Function name: **Motor_init**			
Arguments	Input	Motor_num	Type: u8 Range:1- 4
		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: u8 Range: 1-4
		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		

Function name: **Motor_stop**			
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 stop		

Function name: **DIO_init**			
Arguments	Input	port	Type: pointer to u8
		Description: carry the number of the motor that we want to give order to.	
		selection	Type:pointer to string Range: 0-2
		Description: select which module to control.	
	Output	----------	
	Input/Output	----------	
Return	E_NOK	1	
	E_OK	0	
Description	Setting initial conditions for DIO		

<table>
<tr><td colspan="4">Function name: DIO_read</td></tr>
<tr><td rowspan="6">Arguments</td><td rowspan="5">Input</td><td>port</td><td>Type: pointer to u8</td></tr>
<tr><td colspan="2">Description: carry ports</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">Data_read (pointer to u8)</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">Read from selected pins</td></tr>
</table>

<table>
<tr><td colspan="4">Function name: DIO_write</td></tr>
<tr><td rowspan="6">Arguments</td><td rowspan="5">Input</td><td>ports</td><td>Type: pointer to u8</td></tr>
<tr><td colspan="2">Description: carry the number of the motor that we want to give order to.</td></tr>
<tr><td>selection</td><td>Type:pointer to 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">Data_write (poiter to u8)</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:pointer to 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_update**			
Arguments	Input		
		distance	Type:pointer to u8
		Range: 1- 100	
	Output	Action (pointer to u8)	
	Input/Output		
Return	E_NOK	1	
	E_OK	0	
Description	To give new Action 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 Argent_distance
	Output	----------
	Input/Output	----------
Return	E_NOK	1
	E_OK	0

Description	Setting initial conditions for Timer

Function name: **ultrasonic_distanecePhase**		
Arguments	Input	----------
	Output	Distance (type: pointer to (u8))
	Input/Output	----------
Return	E_NOK	1
	E_OK	0
Description	To detect distance at some point	

Function name: **stear_Start**			
Arguments	Input	unit	Type:u8 Range: 0-1
		Description: carry the unit desired to control .	
	Output	----------	
	Input/Output	Direction (Type: pointer to u8)	
Return	E_NOK	1	
	E_OK	0	
Description	To start controlling robot movement from H brige as unit		

Function name: **stear_init**			
Arguments	Input	unit	Type:u8 Range: 0-1
		Description: carry the unit desired to control .	
	Output	----------	
	Input/Output	Direction (Type: pointer to u8)	
Return	E_NOK	1	
	E_OK	0	
Description	To start controlling robot movement from H brige as unit		

Function name: **stear_update**			
Arguments	Input	unit	Type:u8 Range: 0-1
		Description: carry the unit desired to control .	
	Output	----------	
	Input/Output	New_Direction (Type: pointer to u8) Range:0-3	
Return	E_NOK	1	
	E_OK	0	
Description	To start controlling robot movement from H brige as unit		