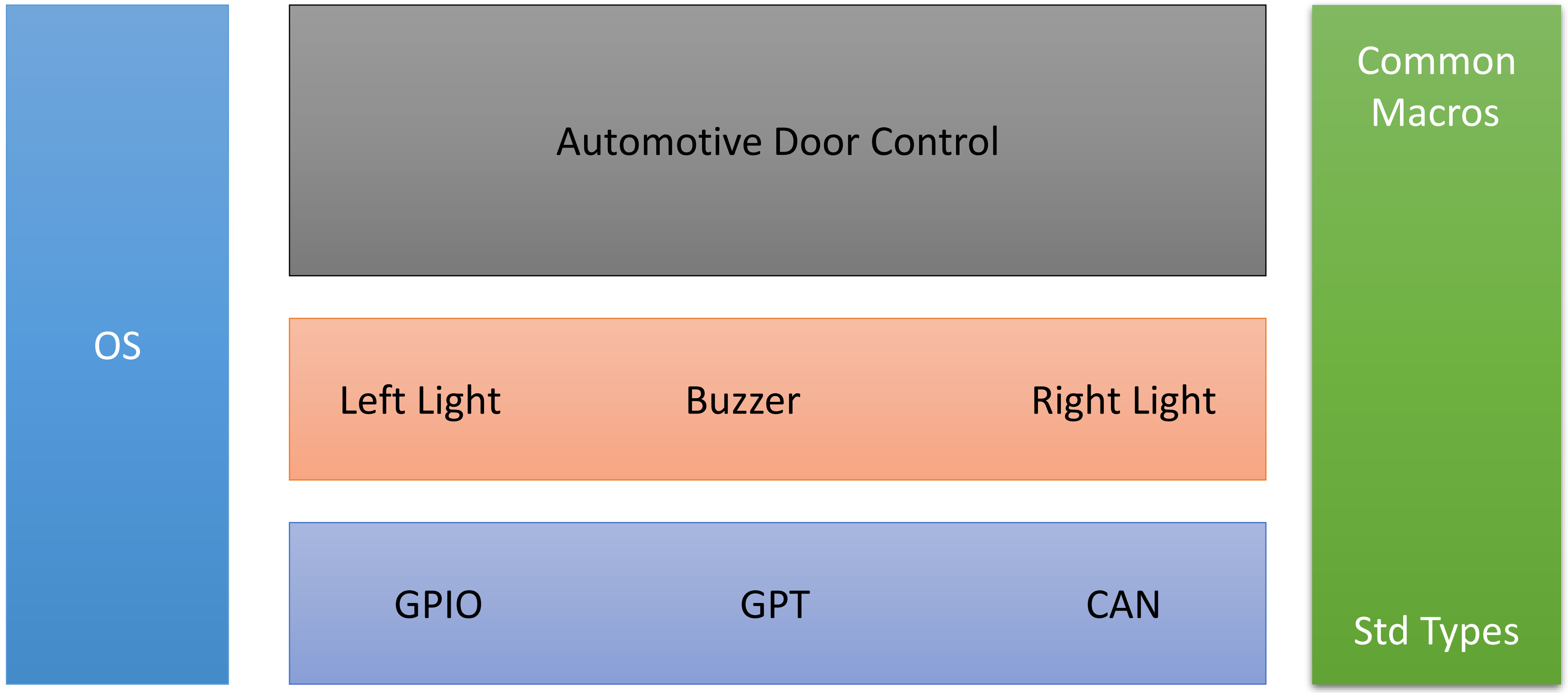
Must include the “ADC driver”

API - Functions
void S_Init(void)
uint8 S_ReadValue(uint8 ADC_Channel)

Function Name	S_Init	
Arguments	INPUTS	None
	Output	None
	Input/Output	None
Return	void	
Description	Initializes the Speed sensor.	

Function Name	S_ReadValue	
Arguments	INPUTS	ADC_Channel
	Output	uint8
	Input/Output	None
Return	uint8	
Description	Get the state of the Speed sensor.	

ECU 2


OS
Automotive Door Control
Left Light
Buzzer
Right Light
GPIO
GPT
CAN
Common Macros
Std Types

GPIO

API - Types
Gpio_ChannelType
Gpio_PortIDType
Gpio_LevelType
Gpio_PortLevelType

API - Functions
void GPIO_Init(const Gpio_ConfigType * ConfigPtr)
Gpio_LevelType GPIO_ReadChannel(Gpio_ChannelType ChannelID)
void GPIO_WriteChannel(Gpio_ChannelType ChannelID, Gpio_LevelType Level)
Gpio_LevelType GPIO_FlipChannel(Gpio_ChannelType ChannelID)
Gpio_PortLevelType GPIO_ReadPort(Gpio_PortIDType PortID)
void GPIO_WritePort(Gpio_PortType PortID, Gpio_PortLevelType Level)

Configurations

- PortPinMode
- PortPinLevelValue
- PortPinDirection
- PortPinInternalAttach

Name	Channel ID
Type	uint8
Range	
Description	Numeric ID of Dio Pins.

Name	Gpio_LevelType	
Type	uint8	
Range	0	Physical state 0V
	1	Physical state 5V or 3.3V
Description	These are the possible levels a DIO channel can have (input or output).	

Name	Gpio_ConfigType
Type	Structure
Range	uint8
Description	This structure contains all post-build configurable parameters of the DIO driver. A pointer to this structure is passed to the DIO driver initialization function for configuration.

Name	Gpio_PortIDType
Type	uint8
Range	
Description	Numeric ID of Port Numbers.

Name	Gpio_PortLevelType	
Type	uint8	
Range	0	Physical state 0V
	1	Physical state 5V or 3.3V
Description	These are the possible levels a Port can have (input or output).	

Function Name	GPIO_Init		
Arguments	INPUTS	* ConfigPtr	Gpio_ConfigType
		Pointer to post-build configuration data	
	Output	None	
	Input/Output	None	
Return	void		
Description	Initializes the GPIO module.		

Function Name	GPIO_ReadChannel		
Arguments	INPUTS	ChannelID	Gpio_ChannelType
		ID of DIO pin	
	Output	GPIO_LevelType	
	Input/Output	None	
Return	E_OK	1	
	E_NOT_OK	0	
Description	Returns the value of the specified DIO pin.		

Function Name	GPIO_WriteChannel		
Arguments	INPUTS	ChannelID	Gpio_ChannelType
		ID of DIO pin	
		Level	Gpio_LevelType
		Value to be written	
	Output	None	
	Input/Output	None	
Return	void		
Description	Set level of the specified DIO pin.		

Function Name	GPIO_ReadPort		
Arguments	INPUTS	PortID	Gpio_PortIDType
		ID of DIO port	
	Output	GPIO_PortLevelType	
	Input/Output	None	
Return	E_OK	1	
	E_NOT_OK	0	
Description	Returns the level of the specified DIO port.		

Function Name	GPIO_WritePort		
Arguments	INPUTS	PortID	Gpio_PortIDType
		ID of DIO port	
		Level	Gpio_LevelType
		Value to be written	
	Output	None	
	Input/Output	None	
Return	void		
Description	Set level of the specified port.		

CAN

API - Types
CAN_ConfigType “Structure”

API - Functions
void CAN_Init(const CAN_ConfigType* Config)
Std_ReturnType CAN_SetBaudrate(uint8 Controller, uint16 BaudRateConfigID)
void CAN_Write(uint32 Data)
uint32 CAN_Read(void)

Function Name	CAN_Init		
Arguments	INPUTS	* ConfigPtr	CAN_ConfigType
		Pointer to a selected configurations	
	Output	None	
	Input/Output	None	
Return	void		
Description	Initializes the CAN module.		

Function Name	CAN_SetBaudRate		
Arguments	INPUTS	Controller	uint8
		CAN controller whose baud rate will be set	
		BaudRate	uint16
		Requested baud rate in kbps	
	Output	None	
	Input/Output	None	
Return	E_OK	1	
	E_NOK	0	
Description	Set level of the specified port.		

Function Name	CAN_Write		
Arguments	INPUTS	Data	uint32
		Data to be sent	
	Output	None	
	Input/Output	None	
Return	void		
Description	Send data from CAN controller.		

Function Name	CAN_Read	
Arguments	INPUTS	None
	Output	uint32
	Input/Output	None
Return	uint32	
Description	Receive data from CAN controller.	

GPT

API - Types
GPT_ConfigType “Structure”
GPT_ValueType uint8

API - Functions
void Timer_Init(const GPT_ConfigType * Config_Ptr)
void Timer_Start(GPT_ValueType Value)
void Timer_Stop(void)

Name	Timer_ValueType
Type	uint8
Range	The range of this type is µC dependent (width of the timer register) and has to be described by the supplier.
Description	Type for reading and setting the timer values (in number of ticks).

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

Function Name	Timer_Init		
Arguments	INPUTS	* ConfigPtr	Timer_ConfigType
		Pointer to a selected configurations	
	Output	None	
	Input/Output	None	
Return	void		
Description	Initializes the Timer module.		

Function Name	Timer_Start		
Arguments	INPUTS	Value	GPT_ValueType
		Target time in ticks	
	Output	None	
	Input/Output	None	
Return	void		
Description	Start the timer module.		

Function Name	Timer_Stop	
Arguments	INPUTS	None
	Output	None
	Input/Output	None
Return	void	
Description	Stop the timer module.	

Left Light

Must include the “GPIO driver”

API - Functions
Void LL_Init(Gpio_ChannelType ChannelID)
void LL_ON()
void LL_OFF()

Function Name	LL_Init		
Arguments	INPUTS	ChannelID	Gpio_ChannelType
		ID of a given channel	
	Output	None	
	Input/Output	None	
Return	void		
Description	Initializes the Left Light.		

Function Name	LL_ON	
Arguments	INPUTS	None
	Output	None
	Input/Output	None
Return	void	
Description	Make Left Light ON.	

Function Name	LL_OFF	
Arguments	INPUTS	None
	Output	None
	Input/Output	None
Return	void	
Description	Make Left Light OFF.	

Buzzer

Must include the “GPIO driver”

API - Functions
Void B_Init(Gpio_ChannelType ChannelID)
void B_ON()
void B_OFF()

Function Name	B_Init		
Arguments	INPUTS	ChannelID	Gpio_ChannelType
		ID of a given channel	
	Output	None	
	Input/Output	None	
Return	void		
Description	Initializes the Buzzer.		

Function Name	B_ON	
Arguments	INPUTS	None
	Output	None
	Input/Output	None
Return	void	
Description	Make Buzzer ON.	

Function Name	B_OFF	
Arguments	INPUTS	None
	Output	None
	Input/Output	None
Return	void	
Description	Make Buzzer OFF.	

Right Light

Must include the “GPIO driver”

API - Functions
Void RL_Init(Gpio_ChannelType ChannelID)
void RL_ON()
void RL_OFF()

Function Name	RL_Init		
Arguments	INPUTS	ChannelID	Gpio_ChannelType
		ID of a given channel	
	Output	None	
	Input/Output	None	
Return	void		
Description	Initializes the Right Light.		

Function Name	RL_ON	
Arguments	INPUTS	None
	Output	None
	Input/Output	None
Return	void	
Description	Make Right Light ON.	

Function Name	RL_OFF	
Arguments	INPUTS	None
	Output	None
	Input/Output	None
Return	void	
Description	Make Right Light OFF.	