Modulename	SWlayer		Dependancies				
DIO DRIVERS	ECU HW drivers						
ADC DRIVERS	ECU HW drivers		GPIO DRIVERS				
CAN DRIVERS	ECU HW drivers		GPIO DRIVERS				
GPT DRIVERS	ECU HW drivers						
switchdriver	ONBOARD layer		GPIO DRIVERS				
doorsensordriver	ONBOARD layer		GPIO DRIVERS				
speed sensor	ONBOARD layer		GPIO DRIVERS	GPT DRIVERS	ADC DRIVERS		
comm handler	ONBOARD layer		CAN DRIVERS				
Comm. manger	Service layer		Comm. Handler				
APP	APP layer		switchdriver	doorsensordriver	speed sensor driver		comm manager
OS	system layer		ECU HW drivers	APP			
DIO API							
name	DIO_PIN_t						
type	Enumeration						
range	0:rangof pins						
description	enum listing pins						
name	DIO_PORT_t						
type	Enumeration						
range	0:rangof Ports						
description	enum listing ports						
name	DIO_PIN_DIR_t						
type	Enumeration						
range	PIN_INPUT	0					
	PIN_OUTPUT	1					
description	enum listing pins DIR						
name	DIO_PORT_DIR_t						
type	uint32						
range	0:all pins ar 1						
description	ports DIR						
name	DIO_PORT_LEVEL_t						
type	uint32						
range	0:all pins ar 1						
description	PORT LEVELS						
name	DIO_PIN_LEVEL_t						
type	Enumeration						
range	PIN_LOW	0	PIN IS LOW				
	PIN_HIGH	1	PIN IS HIGH				
description	enum listing PIN LEVELS						
name	PIN_ALTR_FUNC_t						
type	Enumeration						
range							
description	enum listing PIN alternative functions						
function name	DIO_Init						
arguments	NONE						
return	NONE						
description	init GPIO pins and their altrnative funct. from config file						
function name	DIO_SET_PORT						
arguments	input	PORTNUM	DIO_PORT_t				
		the port needed to set					
		PORT_LEVEL	DIO_PORT_LEVEL_t				
		port LEVEL					

return	NONE						
description	set port value						
function name	DIO_SET_PIN						
arguments	input	PIN_NUM	DIO_PIN_t				
		the pin u want to set as an input					
		PORTNUM	DIO_PORT_t				
		the port of that pin					
		PIN_LEVEL	DIO_PIN_LEVEL_T				
		PIN LEVEL					
return	NONE						
description	set pin value						
function name	DIO_TOG_PIN						
arguments	input	PIN_NUM	DIO_PIN_t				
		the pin u want to set as an input					
		PORTNUM	DIO_PORT_t				
		the port of that pin					
return	NONE						
description	Toggle pin						
function name	DIO_TOG_PORT						
arguments	input	PORTNUM	DIO_PORT_t				
		the port of that pin					
return	NONE						
description	Toggle port						
function name	DIO_SET_PIN_DIR						
arguments	input	PIN_NUM	DIO_PIN_t				
		the pin u want to set as an input					
		PORTNUM	DIO_PORT_t				
		the port of that pin					
		PIN_DIR	DIO_PIN_DIR_T				
		pin direction					
return	NONE						
description	set pin dir						
function name	DIO_SET_PORT_DIR						
arguments	input	PORTNUM	DIO_PORT_t				
		the port					
		PORT_DIR	DIO_PORT_DIR_T				
		PORT direction					
return	NONE						
description	set port dir						
function name	DIO_SET_PIN_ALTR						
arguments	input	PIN_NUM	DIO_PIN_t				
		the pin u want to set as an input					
		PORTNUM	DIO_PORT_t				
		the port					
		PIN_ALTR	PIN_ALTR_FUNC_t				
		pin alternative func					
return	NONE						
description	set pin as alternative func						
GPT API							
function name	TIMER_Init						
arguments	NONE						
return	NONE						
description	init TIMER FREQ , PRESCALE , FUNC BASED ON CONFIG						

name	TIMER_NUM_T						
type	Enumeration						
range	0:TIMERS NUMBER						
description	enum listing ALL TIMERS						
function name	TIMERCOUNTER_CLR						
arguments	INPUT	TIMER_NUM	TIMER_NUM_T				
		which timer to use					
return	NONE						
description	clr the counter or timer value						
function name	TIMERCOUNTER_START						
arguments	INPUT	TIMER_NUM	TIMER_NUM_T				
		which timer to use					
return	NONE						
description	start the counter or timer value						
function name	TIMERCOUNTER_STOP						
arguments	INPUT	TIMER_NUM	TIMER_NUM_T				
		which timer to use					
return	NONE						
description	STOP the counter or timer value						
function name	TIMERCOUNTER_GET						
arguments	INPUT	TIMER_NUM	TIMER_NUM_T				
		which timer to use					
return	NONE						
description	GET the counter or timer value						
CAN API							
name	CAN_MSG_t						
type	structure						
elements	CAR_STATE_S	_C_M	the data is cr state				
description	a struct to hold the msg data the can protocol will send						
function name	CAN_Init						
arguments	NONE						
return	NONE						
description	init CAN protocol with config						
function name	CAN_SEND						
arguments	input	CAN_MSG_t	msg				
		the msg that will be sent					
return	NONE						
description	CAN SEND MSG GIVIN						
function name	CAN_RESCEVE						
arguments	NONE						
return	CAN_MSG_t						
description	can recieve msg from another ECU						
switch model API							
name	switch_state_t						
type	Enumeration						
range	SWITCH_OFF	0	switch not pressed				
	SWITCH_ON	1	switch pressed				
description	enum listing switch states						
name	_PORT_ARR						

type	Uint32 *						
description	an array hold all the HW Port Adresses						
function name	Switch_Init						
arguments	input	PIN_NUM	DIO_PIN_t				
		the pin u want to set as an input					
		PORTNUM	DIO_PORT_t				
		the port of that pin					
return	NONE						
description	init switch input pin						
function name	Get_Switch_State						
arguments	input	PIN_NUM	DIO_PIN_t				
		the pin u want to set as an input					
		PORTNUM	DIO_PORT_t				
		the port of that pin					
return	switch_state_t						
description	using DIO API and _PORT_ARR and switch_pin_port structure to get the state of the switch 0 for not pressed , 1 for pressed						
door API							
name	door_state_t						
type	Enumeration						
range	DOOR_CLOSED	0	door is closed				
	DOOR_OPEN	1	door is open				
description	enum listing door states						
name	_PORT_ARR						
type	Uint32 *						
description	an array hold all the HW Port Adresses						
function name	DoorSensorInit						
arguments	input	PIN_NUM	DIO_PIN_t				
		the pin u want to set as an input					
		PORTNUM	DIO_PORT_t				
		the port of that pin					
return	None						
description	init the door sensor input pin						
function name	Get_DoorSensor_State						
arguments	input	PIN_NUM	DIO_PIN_t				
		the pin u want to set as an input					
		PORTNUM	DIO_PORT_t				
		the port of that pin					
return	door_state_t						
description	using DIO API and _PORT_ARR to get the state of the switch 0 for not pressed , 1 for pressed						
motion API							
name	motion_state_t						
type	Enumeration						
range	NOT_MOVING	0	car isn't moving				
	MOVING	1	car is moving				
description	enum listing motion states						
name	_PORT_ARR						
type	Uint32 *						
description	an array hold all the HW Port Adresses						

function name	SpeedSensorInit						
arguments	input	PIN_NUM	DIO_PIN_t				
		the pin u want to set as an input					
		PORTNUM	DIO_PORT_t				
		the port of that pin					
return	None						
description	init the motionsensor input pin might init the Timer						
function name	Get_SpeedSensor_State						
arguments	None						
return	motion_state_t						
description	use Timer and GPIO APIs to get the motion state						
can handler API							
name	can_handler_msg_s						
type	structure						
elements	comm_manger_msg_s		C_M_S	the msg from comm manager			
	e2			optional can handler padding			
description	a struct to hold comm_manger_msg_s and can handler padding						
function name	CAN_HANDLER_INIT						
arguments	None						
return	NONE						
description	init the comm. protocol used based on config						
function name	CAN_HANDLER_SEND						
arguments	input	comm_manger_msg_s	C_M_S				
		the comm manger msg structre					
return	NONE						
description	send the can_handler_msg_s msg using can API						
function name	CAN_HANDLER_RCV						
arguments	NONE						
return	comm_manger_msg_s						
description	return the comm_manger_msg_s receved from theCAN using CAN API						
comm manger API							
name	comm_manger_msg_s						
type	structure						
elements	CAR_STATE_S	_C_S	have the door state				
	e2		optional comm manger padding				
description	a struct to hold all the states and comm manger padding						
name	protocol_t						
type	Enumeration						
range	lis of all the protocol that can be used						
description	enum listing the the ECU comm protocols						
function name	COMM_MANAGER_INIT						
arguments	None						
return	NONE						
description	init all the comm. protocols used based on config						
function name	COMM_MANAGER_send						
arguments	input	CAR_STATE_S	C_S				
		the car state structre					
		protocol_t	protocol				

		the protocol used to send					
return	NONE						
description	send msg using the choosed protocol API as comm_manger_msg_s						
function name	COMM_MANAGER_rcv						
arguments	input	protocol_t	protocol				
		the protocol used to send					
return	CAR_STATE_S						
description	rcv car state using that comm protocol API						
app							
name	car_state_t						
type	Enumeration						
range	N_C_S	0	switch not pressed, door is closed ,car not moving				
	N_C_M	1	switch not pressed, door is closed ,car is moving				
	N_O_S	2	switch not pressed, door is open ,car not moving				
	N_O_M	3	switch not pressed, door is open ,car is moving				
	P_C_S	4	switch pressed, door is closed ,car not moving				
	P_C_M	5	switch pressed, door is closed ,car is moving				
	P_O_S	6	switch pressed, door is open ,car not moving				
	P_O_M	7	switch pressed, door is open ,car is moving				
description	enum listing the car sensors states						
name	CAR_STATE_S						
type	structure						
elements	door_state_t	_D_S	have the door state				
	motion_state_t	_M_S	have the motion state				
	switch_state_t	_S_S	have the switch state				
	car_state_t	_C_S	have the car state which will be the result of the three				
description	a struct to hold all the states and change it in the run time						
name	STATES						
type	CAR_STATE_S						
description	that is the global structure that will set the states in						
function name	switch_state_task						
arguments	NONE						
periodacity	20ms						
return	NONE						
description	setting new switch_state_t and new car_state_t in the STATES global variable using switch API						
function name	door_state_task						
arguments	NONE						
periodacity	10ms						
return	NONE						
description	setting new door_state_t and new car_state_t in the STATES global variable using door API						
function name	motion_state_task						
arguments	NONE						
periodacity	5ms						
return	NONE						
description	setting new motion_state_t and new car_state_t in the STATES global variable using motion API						
function name	data_send_task						

arguments	NONE						
periodacity	5ms						
return	NONE						
description	sending the STATES global variable as a msg using the can protocol using COMM_MANAGER_SEND						