APIs Design for a moving Robot modules

By/ Ahmed Maged

DIO APIs

Function Name	DIO_eSetPinDirection(PrtID_t PortIdCpy , PinId_t PinIdCpy, DirID_t PinDirCpy)		
Arguments	inputs	PortIdCpy	enumeration
		description: the PORT contain the PIN to set direction	
		PinIdCpy	enumeration
		description: the pin to set direction	
		PinDirCpy	enumeration
		the direction you want whether input or output	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to set the pin direction input or output		

Name	PortIdCpy		
Type	enumeration		
Range	PORTA	0	For port A
	PORTB	1	For port B
	PORTC	2	For port C
	PORTD	3	For port D
Description	These values are to determine which port to be affected by the function		

Name	PinIdCpy		
Type	enumeration		
Range	PIN0	0	For pin 0
	PIN1	1	For pin 1
	PIN2	2	For pin 2
	PIN3	3	For pin 3
	PIN4	4	For pin 4
	PIN5	5	For pin 5
	PIN6	6	For pin 6
	PIN7	7	For pin 7
Description	These values are to determine which pin to be affected by the function		

Name	PinDirCpy		
Type	enumeration		
Range	Dir_OUTPUT	0	To be output
	Dir_INPUT	1	To be input
Description	These values are to determine the direction of pin whether output or input		

Function Name	DIO_eSetPinValue(PrtID_t PortIdCpy , PinId_t PinIdCpy, PinVal_t PinValueCpy)		
Arguments	inputs	PortIdCpy	enumeration
		description: the PORT contain the PIN to set direction	
		PinIdCpy	enumeration
		description: the pin to set direction	
		PinValueCpy	enumeration
		the direction you want wether input or output	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to set the pin value HIGH or LOW		

Name	PortIdCpy		
Type	enumeration		
Range	PORTA	0	For port A
	PORTB	1	For port B
	PORTC	2	For port C
	PORTD	3	For port D
Description	These values are to determine which port to be affected by the function		

Name	PinIdCpy		
Type	enumeration		
Range	PIN0	0	For pin 0
	PIN1	1	For pin 1
	PIN2	2	For pin 2
	PIN3	3	For pin 3
	PIN4	4	For pin 4
	PIN5	5	For pin 5
	PIN6	6	For pin 6
	PIN7	7	For pin 7
Description	These values are to determine which pin to be affected by the function		

Name	PinValueCpy		
Type	enumeration		
Range	LOW	0	To make the output low
	HIGH	1	To make the output high
Description	These values are to determine the value to be written to the output register.		

Function Name	DIO_eGetPinValue(PrtID_t PortIdCpy , PinId_t PinIdCpy, PinVal_t PinValueCpy , u8 * pOutputRegister)		
Arguments	inputs	PortIdCpy	enumeration
		description: the PORT contain the PIN to	
		PinIdCpy	enumeration
		description: the pin to set direction	
	outputs	pOutputRegister	u8 *
		description: the return location for the	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to get the pin value whether HIGH or LOW		

Name	PortIdCpy		
Type	enumeration		
Range	PORTA	0	For port A
	PORTB	1	For port B
	PORTC	2	For port C
	PORTD	3	For port D
Description	These values are to determine which port to be affected by the		

Name	PinIdCpy		
Type	enumeration		
Range	PIN0	0	For pin 0
	PIN1	1	For pin 1
	PIN2	2	For pin 2
	PIN3	3	For pin 3
	PIN4	4	For pin 4
	PIN5	5	For pin 5
	PIN6	6	For pin 6
	PIN7	7	For pin 7
Description	to determine which pin to be affected by the function		

MOTOR APIs

Function Name	MOTOR_eInit(PrtID_t MotorPortId, ChID_t ChannelIDcpy)		
Arguments	inputs	MotorPortId	enumeration
		description:	The motor existence port
		ChannelIDcpy	enumeration
		description:	The motor existence pins
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the motors		

Name	MotorPortId		
Type	enumeration		
Range	PORTA	0	For port A
	PORTB	1	For port B
	PORTC	2	For port C
	PORTD	3	For port D
Description	These values are to determine which port to be		

Name	ChannelIDcpy		
Type	enumeration		
Range	CHNL0	0	For channel 0
	CHNL1	1	For channel 1
Description	To determine which channel to be affected by the function, for every channel there are 2 pins specified in config file.		

Function Name	MOTOR_eStop(ChID_t ChannelIDcpy)		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	Call this api to stop the motor.		

Name	ChannelIDcpy		
Type	enumeration		
Range	CHNL0	0	For channel 0
	CHNL1	1	For channel 1
Description	To determine which channel to be affected by the function, for every channel there are 2 pins specified in config file.		

Function Name	MOTOR_eStart(ChID_t ChannelIDcpy, DirID_t DirectionCpy)		
Arguments	inputs	ChannelIDcpy	enumeration
		description: enumeration contain the channel to be affected by the fn.	
		DirectionCpy	enumeration
		Description: enumeration contain the direction.	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to start the motors whether the (clock wise or counter clock wise)		

Name	DirectionCpy		
Type	enumeration		
Range	CW	0	Move clock wise
	CCW	1	Move counter clock wise
Description	These values are controlling the direction of Motor.		

ROBOT Control APIs

Function Name	ROBOT_eInit(void)		
Arguments	inputs	N/A	
		description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the Robot Control module and the needed other modules		

Function Name	Robot_eUpdateMoving(void)		
Arguments	inputs	N/A	
		description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api periodically to update the moving direction and speed		

Timer APIs

Function Name	Timer_eInit(void)		
Arguments	inputs	N/A	
		description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	Call this api to initialize the timer as specified in the configuration file. Timer module must use Timer0 in hardware.		

Function Name	Timer_eStart(ChID_t ChannelIDcpy)		
Arguments	inputs	ChannelIDcpy	enumeration
		description: the channel to start	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to make the timer start counting from zero		

Name	ChannelIDcpy		
Type	enumeration		
Range	CHNL0	0	For Timer 0
	CHNL1	1	For Timer 1
Description	To determine which timer to be affected by the function.		

Function Name	Timer_eGetCurrentTiming_mS(ChID_t ChannelIDcpy , u32* u32Current_mS)		
Arguments	inputs	ChannelIDcpy	enumeration
		description: the channel to read	
	outputs	u32Current_mS	u32
		Description: used to return the elapsed time (in mS) since the timer was started from zero using Timer_eStart API.	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api get the current second elapsed from calling Timer_eStart();		

Name	u32Current_mS	
Type	U32	
Range	0	The least value
	3 600 000 000	The max value you can get, Equal 1000 hours
Description	These is the elapsed time since starting count from zero.	

Name	ChannelIDcpy		
Type	enumeration		
Range	CHNL0	0	For Timer 0
	CHNL1	1	For Timer 1
Description	To determine which timer to be affected by the function.		

Function Name	Timer_eStop(ChID_t ChannelIDcpy)		
Arguments	inputs	ChannelIDcpy	enumeration
		description: the channel to stop	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to stop the timer		

PWM APIs

Function Name	PWM_eInit(void)		
Arguments	inputs	N/A	
		description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the pwm module to the Timer1 hardware module		

Function Name	PWM_eStart(ChID_t ChannelIDcpy, u8 DutyCycle_cpy, u8 Freq_cpy)		
Arguments	inputs	ChannelIDcpy	enumeration
		description: the channel to start the pwm signal	
		DutyCycle_cpy	U8
		description: the DutyCycle of the signal	
		Freq_cpy	U8
		description: the frequency of the signal	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to start the pwm		

Name	DutyCycle_cpy	
Type	U8	
Range	0	The least value (equal logic LOW)
	100	The max value (equal logic HIGH)
Description	These is the DutyCycle of the signal	

Name	Freq_cpy	
Type	U32	
Range	0	The least freq
	100000	The max freq (10uS period)
Description	These is the DutyCycle of the signal	

Name	ChannelIDcpy		
Type	enumeration		
Range	CHNL0	0	For Timer 0
	CHNL1	1	For Timer 1
Description	To determine which timer to be affected by the function.		

Function Name	PWM_eStop(ChID_t ChannelIDcpy)		
Arguments	inputs	ChannelIDcpy	enumeration
		description: the channel to start the pwm signal	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to stop the stop		

Name	ChannelIDcpy		
Type	enumeration		
Range	CHNL0	0	For Timer 0
	CHNL1	1	For Timer 1
Description	To determine which timer to be affected by the function.		

LCD APIs

Function Name	LCD_eInit(void)		
Arguments	inputs	N/A	
		description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the lcd as specified in the lcd configuration file but limited to PORTC (from pin0 : pin10)		

Function Name	LCD_eSendCommand(Cmd_t CmdCpy)		
Arguments	inputs	u8CmdCpy	enumeration
		description: a copy of the command to send to the lcd	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to set the pin direction input or output		

Name	CmdCpy		
Type	enumeration		
Range	lcd_Clear	0	Clear the screen
	lcd_Home	1	Move to the first position in first row
	lcd_DisplayOff	2	Disable the display
	lcd_DisplayOn	3	enable the display
Description	These values are the commands to be sent to the lcd.		

Function Name	LCD_eGotoxy(const u8 XPosCpy,const u8 YPosCpy)		
Arguments	inputs	XPosCpy	const u8
		description: the horizontal position starting from 0:15 for 2x16 lcd	
		YPosCpy	const u8
		description: the vertical position (0:1) for 2x16 lcd	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to go to specific position on the lcd		

Name	XPosCpy	
Type	U8	
Range	0	The first position in the screen starting from left
	15	The last position in the screen starting from left
Description	These values are the horizontal positions in a 2x16 LCD	

Name	YPosCpy	
Type	U8	
Range	0	The first row in the screen starting from upper row
	1	The second row in the screen starting from upper row
Description	These values are the horizontal positions in a 2x16 LCD	

Function Name	LCD_bWriteChar(const u8 u8DataCpy)		
Arguments	inputs	u8DataCpy	const u8
		description: the charcter to be writen in ascii representation	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to write a specific character in the current cursor position		

Name	u8DataCpy	
Type	U8	
Range	0	The least decimal value
	127	The last decimal value
Description	These values are the decimal representation of ascii code.	