Robot case study

Static design:

Modules:

Drivers:

- Timer
- motor
- LCD
- GPIO

Software components:

- Power up
- Run

APIs:

GPIO:

- Void GPIO_SetDir(u8 PORT, u8 PIN)
- Void GPIO_WriteHigh(u8 PORT, u8 PIN)
- Void GPIO_WriteLow(u8 PORT, u8 PIN)
- Void GPIO_ModeSelect(u8 PORT, u8 PIN)
- Void GPIO_ReadPin(u8 PORT, u8 PIN)

Timer:

- Void Timer_Init(u8 TIMER_NO, struct Tiemr_Config_Params* Ptr)
- Void Timer_ISR(void)
- Void Timer_SetTime(u8 TIMER_NO, u32 Time)
- Void Timer_SetPWM(u8 TIMER_NO, u8 Pwm, u32 Freq)
- Void Timer_Clear(u8 TIMER_NO)
- u8 Timer_GetTime(u8 TIMER_NO, u32* TimeRead)
- u8 Timer_Stop(u8 TIMER_NO)

Motor:

- void MOTOR_Init(u8 MOTOR_NO)
- void MOTOR_Move(u8 MOTOR_NO, u8 DIRECTION)

LCD:

- void LCD_Init(u8 BUS_SELECTION)
- u8 LCD_PutChar(char Character)
- u8 LCD_PutString(char* Str)
- u8 LCD_SendCommand(char command)
- void LCD_SetCursorPos(u8 row, u8 col)

Power up:

- void PowerUp_Init(void)
- void PowerUp_Update(void)
- void PowerUp_MoveToDir(u8 MOTOR_NO, u8 DIRECTION)

Run:

- void Run_Init(void)
- void Run_Update(void)
- void MoveToDir(u8 MOTOR_NO, u8 DIRECTION)

Data Types:

Enumerations:

Name:	PORT_NO	
Type:	Enumerations	
Range:	PORTA	0
	PORTB	1
	PORTC	2
	PORTD	3
	PORTE	4
	PORTF	5
Description:	PORTs Declarations as constant symbols for Tiva C TM4C123 MCU	

Name:	PIN_NO	
Type:	Enumerations	
Range:	PIN0	0
	PIN1	1
	PIN2	2
	PIN3	3
	PIN4	4
	PIN5	5
	PIN6	6
	PIN7	5
Description:	PORTs Declarations as constant symbols	

Name:	TIMER_NO	
Type:	Enumeration	
Range:	TIMER1	1
	TIMER2	2
	TIMER3	3
Description:	The count of Timer peripherals as constant symbols on MCU	

Name:	MOTOR_NO	
Type:	Enumeration	
Range:	MOTOR1	1
	MOTOR2	2
	MOTOR3	3
	MOTOR4	4
Description:	The count of Motors as constant symbols on board	

Name:	BUS_SELECTION	
Type:	Enumeration	
Range:	FOURBIT_BUSS	1
	EIGHTBIT_BUSS	2
Description:	Selection of LCD busses (either 4 bits or 8 bits)	

Name:	DIRECTION	
Type:	Enumeration	
Range:	UP	0
	DOWN	2
	RIGHT	3

	LEFT	4
Description:	Direction set of motor	

Variables:

Name:	Time
Type:	variable
Range:	Max value of uin32
Description:	Represents the number of timer counts to be set
Name:	pwm
Type:	variable
Range:	0:100
Description:	Represents the percentage of high period to low period of signal

Name:	Freq
Type:	variable
Range:	Max value of uin32
Description:	Represents the set value of operated signal frequency

Name:	Character
Type:	variable
Range:	Range of signed char type
Description:	Represents the input characters to LCD

Name:	Command
Type:	variable
Range:	Max value of uin32
Description:	Represents the command in hex representation sent to LCD

Name:	col
Type:	variable
Range:	Max value of u8
Description:	Represents the number of columns in the range which is supported in LCD

Name:	Freq
Type:	variable
Range:	Max value of uin32
Description:	Represents the set value of operated signal frequency

Name:	Str
Type:	pointer
Range:	Str will point to string of characters of length = passed string
Description:	Pointer to string passed to the function

Structures:

Name:	Timer_Config_Params	
Type:	structure	
Elements:	Timer_Mode	Selects the mode of operation of timer
	Timer_ClockSource_Select	Selects external or internal clock source and it's generator
	Timer_Prescalar	Selects the division value of clock
	TIMER_SHOT	Selects whether it's running in continuous mode or one shot
	TIMER_INTERRUPT	Enable or disable interrupt handler for this timer
	Timer_Enable	Enable or disable the timer
	Timer_Period	Period of timer counter
Description:	This structure initializes the timer registers	

APIs:

GPIO APIs:

Function name	GPIO_SetDir		
Arguments	Inputs	PIN	U8
	Outputs	PORT	U8
		PORT is pre-defined as a pointer to absolute address then dereferenced to get the actual value directly	
	Inputs/outputs		
Return	E_OK	1	
	E_NOK	0	
Description	Function sets the direction (input/output) for a specific pin in port		

Function name	GPIO_WriteHigh		
Arguments	Inputs	PIN	U8
	Outputs	PORT	U8
		PORT is pre-defined as a pointer to absolute address then dereferenced to get the actual value directly	
	Inputs/outputs		
Return	void		
Description	Function changes the state of GPIO pin to High		

Function name	GPIO_WriteLow		
Arguments	Inputs	PIN	U8
	Outputs	PORT	U8
		PORT is pre-defined as a pointer to absolute address then dereferenced to get the actual value directly	
	Inputs/outputs		
Return	void		
Description	Function changes the state of GPIO pin to Low		

Function name	GPIO_ModeSelect		
Arguments	Inputs	PIN	U8
		PORT	U8
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function selects the functionality of pin		

Function name	GPIO_ReadPin		
Arguments	Inputs	PIN	U8
		PORT	U8
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function to read pin		

Timer APIs:

Function name	Timer_Init		
Arguments	Inputs	TIMER_NO	U8
	Outputs	Ptr	Pointer to struct Timer_config_Params
	Inputs/outputs		
Return	void		
Description	Function initialize the timer configurations		

Function name	Timer_ISR	
Arguments	Inputs	void
	Outputs	
	Inputs/outputs	
Return	void	
Description	Interrupt handler function of timer	

Function name	Timer_SetPWM		
Arguments	Inputs	TIMER_NO	U8
		No of timer device	
		Pwm	U8
		Pwm value in percentage	
		Freq	U32
		Value of operation frequency of the signal	
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function set the value of pwm to control the output voltage		

Function name	Timer_SetTime		
Arguments	Inputs	TIMER_NO	U8
		No of timer device	
		Time	U32
		Value to be set in timer's register to assign the duration of timer	
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function set the timer to count for a specific duration of time		

Function name	Timer_TimeClear		
Arguments	Inputs	TIMER_NO	U8
		No of timer device	
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function Clears the preset time and reset the timer to another round of counting		

Function name	Timer_GetTime		
Arguments	Inputs	TIMER_NO	U8
		No of timer device	
	Outputs	TimeRead	Pointer to U32
	Inputs/outputs		
Return	E_OK	1	
	E_NOK	0	
Description	Function read the current timer value in the counting register		

Function name	Timer_Stop		
Arguments	Inputs	TIMER_NO	U8
		No of timer device	
	Outputs		
	Inputs/outputs		
Return	E_OK	1	
	E_NOK	0	
Description	Function stop the timer		

Function name	Motor_Init		
Arguments	Inputs	MOTOR_NO	U8
		No of motor device	
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function initialize motor variables		

Function name	Motor_Move		
Arguments	Inputs	MOTOR_NO	U8
		No of motor device	
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function initialize motor variables		

LCD APIs:

Function name	LCD_Init		
Arguments	Inputs	BUS_SELECTION	u8
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function initialize LCD software driver		

Function name	LCD_PutChar		
Arguments	Inputs	Character	u8
	Outputs		
	Inputs/outputs		
Return	Success or failure of sending operation	u8	
Description	Function put character on serial port to send		

Function name	LCD_PutString		
Arguments	Inputs	Str	u8*
	Outputs		
	Inputs/outputs		
Return	Success or failure of sending operation	u8	
Description	Function put String in buffer and send it on serial port byte by byte		

Function name	LCD_SendCommand		
Arguments	Inputs	Command	u8
	Outputs		
	Inputs/outputs		
Return	Success or failure of sending operation	u8	
Description	Function puts char represents special commands to LCD to send it on serial port		

Function name	LCD_SetCursorPos		
Arguments	Inputs	row	u8
		Col	u8
	Outputs		
	Inputs/outputs		

Return	Success or failure of sending operation	u8
Description	Function select the position of cursor on LCD to start writing or reading operation from this place	

Power up APIs:

Function name	PowerUp_Init		
Arguments	Inputs	void	
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function initialize necessary variables for robot power up duration		

Function name	PowerUp_Update		
Arguments	Inputs	void	
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function updates the state of robot during the duration of power up		

Function name	PowerUp_MoveToDir		
Arguments	Inputs	DIRECTION	u8
		MOTOR_NUM	U8
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function accepts the movement command and steer the robot to the intended direction during the power up stage/duration		

Run APIs:

Function name	Run_Init		
Arguments	Inputs	void	
	Outputs		
	Inputs/outputs		

Return	void
Description	Function initialize any variables or data structure needed to start Run stage/duration of the robot

Function name	Run_Update		
Arguments	Inputs	void	
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function updates the state of robot during the duration of running		

Function name	Run_MoveToDir		
Arguments	Inputs	DIRECTION	u8
		MOTOR_NUM	U8
	Outputs		
	Inputs/outputs		
Return	void		
Description	Function accepts the movement command and steer the robot to the intended direction		