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		

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	Motor_num	Type: int
		Description: carry the number of the motor that we want to give order to.	
		H_bridge_num	Type:int
		Description: carry the number of the H bridge that we want to give order to.	
		selection	Type:string
		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="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_Stop**			
Arguments	Input		
		Stop_move	Type:bool
		Description: to stop move.	
	Output	----------	
	Input/Output	Stop_move	
Return	E_NOK	1	
	E_OK	0	
Description	To stop moving of the robot		

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

<table>
<tr><td colspan="4">Function name: PWM_Start</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: motor number to send data 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>speed</td><td>Type:float</td></tr>
<tr><td colspan="2">Description: select desired speed.</td></tr>
<tr><td>Output</td><td colspan="2">----------</td></tr>
<tr><td>Input/Output</td><td colspan="2">Motor num/ H_bridge_num</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">To start PWM signal</td></tr>
</table>

<table>
<tr><td colspan="4">Function name: PWM_Stop</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: motor_number to send data</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>Output</td><td colspan="2">----------</td></tr>
<tr><td>Input/Output</td><td colspan="2">Motor_num/ H_bridge_num</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">To stop PWM signal</td></tr>
</table>

Function name: **Timer_init**		
Arguments	Input	----------
	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_Stop**			
Arguments	Input	Current _value	Type: float
		Description: carry the current value	
	Output	----------	
	Input/Output	Current _value	
Return	E_NOK	1	
	E_OK	0	
Description	To stop timer and returns the current value		