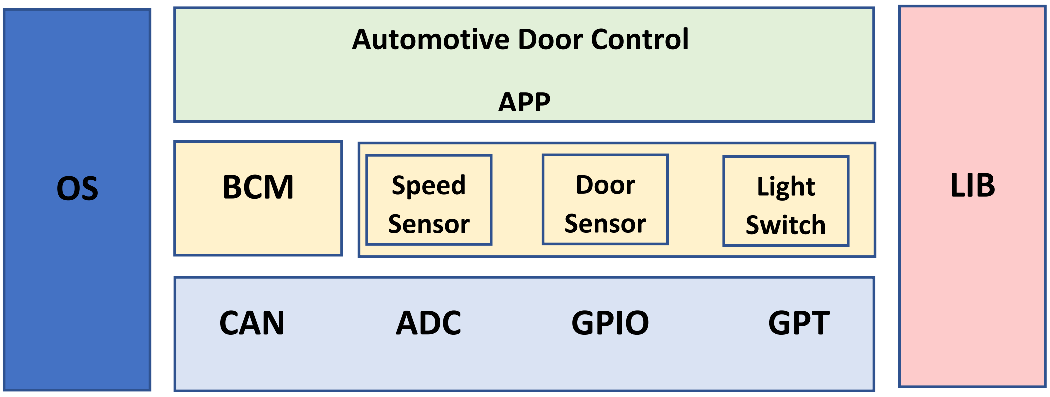
Automotive Door Control System Design

Static Design

ECU 1

Layered Architecture:



GPIO Module

API – TYPES

GPIO_PIN_TYPE
GPIO_PORT_TYPE
GPIO_PIN_LEVEL_TYPE
GPIO_PORT_LEVEL_TYPE
GPIO_CONFIG_TYPE

API – FUNCTIONS

GPIO_PIN_LEVEL_TYPE **GPIO_Read_Pin** (GPIO_PORT_TYPE **GPIO_port**, GPIO_PIN_TYPE **GPIO_pin**) ;
void **GPIO_Write_Pin** (GPIO_PORT_TYPE **GPIO_port**, GPIO_PIN_TYPE **GPIO_pin**, GPIO_PIN_LEVEL_TYPE **GPIO_pin_level**);
GPIO_PORT_LEVEL_TYPE **GPIO_Read_Port** (GPIO_PORT_TYPE **GPIO_port**) ;
void **GPIO_Write_port** (GPIO_PORT_TYPE **GPIO_port**, GPIO_port_level_TYPE **GPIO_port_level**) ;
GPIO_PIN_LEVEL_TYPE **GPIO_toggle_Pin** (GPIO_PORT_TYPE **GPIO_port**, GPIO_PIN_TYPE **GPIO_pin**);
void **GPIO_init** (const GPIO_CONFIG_TYPE * **GPIO_pin_config**) ;

API – Configuration

- Port_Pin_Mode
- Port_Pin_Direction
- Port_Pin_Level
- Port_Pin_out_current
- Port_Pin_Internal_Attach

Name	GPIO_PIN_TYPE
Type	U8
Range	0 - 15
Description	It defines which pin selected in the port.

Name	GPIO_PORT_TYPE
Type	U8
Range	0 - 5
Description	It defines which port selected.

Name	GPIO_PIN_LEVEL_TYPE	
Type	U8	
Range	0	Low Level
	1	High Level
Description	It defines which level state the pin have in both cases (In/Out).	

Name	GPIO_PORT_LEVEL_TYPE
Type	U16
Range	0 – 65,536
Description	It defines which level state the port have in both cases (In/Out).

Name	GPIO_CONFIG_TYPE
Type	Struct
Range	
Description	It contains all the configuration of the GPIO Module. It is used in the GPIO_Init Function to set the GPIO Pins.

Function Name	GPIO_Read_Pin		
Arguments	Inputs	GPIO_port	GPIO_PORT_TYPE
		GPIO_pin	GPIO_PIN_TYPE
	outputs		
	In / out		
Return	pin_state	GPIO_PIN_LEVEL_TYPE	
Description	This function is used to get the current voltage level of specified pin		

Function Name	GPIO_Read_Port		
Arguments	Inputs	GPIO_port	GPIO_PORT_TYPE
	outputs		
	In / out		
Return	port_state	GPIO_PORT_LEVEL_TYPE	
Description	This function is used to get the current voltage level of specified pin		

Function Name	GPIO_Write_Pin		
Arguments	Inputs	GPIO_port	GPIO_PORT_TYPE
		GPIO_pin	GPIO_PIN_TYPE
		GPIO_pin_level	GPIO_PIN_LEVEL_TYPE
	outputs		
	In / out		
Return	void		
Description	This function is used to set the voltage level of specified pin .		

Function Name	GPIO_Write_port		
Arguments	Inputs	GPIO_port	GPIO_PORT_TYPE
		GPIO_port_level	GPIO_PORT_LEVEL_TYPE
	outputs		
	In / out		
Return	void		
Description	This function is used to set the voltage levels of pins of specified port .		

Function Name	GPIO_init		
Arguments	Inputs	GPIO_pin_config	* GPIO_CONFIG_TYPE
	outputs		
	In / out		
Return	void		
Description	This function is used to GPIO pins and ports .		

CAN Module

API – TYPES

CAN_CONFIG_TYPE

API – FUNCTIONS

void **CAN_Write** (u32 data);
u32 **CAN_Read** (void);
void **CAN_Init** (const CAN_CONFIG_TYPE *** CAN _config**);

Name	CAN_CONFIG_TYPE
Type	Struct
Range	
Description	It contains all the configuration of the CAN Module. It is used in the CAN_Init Function.

Function Name	CAN_Write		
Arguments	Inputs	data	U32
	outputs		
	In / out		
Return	Void		
Description	This function is used write data on the CAN Bus .		

Function Name	CAN_Read		
Arguments	Inputs	Void	
	outputs		
	In / out		
Return	U32		
Description	This function is used to read data from CAN Bus .		

Function Name	CAN_Init		
Arguments	Inputs	CAN _config	* CAN_CONFIG_TYPE
	Outputs		
	In / out		
Return	Void		
Description	This function is used initialize the CAN hardware.		

ADC Module

API – TYPES

ADC_CONFIG_TYPE
ADC_CHUNNEL_TYPE

API – FUNCTIONS

u32 **ADC_ReadChunnel** (ADC_CHUNNEL_TYPE **ADC_Chunnel**);
void **ADC_Init** (const ADC_CONFIG_TYPE ***** **ADC_config**);

Name	ADC_CONFIG_TYPE
Type	Struct
Range	
Description	It contains all the configuration of the ADC Module. It is used in the ADC_Init Function.

Name	ADC_CHUNNEL_TYPE
Type	U8
Range	0 – 10
Description	It defines which Chunnel is used in ADC module .

Function Name	ADC_ReadChunnel		
Arguments	Inputs	ADC_Chunnel	ADC_CHUNNEL_TYPE
	outputs		
	In / out		
Return	U32		
Description	This function is used to read the converted data from ADC Chunnel .		

Function Name	ADC_Init		
Arguments	Inputs	ADC_Config	* ADC_CONFIG_TYPE
	Outputs		
	In / out		
Return	Void		
Description	This function is used initialize the ADC hardware.		

GPT Module

API – TYPES

GPT_CONFIG_TYPE
GPT_LoadValue_TYPE

API – FUNCTIONS

void **GPT_Start** (GPT_LoadValue_TYPE **Value**);
void **GPT_Stop** (void);
void **GPT_Init** (const GPT_CONFIG_TYPE * **GPT_config**);

Name	GPT_CONFIG_TYPE
Type	Struct
Range	
Description	It contains all the configuration of the GPT TIMER Module. It is used in the GPT_Init Function.

Name	GPT_LoadValue_TYPE
Type	U32
Range	0 – 65635
Description	It defines the value loaded into the timer counter .

Function Name	GPT_Start		
Arguments	Inputs	Value	GPT_LoadValue_TYPE
	Outputs		
	In / out		
Return	Void		
Description	This function is used start counting of the GPT Timer .		

Function Name	GPT_Stop	
Arguments	Inputs	Void
	Outputs	
	In / out	
Return	Void	
Description	This function is used stop counting of the GPT Timer .	

Function Name	GPT_Init		
Arguments	Inputs	GPT_Config	* GPT_CONFIG_TYPE
	Outputs		
Return	Void		
Description	This function is used initialize the GPT Timer.		

Door Sensor

Include: GPIO Driver

API – FUNCTIONS

void **Door_GetStatus** (GPIO_PORT_TYPE **port,** GPIO_PIN_TYPE **pin**);

void **Door_Init** (void);

Function Name	Door_GetStatus		
Arguments	Inputs	Port	GPIO_PORT _TYPE
		Pin	GPIO_PIN_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used to get the current state of the door sensor.		

Function Name	Door_Init	
Arguments	Inputs	Void
	Outputs	
Return	Void	
Description	This function is used initialize the Door Sensor hardware.	

Light Switch Module

Include: GPIO Driver

API – FUNCTIONS

void **Light_GetStatus** (GPIO_PORT_TYPE **port,** GPIO_PIN_TYPE **pin**);

void **Light_Init** (void);

Function Name	Light_GetStatus		
Arguments	Inputs	Port	GPIO_PORT _TYPE
		Pin	GPIO_PIN_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used to get the current state of the light switch.		

Function Name	Light_Init	
Arguments	Inputs	Void
	Outputs	
Return	Void	
Description	This function is used initialize the Light switch hardware.	

Speed Sensor

Include: ADC Driver

API – FUNCTIONS

void **Speed_GetValue** (ADC_CHUNNEL_TYPE **Adc_Chunnel**);

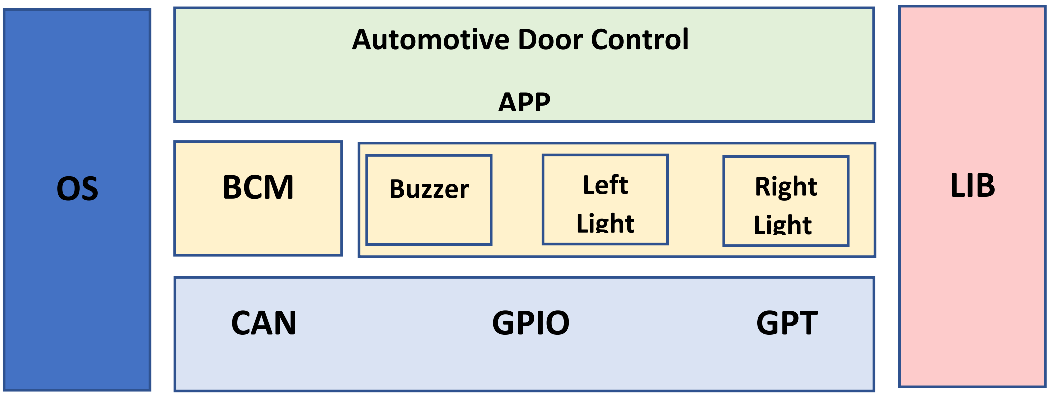
void **Speed_Init** (void);

Function Name	Speed_GetValue		
Arguments	Inputs	Adc_Chunnel	ADC_CHUNNEL_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used to monitor the speed value .		

Function Name	Speed_Init	
Arguments	Inputs	Void
	Outputs	
Return	Void	
Description	This function is used initialize the speed Sensor hardware.	

ECU 2

Layered Architecture:



GPIO Module

API – TYPES

GPIO_PIN_TYPE
GPIO_PORT_TYPE
GPIO_PIN_LEVEL_TYPE
GPIO_PORT_LEVEL_TYPE
GPIO_CONFIG_TYPE

API – FUNCTIONS

GPIO_PIN_LEVEL_TYPE **GPIO_Read_Pin** (GPIO_PORT_TYPE **GPIO_port**, GPIO_PIN_TYPE **GPIO_pin**) ;
void **GPIO_Write_Pin** (GPIO_PORT_TYPE **GPIO_port**, GPIO_PIN_TYPE **GPIO_pin**, GPIO_PIN_LEVEL_TYPE **GPIO_pin_level**);
GPIO_PORT_LEVEL_TYPE **GPIO_Read_Port** (GPIO_PORT_TYPE **GPIO_port**) ;
void **GPIO_Write_port** (GPIO_PORT_TYPE **GPIO_port**, GPIO_port_level_TYPE **GPIO_port_level**) ;
GPIO_PIN_LEVEL_TYPE **GPIO_toggle_Pin** (GPIO_PORT_TYPE **GPIO_port**, GPIO_PIN_TYPE **GPIO_pin**);
void **GPIO_init** (const GPIO_CONFIG_TYPE *** GPIO_pin_config**) ;

API – Configuration

- Port_Pin_Mode
- Port_Pin_Direction

- Port_Pin_Level
- Port_Pin_out_current
- Port_Pin_Internal_Attach

Name	GPIO_PIN_TYPE
Type	U8
Range	0 – 15
Description	It defines which pin selected in the port.

Name	GPIO_PORT_TYPE
Type	U8
Range	0 – 5
Description	It defines which port selected.

Name	GPIO_PIN_LEVEL_TYPE	
Type	U8	
Range	0	Low Level
	1	High Level
Description	It defines which level state the pin have in both cases (In/Out).	

Name	GPIO_PORT_LEVEL_TYPE
Type	U16
Range	0 – 65,536
Description	It defines which level state the port have in both cases (In/Out).

Name	GPIO_CONFIG_TYPE
Type	Struct
Range	
Description	It contains all the configuration of the GPIO Module. It is used in the GPIO_Init Function to set the GPIO Pins.

Function Name	GPIO_Read_Pin		
Arguments	Inputs	GPIO_port	GPIO_PORT_TYPE
		GPIO_pin	GPIO_PIN_TYPE
	outputs		

	In / out	
Return	pin_state	GPIO_PIN_LEVEL_TYPE
Description	This function is used to get the current voltage level of specified pin	

Function Name	GPIO_Read_Port		
Arguments	Inputs	GPIO_port	GPIO_PORT_TYPE
	Outputs		
	In / out		
Return	port_state	GPIO_PORT_LEVEL_TYPE	
Description	This function is used to get the current voltage level of specified pin		

Function Name	GPIO_Write_Pin		
Arguments	Inputs	GPIO_port	GPIO_PORT_TYPE
		GPIO_pin	GPIO_PIN_TYPE
		GPIO_pin_level	GPIO_PIN_LEVEL_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used to set the voltage level of specified pin .		

Function Name	GPIO_Write_port		
Arguments	Inputs	GPIO_port	GPIO_PORT_TYPE
		GPIO_port_level	GPIO_PORT_LEVEL_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used to set the voltage levels of pins of specified port .		

Function Name	GPIO_init		
Arguments	Inputs	GPIO_pin_config	* GPIO_CONFIG_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used to GPIO pins and ports .		

CAN Module

API – TYPES

CAN_CONFIG_TYPE

API – FUNCTIONS

void **CAN_Write** (u32 data);

u32 **CAN_Read** (void);

void **CAN_Init** (const CAN_CONFIG_TYPE * **CAN _config**);

Name	CAN_CONFIG_TYPE
Type	Struct
Range	
Description	It contains all the configuration of the CAN Module. It is used in the CAN_Init Function.

Function Name	CAN_Write		
Arguments	Inputs	Data	U32
	outputs		
	In / out		
Return	Void		
Description	This function is used write data on the CAN Bus .		

Function Name	CAN_Read	
Arguments	Inputs	Void
	outputs	
	In / out	
Return	U32	
Description	This function is used to read data from CAN Bus .	

Function Name	CAN_Init		
Arguments	Inputs	CAN _config	* CAN_CONFIG_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used initialize the CAN hardware.		

GPT Module

API – TYPES

GPT_CONFIG_TYPE
GPT_LoadValue_TYPE

API – FUNCTIONS

void **GPT_Start** (GPT_LoadValue_TYPE **Value**);
void **GPT_Stop** (void);
void **GPT_Init** (const GPT_CONFIG_TYPE *** GPT_config**);

Name	GPT_CONFIG_TYPE
Type	Struct
Range	
Description	It contains all the configuration of the GPT TIMER Module. It is used in the GPT_Init Function.

Name	GPT_LoadValue_TYPE
Type	U32
Range	0 – 65635
Description	It defines the value loaded into the timer counter .

Function Name	GPT_Start		
Arguments	Inputs	Value	GPT_LoadValue_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used start counting of the GPT Timer .		

Function Name	GPT_Stop	
Arguments	Inputs	Void
	outputs	
	In / out	
Return	Void	
Description	This function is used stop counting of the GPT Timer .	

Function Name	GPT_Init		
Arguments	Inputs	GPT_Config	* GPT_CONFIG_TYPE
	Outputs		
Return	Void		
Description	This function is used initialize the GPT Timer.		

Light Right Module

Include: GPIO Driver

API – FUNCTIONS

void **Light_Right_TurnON** (GPIO_PORT_TYPE **port,** GPIO_PIN_TYPE **pin**);

void **Light_Right_TurnOFF** (GPIO_PORT_TYPE **port,** GPIO_PIN_TYPE **pin**);

Function Name	Light_Right_TurnON		
Arguments	Inputs	Port	GPIO_PORT _TYPE
		Pin	GPIO_PIN_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used turn on the right light .		

Function Name	Light_Right_TurnOFF		
Arguments	Inputs	Port	GPIO_PORT _TYPE
		Pin	GPIO_PIN_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used turn off the right light .		

Light Left Module

Include: GPIO Driver

API – FUNCTIONS

void **Light_Left_TurnON** (GPIO_PORT_TYPE **port,** GPIO_PIN_TYPE **pin**);

void **Light_Left_TurnOFF** (GPIO_PORT_TYPE **port,** GPIO_PIN_TYPE **pin**);

Function Name	Light_Left_TurnON		
Arguments	Inputs	Port	GPIO_PORT _TYPE
		Pin	GPIO_PIN_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used turn on the Left light .		

Function Name	Light_Left_TurnOFF		
Arguments	Inputs	Port	GPIO_PORT _TYPE
		Pin	GPIO_PIN_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used turn off the Left light .		

Buzzer Module

Include: GPIO Driver

API – FUNCTIONS

void **Buzzer_TurnON** (GPIO_PORT_TYPE **port,** GPIO_PIN_TYPE **pin**);

void **Buzzer_TurnOFF** (GPIO_PORT_TYPE **port,** GPIO_PIN_TYPE **pin**);

Function Name	Buzzer_TurnON		
Arguments	Inputs	Port	GPIO_PORT _TYPE
		Pin	GPIO_PIN_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used turn on the Buzzer.		

Function Name	Buzzer_TurnOFF		
Arguments	Inputs	Port	GPIO_PORT _TYPE
		Pin	GPIO_PIN_TYPE
	outputs		
	In / out		
Return	Void		
Description	This function is used turn off the Buzzer.		