Moving Robot API design documentation

DIO Module

Data Type Table

Name	DIO_port		
Type	Enumeration		
Range	PORTA	0	Symbolic name for PortA
	PORTB	1	Symbolic name for PortB
	PORTC	2	Symbolic name for PortC
	PORTD	3	Symbolic name for PortD
Description	Define symbolic names for ports		

Name	DIO_pin	
Type	Enumeration	
Range	Pin0 ~ Pin7	Available pins on each port
Description	Define symbolic names for available pins	

Name	DIO_Level		
Type	Enumeration		
Range	STD_LOW	0	Physical pin level = 0
	STD_HIGH	1	Physical pin level = 1
Description	Digital pin value		

Name	DIO_Direction	
Type	Enumeration	
Range	Input	0
	Output	1
Description	Define DIO port pin direction	

API Design Table

Function name	Dio_InitPortPin		
Arguments	Input	DIO_port	Enumeration
		Port number / symbolic name	
		DIO_Pin	Enumeration
		Pin number / symbolic name	
		DIO_Direction	Enumeration
		Define port pin direction.	
	Output		
	Input/ Output		
Return	E_OK	1	
	E_NOT_OK	0	
Description	Responsible for initializing a port pin direction. Must be specified before read/write access on a pin.		

Function name	Dio_Read		
Arguments	Input	DIO_Port	Enumeration
		Port number / symbolic name	
		DIO_Pin	Enumeration
		Pin number / symbolic name	
	Output	DIO_LEVEL	Enumeration *
		Pointer to Physical level of the specified pin.	
	Input/ Output		
Return	E_OK	1	
	E_NOT_OK	0	

Description	Responsible for reading the physical current value of a hardware port pin. It should be able to read the value of the pin whether it's input or output without affecting its current state. If the pin is uninitialized, the function should return an error and not do anything. For the output parameters it must check for a null pointer exception before proceeding.

Function name	Dio_Write		
Arguments	Input	DIO_Port	Enumeration
		Port number / symbolic name	
		DIO_Pin	Enumeration
		Pin number / symbolic name	
		DIO_LEVEL	Enumeration
		Physical level to write on the specified pin.	
	Output		
	Input / Output		
Return	E_OK	1	
	E_NOT_OK	0	
Description	Responsible for writing the physical value of a hardware port pin. It should be able to write the value of the pin if it's output without. If the pin is uninitialized, the function should return an error and not do anything.		

Timer Module

Data Type Table

Name	TMR_Channel	
Type	Enumeration	
Range	0,1,2	Channel 0, 1, 2
Description	Define symbolic names for available timer channels	

Name	TMR_CLK	
Type	Enumeration	
Range	CLK	0
	CLK/8	1
	CLK/64	2
	CLK/256	3
	CLK/1024	4
	External on pin T0 / falling edge	5
	External on pin T0 / rising edge	6
Description	Define symbolic names for available clock sources	

Name	TMR_Config	
Type	Structure	
Elements	TMR_Channel	enumeration
	TMR_CLK	Enumeration
	TMR_IRQ_EN	Boolean
Description	Configuration parameters for timer module	

Name	TMR_Ticks	
Type	Uint16	
Range	channel 0, channel 2	0 - 255
	channel 1	0 - 65535
Description	Number of ticks before next timer flag	

API Design Table

Function name	TMR_init		
Arguments	Input	TMR_Config	structure

		Pointer to structure address holding timer configuration parameters.
	Output	
	Input / Output	
Return	E_OK	1
	E_NOT_OK	0
Description	Function responsible for initializing a timer channel according to configuration parameters. This function isn't supposed to handle PWM mode. Function must be initialized before using Start/Stop functions Function return E_NOT_OK when: • If a nullptr detected • If the TMR_Channel isn't valid • Channel not initialized	

Function name	TMR_Start		
Arguments	Input	TMR_Channel	Enumeration
		Channel number for the timer to start.	
		TMR_Ticks	uint16
		Initial value to store in the timer register	
	Output		
	Input / Output		
Return	E_OK	1	
	E_NOT_OK	0	
Description	Function responsible for starting the work of the timer specified with a starting value TMR_Ticks in the timer/counter register.		

Function name	TMR_Stop		
Arguments	Input	TMR_Channel	Enumeration
		Channel number for the timer to stop.	
	Output		

	Input / Output	
Return	E_OK	1
	E_NOT_OK	0
Description	Function responsible for stop the operation of the timer channel specified by the parameter TMR_Channel.	

Function name	TMR_GetStatus		
Arguments	Input	TMR_Channel	Enumeration
		Channel number for the timer to start.	
	Output	TMR_status	Boolean *
		Pointer to where to store the timer status. A value 1 should be stored if overflow occurred and 0 otherwise.	
	Input / Output		
Return	E_OK	1	
	E_NOT_OK	0	
Description	Function responsible for getting flag status.		

PWM module

Data Type Table

Name	PWM_Config		
Type	Structure		
Elements	TMR_Channel	enumeration	
	TMR_CLK	Enumeration	
	TMR_Operation_Mode	Boolean	0: PWM 1: Fast PWM
Description	Configuration parameters for timer module		

API Design Table

Function name	PWM_Init		
Arguments	Input	PWM_Config	structure
		Pointer to structure address holding pwm configuration parameters.	
	Output		
	Input / Output		
Return	E_OK	1	
	E_NOT_OK	0	
Description	Function responsible for initializing a timer channel for PWM operation according to configuration parameters. This function isn't supposed to handle other timer modes. Function must be initialized before using Start/Stop functions Function return E_NOT_OK when: • If a nullptr detected • If the TMR_Channel isn't valid		

Function name	PWM_Start		
Arguments	Input	TMR_Channel	Enumeration
		Channel number for the pwm to start.	
	Output		
	Input / Output		
Return	E_OK	1	
	E_NOT_OK	0	
Description	Function responsible for starting the work of the pwm specified. If the TMR_Channel passed isn't available for PWM operation, the function should return an error and do nothing.		

Function name	PWM_Stop		
Arguments	Input	TMR_Channel	Enumeration
		Channel number for the pwm to stop.	

	Output	
	Input / Output	
Return	E_OK	1
	E_NOT_OK	0
Description	Function responsible for stopping the work of the pwm specified. If the TMR_Channel passed isn't available for PWM operation, the function should return an error and do nothing.	

LCD Module

Data Type Table

Name	LCD_Command	
Type	Enumeration	
Range	CLEAR_DISPLAY	0
	CURSOR_HOME	1
	DISPLAY_ON_OFF	2
	READ_BUSY_FLAG	3
Description	A set of commands to pass to character LCD controller	

API Design Table

Function name	LCD_Init	
Arguments	Input	
	Output	
	Input / Output	
Return	E_OK	1
	E_NOT_OK	0
Description	Function responsible for initializing LCD module as well as clearing the screen and initializing the cursor. Pins for: LCD_Data_Pins, LCD_EN_Pin, LCD_RS_Pin, and LCD_RW_Pin must be predefined before function call.	

Function name	LCD_Display		
Arguments	Input	LCD_String_Disp	Uint8*
		Pointer to the start of the string to display	
		LCD_String_Length	Uint8
		Length of the LCD_String_Disp	
	Output		
	Input / Output		
Return	E_OK	1	
	E_NOT_OK	0	
Description	Function responsible for displaying a string on the screen. It should display the text from the current cursor location. It's not responsible for setting the cursor or clearing the screen		

Function name	LCD_Command		
Arguments	Input	LCD_Command	Enumeration
		Symbolic name of a command to pass to LCD controller	
	Output		
	Input / Output		
Return	E_OK	1	
	E_NOT_OK	0	
Description	Function responsible for carrying out a command to the LCD controller		

Motor Module

Data Type Table

Name	MOTOR_ID	
Type	Enumeration	
Range	RIGHT_MOTOR	0
	LEFT_MOTOR	1
Description	Define symbolic names for available motors	

Name	MOTOR_Config		
Type	Structure		
Elements	MOTOR_ID	Enumeration	Identifier to reference the motor with functions start/stop
	MOTOR_Dir1_Pin	DIO_pin	DIO port pin for direction
	MOTOR_Dir2_Pin	DIO_pin	DIO port pin for direction
	MOTOR_Speed_Pin	DIO_pin	DIO port pin for speed
	MOTOR_PWM_CH	TMR_Channel	Channel of PWM to assign to the motor
Description	Structure defining configuration parameters for each motor		

API Design Table

Function name	MOTOR_Init		
Arguments	Input	MOTOR_Config	structure
		Pointer to structure address holding motor configuration parameters.	
	Output		
	Input / Output		
Return	E_OK	1	
	E_NOT_OK	0	
Description	Function responsible for initializing one motor instance and specify its DIO pins and their required configuration. For the input parameter MOTOR_Config a null pointer check must be performed. If a nullptr detected, it should return an error and do nothing		

Function name	MOTOR_Start		
Arguments	Input	MOTOR_ID	Enumeration
		Motor identifier to select which motor to start	

		MOTOR_Speed	Uint8
		Define the PWM signal where 0 means motor stopping and (2^n)-1 means max speed. And n is the number of bits of the PWM channel.	
	Output		
	Input / Output		
Return	E_OK	1	
	E_NOT_OK	0	
Description	Function responsible for starting the work of the specified motor according to the specified speed. If the passed MOTOR_ID argument is not valid, the function should return an error and do nothing.		

Function name	MOTOR_Stop		
Arguments	Input	MOTOR_ID	Enumeration
		Motor identifier to select which motor to start	
	Output		
	Input / Output		
Return	E_OK	1	
	E_NOT_OK	0	
Description	Function responsible for stopping the work of the specified motor. If the passed MOTOR_ID argument is not valid, the function should return an error and do nothing.		

ROBOT CONTROL Module

Function name	ROBOT_Init	
Arguments	Input	
	Output	
	Input / Output	

Return	E_OK	1
	E_NOT_OK	0
Description	Function responsible for initializing Robot module. Pins for: Motors and LCD must be predefined before function call.	

Function name	ROBOT_Start	
Arguments	Input	
	Output	
	Input / Output	
Return	E_OK	1
	E_NOT_OK	0
Description	Function responsible for starting the work of the robot and its associated modules (LCD, Motors, Timers, PWM).	

Function name	ROBOT_Stop	
Arguments	Input	
	Output	
	Input / Output	
Return	E_OK	1
	E_NOT_OK	0
Description	Function responsible for stopping the work of the robot and its associated modules (LCD, Motors, Timers, PWM).	

Function name	ROBOT_MoveUpdate	
Arguments	Input	
	Output	
	Input / Output	

Return	E_OK	1
	E_NOT_OK	0
Description	Function responsible for updating the work of the robot. It's a periodic function and should be constantly called to control the robot state from Power to running to stop mode.	