Function name: **Motor_init**			
Arguments	Input	Motor_num	Type: int
		Description: carry the number of the motor that we want to give order to.	
	Output	----------	
	Input/Output	----------	
Return	E_OK	0	
	E_NOK	1	
Description	Setting initial conditions for motors		

Function name: **Motor_start**			
Arguments	Input	Motor_num	Type: int
		Description: carry the number of the motor that we want to give order to.	
	Output	----------	
	Input/Output	----------	
Return	E_OK	0	
	E_NOK	1	
Description	Let motors start		

<table>
<tr><td colspan="4">Function name: Motor_stop</td></tr>
<tr><td rowspan="4">Arguments</td><td rowspan="2">Input</td><td>Motor_num</td><td>Type: int</td></tr>
<tr><td colspan="2">Description: carry the number of the motor that we want to give order to.</td></tr>
<tr><td>Output</td><td colspan="2">----------</td></tr>
<tr><td>Input/Output</td><td colspan="2">----------</td></tr>
<tr><td rowspan="2">Return</td><td>E_OK</td><td colspan="2">0</td></tr>
<tr><td>E_NOK</td><td colspan="2">1</td></tr>
<tr><td>Description</td><td colspan="3">Let motors stop</td></tr>
</table>

<table>
<tr><td colspan="4">Function name: DIO_init</td></tr>
<tr><td rowspan="8">Arguments</td><td rowspan="6">Input</td><td>Motor_num</td><td>Type: int</td></tr>
<tr><td colspan="2">Description: carry the number of the motor that we want to give order to.</td></tr>
<tr><td>H_bridge_num</td><td>Type:int</td></tr>
<tr><td colspan="2">Description: carry the number of the H bridge that we want to give order to.</td></tr>
<tr><td>selection</td><td>Type:string</td></tr>
<tr><td colspan="2">Description: select which module to control.</td></tr>
<tr><td>Output</td><td colspan="2">----------</td></tr>
<tr><td>Input/Output</td><td colspan="2">----------</td></tr>
<tr><td rowspan="2">Return</td><td>E_NOK</td><td colspan="2">1</td></tr>
<tr><td>E_OK</td><td colspan="2">0</td></tr>
<tr><td>Description</td><td colspan="3">Setting initial conditions for DIO</td></tr>
</table>

<table>
<tr><td colspan="4">Function name: DIO_read</td></tr>
<tr><td rowspan="8">Arguments</td><td rowspan="8">Input</td><td>Motor_num</td><td>Type: int</td></tr>
<tr><td colspan="2">Description: carry the number of the motor that we want to give order to.</td></tr>
<tr><td>H_bridge_num</td><td>Type:int</td></tr>
<tr><td colspan="2">Description: carry the number of the H bridge that we want to give order to.</td></tr>
<tr><td>selection</td><td>Type:string</td></tr>
<tr><td colspan="2">Description: select which module to control.</td></tr>
<tr><td colspan="2">Output</td></tr>
<tr><td colspan="2">Data_read (pointer)</td></tr>
<tr><td colspan="2">Input/Output</td><td colspan="2">----------</td></tr>
<tr><td rowspan="2">Return</td><td>E_NOK</td><td colspan="2">1</td></tr>
<tr><td>E_OK</td><td colspan="2">0</td></tr>
<tr><td>Description</td><td colspan="3">Read from selected pins</td></tr>
</table>

<table>
<tr><td colspan="4">Function name: DIO_write</td></tr>
<tr><td rowspan="8">Arguments</td><td rowspan="8">Input</td><td>Motor_num</td><td>Type: int</td></tr>
<tr><td colspan="2">Description: carry the number of the motor that we want to give order to.</td></tr>
<tr><td>H_bridge_num</td><td>Type:int</td></tr>
<tr><td colspan="2">Description: carry the number of the H bridge that we want to give order to.</td></tr>
<tr><td>selection</td><td>Type:string</td></tr>
<tr><td colspan="2">Description: select which module to control.</td></tr>
<tr><td colspan="2">Output</td></tr>
<tr><td colspan="2">----------</td></tr>
<tr><td colspan="2">Input/Output</td></tr>
<tr><td colspan="2">Data_write (int)</td></tr>
<tr><td rowspan="2">Return</td><td>E_NOK</td><td colspan="2">1</td></tr>
<tr><td>E_OK</td><td colspan="2">0</td></tr>
<tr><td>Description</td><td colspan="3">Write on the selected pins for DIO</td></tr>
</table>

Function name: **LCD_init**		
Arguments	Input	----------
	Output	----------
	Input/Output	----------
Return	E_NOK	1
	E_OK	0
Description	Setting initial conditions for LCD	

Function name: **LCD_Display**			
Arguments	Input	direction	Type: string
		Description: carry the direction that robot move in.	
	Output	----------	
	Input/Output	----------	
Return	E_NOK	1	
	E_OK	0	
Description	Take the direction and display it		

Function name: **RobotControl_init**		
Arguments	Input	----------
	Output	----------
	Input/Output	----------
Return	E_NOK	1
	E_OK	0
Description	Setting initial conditions for RobotControl	

Function name: **RobotControl_Start**			
Arguments	Input	direction	Type: string
		Description: carry the direction desired.	
	Output	----------	
	Input/Output	Direction	
Return	E_NOK	1	
	E_OK	0	
Description	To start controlling robot movement from H brige as unit		

Function name: **RobotControl_update**			
Arguments	Input	Argentcase (type:u8)	
		Decrease_speed_ob	Type:u8
		turnright_ob	Type:u8
		New_direction	Type: string
		Description: new direction	
	Output	----------	
	Input/Output	New_direction/ Decrease_speed_ob/ turnright_ob/argent	
Return	E_NOK	1	
	E_OK	0	
Description	To give new direction of moving of the robot and to deal with different obstacle cases.		

Function name: **PWM_init**			
Arguments	Input	Cycle	
	Output	----------	
	Input/Output	Timer_numer_HWkit	
Return	E_NOK	1	
	E_OK	0	
Description	Setting initial conditions for PWM		

Function name: **PWM_Start**			
Arguments	Input	Motor_num	Type: int
		Description: motor number to send data to	
		H_bridge_num	Type:int
		Description: carry the number of the H bridge that we want to give order to.	
		speed	Type:float
		Description: select desired speed.	
		Ultrasonic	Type:u8
		To pass the signal to ultrasonic sensor	
	Output	----------	
	Input/Output	Motor num/ H_bridge_num	
Return	E_NOK	1	
	E_OK	0	
Description	To start PWM signal		

Function name: **PWM_Stop**			
Arguments	Input	Motor_num	Type: int
		Description: motor_number to send data	
		H_bridge_num	Type:int
		Description: carry the number of the H bridge that we want to give order to.	
	Output	----------	
	Input/Output	Motor_num/ H_bridge_num	
Return	E_NOK	1	
	E_OK	0	
Description	To stop PWM signal		

Function name: **Timer_init**		
Arguments	Input	Timer mode(type :u8)
	Output	----------
	Input/Output	Timer_numer_HW_kit
Return	E_NOK	1
	E_OK	0
Description	Setting initial conditions for Timer	

Function name: **Timer_Start**			
Arguments	Input	duration	Type: float
		Description: carry the desired duration in seconds	
	Output	----------	
	Input/Output	Current _value	
Return	E_NOK	1	
	E_OK	0	
Description	To start timer with certain value and returns the current value		

Function name: **Timer_reset**			
Arguments	Input	new_value	Type: float
		Description: carry the current value	
	Output	----------	
	Input/Output	new _value	
Return	E_NOK	1	
	E_OK	0	
Description	To stop timer or reset it and returns the current value		

Function name: **Ultrasonic_init**		
Arguments	Input	Second_alert_distance First_alert_distance
	Output	----------
	Input/Output	
Return	E_NOK	1
	E_OK	0
Description	Setting initial conditions for Timer	

Function name: **ultrasonic_distanecePhase**		
Arguments	Input	----------
	Output	----------
	Input/Output	distance
Return	E_NOK	1
	E_OK	0
Description	To detect distance