

Moving Robot Design Document

Document Contents:

1. System Description.
2. Layered Architecture.
3. SW Data Type Tables.
4. SW Layers.
 - MCAL Layer.
 1. DIO.
 2. PWM.
 3. ICU.
 - On Board Layer.
 1. DCMotor.
 2. LCD.
 3. UltraSonic.
 - Application Layer.
 1. Steering.
 2. RobotControl.

1. System Description

- Description for whole system.
 1. Robot has four DC motors.
 2. Robot can move in the four Directions:
 - Forward.
 - Backward.
 - Right.
 - Left.
 3. Robot has character LCD to display the moving direction.
 4. Ultrasonic sensor need to add to Robot to detect objects.
 5. If there were no object detected, the Robot moves forward with 80% of its max speed.
 6. If there is an object detected 50 cm distant from the Robot then the Robot should slow down to 30% of its maximum speed.
 7. If there is an object detected 30 cm distant from the Robot then the Robot will stop then turn right and continue moving under the same distance and speed criteria.
 8. If there is an object detected less than 30 cm distant from the Robot then the Robot will stop then moves backward until the distance is 30 cm then stop and turn right and continue moving under the same distance and speed criteria.
- System Architectural Pattern is Monolithic system.
- System Constrains: Not defined.

2. Layered Architecture.

- Layered Architecture diagram



- Layers description and its components.
 1. MCAL Layer: Contains device drivers components for MCU hardware which directly access the MCU registers.
 - ICU.
 - DIO.
 - PWM.
 2. On-Board Layer: Contains device drivers components for the on board hardware (All hardware modules except MCU).
 - UltraSonic.
 - LCD.
 - DCMotorDriver.
 3. Application Layer: Contains the application components which are totally separated from the hardware.
 - RobotControl.
 - Steering.
- Layer types: All layers are closed.

3. SW Data Type Tables.

Name	uint8_t
Type	unsigned char
Range	0:255
Description	

Name	uint16_t
Type	unsigned short
Range	0:65,535
Description	

Name	uint32_t
Type	unsigned int
Range	0:4,294,967,295
Description	

4. SW Layers.

1. MCAL Layer.

- PWM.
 - Description: This module controls the pulse width modulation hardware.
 - Constrains:
 - HW: N/A.
 - SW: N/A.
 - APIs Data Types:

Name	channelNumber
Type	uint8_t
Range	0:4
Description	Specify PWM channel number.

Name	dutyCycle
Type	uint8_t
Range	0:100
Description	Specify PWM duty cycle.

- Configuration Data Types:

Name	PWM_cfg_s			
Type	Structure			
Elements	channelNumber	uint8_t	0:4	Description: Specify PWM channel number.
	dutyCycle	uint8_t	0:100	Description: Specify PWM duty cycle percentage.
	periodInUSec	uint16_t	0:5000	Description: Specify PWM period in micro seconds.
Configuration time	Linking			
Description	PWM configuration structure.			

○ API Tables:

Function name	PWM_Init		
Arguments	Input	a1: PWMConfig	Type: PWM_cfg_s
	Output	N/A	
	Input/Output	N/A	
Return	ERROR_OK	0	
	ERROR_NOK	1	
Synchronous	Yes		
Reentrant	No		
Description	This function initializes the PWM hardware channels.		

Function name	PWM_Start			
Arguments	Input	a1: channelNumber	Type: uint8_t	
	Output	N/A		
	Input/Output	N/A		
Return	ERROR_OK	0		
	ERROR_NOK	1		
Synchronous	Yes			
Reentrant	No			
Description	This function starts the PWM hardware channels.			

Function name	PWM_Stop			
Arguments	Input	a1: channelNumber	Type: uint8_t	
	Output	N/A		
	Input/Output	N/A		
Return	ERROR_OK	0		
	ERROR_NOK	1		
Synchronous	yes			
Reentrant	no			
Description	This function stops the PWM hardware channels.			

Function name	PWM_UpdateDutyCycle		
Arguments	Input	a1: channelNumber	Type: uint8_t
		a2: dutyCycle	Type: uint8_t
	Output	N/A	
	Input/Output	N/A	
Return	ERROR_OK	0	
	ERROR_NOK	1	
Synchronous	yes		
Reentrant	no		
Description	This function updates the PWM hardware channels duty cycle.		

- Sequence Diagram:
 - DCMotorDriver_Init calls PWM_Init
 - DCMotorDriver_MotorMove calls PWM_Start, PWM_Stop, and PWM_UpdateDutyCycle
- DIO.
 - Description: This module controls the Digital Input/Output hardware.
 - Constrains:
 - HW: N/A.
 - SW: N/A.
 - APIs Data Types:

Name	DIO_Port_e		
Type	Enumeration		
Range	DIO_PORT_A	0	Description: DIO port A
	DIO_PORT_B	1	Description: DIO port B
	DIO_PORT_C	2	Description: DIO port C
	DIO_PORT_D	3	Description: DIO port D
Description	Specify DIO port number.		

Name	DIO_Pin_e		
Type	Enumeration		
Range	DIO_PIN0	0x01	Description:

			DIO pin 0
	DIO_PIN1	0x02	Description: DIO pin 1
	DIO_PIN2	0x04	Description: DIO pin 2
	DIO_PIN3	0x08	Description: DIO pin 3
	DIO_PIN4	0x10	Description: DIO pin 4
	DIO_PIN5	0x20	Description: DIO pin 5
	DIO_PIN6	0x40	Description: DIO pin 6
	DIO_PIN7	0x80	Description: DIO pin 7
Description		Specify DIO port pin number.	

Name	DIO_Pin_State_e		
Type	Enumeration		
Range	LOW_STATE	0	Description: Low state signal
	HIGH_STATE	1	Description: High state signal
Description		Specify DIO pin Input/Output state.	

Name	DIO_Direction_e		
Type	Enumeration		
Range	OUTPUT	0	Description: Output pin
	INPUT	1	Description: Input pin
Description		Specify DIO pin Input/Output detection.	

○ Configuration Data Types:

Name	DIO_cfg_s		
Type	Structure		
Elements	portNumber	Type:	Description:

		DIO_Port_e	Specify DIO port number.
	pinNumber	Type: DIO_Pin_e	Description: Specify DIO pin number.
	direction	Type: DIO_Direction_e	Description: Specify DIO pin direction.
Configuration time	Linking		
Description	DIO configuration structure.		

○ API Tables:

Function name	DIO_Init			
Arguments	Input	a1: DIOConfig	Type: DIO_cfg_s	
	Output	N/A		
	Input/Output	N/A		
Return	ERROR_OK	0		
	ERROR_NOK	1		
Synchronous	yes			
Reentrant	no			
Description	This function initializes the DIO hardware.			

Function name	DIO_WritePin				
Arguments	Input	a1: portNumber	Type: DIO_Port_e		
		a2: pinNumber	Type: DIO_Pin_e		
		a3: outputState	Type: DIO_Pin_State_e		
	Output	N/A			
	Input/Output	N/A			
Return	ERROR_OK	0			
	ERROR_NOK	1			
Synchronous	yes				
Reentrant	no				
Description	This function writes a state value to the DIO				

	hardware output pin.
--	----------------------

Function name	DIO_ReadPin			
Arguments	Input	a1: portNumber	Type: DIO_Port_e	
		a2: pinNumber	Type: DIO_Pin_e	
	Output	a1: inputState	Type: DIO_Pin_State_e	
	Input/Output	N/A		
Return	ERROR_OK	0		
	ERROR_NOK	1		
Synchronous	yes			
Reentrant	no			
Description	This function reads a state value from the DIO hardware input pin.			

- Sequence Diagram:
 - LCD_Init calls DIO_Init
 - LCD_Write calls DIO_WritePin
 - UltraSonic_Init calls DIO_Init
 - UltraSonic_Trigger calls DIO_WritePin
- ICU.
 - Description: This module controls the ICU hardware.
 - Constrains:
 - HW: N/A.
 - SW: ICU_SetCallBackFunction API must be called after initialization if the ICU callback function is not passed in the ICU configuration structure to the ICU_Init API; ICU_SetCallBackFunction API must be called in ICU ISR.
 - APIs Data Types:

Name	ICU_Edge_Type_e		
Type	Enumeration		
Range	FAILING	0	Description: Failing edge
	RISING	1	Description: Rising edge
Description	Specify ICU input signal edge type.		

Name	ICU_CallBackFuncPtr
Type	Typdef void (*)(void)
Range	N/A
Description	Specify ICU callback function pointer.

Name	channelNumber
Type	uint8_t
Range	0:4
Description	Specify ICU channel number.

Name	capturedTimeInUsec
Type	uint32_t
Range	N/A
Description	Specify ICU time in micro seconds.

○ Configuration Data Types:

Name	ICU_cfg_s			
Type	Structure			
Elements	channelNumber	uint8_t	0:4	Description: Specify ICU channel number.
	edgeType	Type: ICU_Edge_Type_e	-	Description: Specify ICU edge type.
	callBackFuncPtr	ICU_CallBackFuncPtr	-	Description: Specify the callback function pointer for the specified ICU channel.
Configuration time	Linking			
Description	ICU configuration structure.			

○ API Tables:

Function name	ICU_Init			
Arguments	Input	a1: ICUConfig	Type: ICU_cfg_s	
	Output	N/A		
	Input/Output	N/A		
Return	ERROR_OK	0		
	ERROR_NOK	1		
Synchronous	Yes			
Reentrant	No			
Description	This function initializes the ICU hardware.			

Function name	ICU_SetEdgeType					
Arguments	Input	a1: channelNumber	Type: uint8_t	0:4		
		a2: edgeType	Type: ICU_Edge_Type_e	-		
	Output	N/A				
	Input/Output	N/A				
Return	ERROR_OK		0			
	ERROR_NOK		1			
Synchronous	Yes					
Reentrant	No					
Description	This function sets input signal edge type.					

Function name	ICU_SetCallbackFunction					
Arguments	Input	a1: channelNumber	Type: uint8_t	0:4		
		a2: callBackFuncPtr	Type: ICU_CallBackFuncPtr	-		
	Output	N/A				
	Input/Output	N/A				
Return	ERROR_OK	0				
	ERROR_NOK	1				
Synchronous	Yes					
Reentrant	No					
Description	This function sets the callback function pointer for the specified ICU channel.					

Function name	ICU_GetEdgeTimeInUsec				
Arguments	Input	a1: channelNumber	Type: uint8_t	0:4	
	Output	a1: capturedTimeInUsec		Type: uint32_t	
	Input/Output	N/A			
Return	ERROR_OK	0			
	ERROR_NOK	1			
Synchronous	Yes				
Reentrant	No				
Description	This function gets the time in micro second at which the ICU captured the configured edge.				

- Sequence Diagram:
 - UltraSonic_Init calls ICU_Init, and ICU_SetCallbackFunction
 - UltraSonic_CalculateDistance calls ICU_SetEdgeType, and ICU_GetEdgeTimeInUsec

2. On-Board Layer.

- UltraSonic.
 - Description: This module controls the Digital UltraSonic hardware.
 - Constrains:
 - HW: N/A.
 - SW: UltraSonic_CalculateDistance API must not be called by the upper layers APIs; it must be configured to be a callback function for the ICU module in MCAL layer; UltraSonic_CalculateDistance API is a periodic function performs a sequence of jobs depending on a state machine.
 - APIs Data Types:

Name	measuredDistanceInCm
Type	uint8_t
Range	0:100
Description	Specify UltraSonic measured distance in cm.

- Configuration Data Types: N/A.
- API Tables:

Function name	UltraSonic_Init	
Arguments	Input	N/A
	Output	N/A
	Input/Output	N/A
Return	ERROR_OK	0
	ERROR_NOK	1
Synchronous	Yes	
Reentrant	No	
Description	This function initializes the UltraSonic hardware.	

Function name	UltraSonic_Trigger		
Arguments	Input	N/A	
	Output	N/A	
	Input/Output	N/A	
Return	ERROR_OK	0	
	ERROR_NOK	1	
Synchronous	Yes		
Reentrant	No		
Description	This function triggers the UltraSonic hardware.		

Function name	UltraSonic_CalculateDistance	
Arguments	Input	N/A
	Output	N/A
	Input/Output	N/A
Return	ERROR_OK	0
	ERROR_NOK	1
Synchronous	no	
Reentrant	no	
Description	This function calculates the distance measured by the UltraSonic hardware.	

Function name	UltraSonic_GetMeasuredDistance			
Arguments	Input	N/A		
	Output	a1: measuredDistanceInCm	Type: uint8_t	0:100
	Input/Output	N/A		
Return	ERROR_OK	0		

	ERROR_NOK	1	
Synchronous	Yes		
Reentrant	No		
Description	This function returns the distance measured by the UltraSonic hardware.		

- Sequence Diagram:
 - RobotControl_Init calls UltraSonic_Init
 - RobotControl_MoveUpdate calls UltraSonic_GetMeasuredDistance
- LCD.
 - Description: This module controls the LCD hardware.
 - Constrains:
 - HW: N/A.
 - SW: LCD_Init and LCD_Write APIs is a periodic function performs a sequence of jobs depending on a state machine.
- APIs Data Types:

Name	LCD_Init_State_e		
Type	Enumeration		
Range	NOT_INITIALIZED	0	Description: LCD is not initialized
	INITIALIZED	1	Description: LCD is initialized
Description	Specify LCD initialization state.		

Name	LCD_Write_State_e		
Type	Enumeration		
Range	WRITE_NOT_FINISHED	0	Description: LCD write operation is not finished
	WRITE_FINISHED	1	Description: LCD write operation is finished
Description	Specify LCD write operation state.		

Name	LCD_Data_Command_Selection_e		
Type	Enumeration		
Range	COMMAND_WRITE	0	Description: Write command to LCD
	DATA_WRITE	1	Description: Write data to LCD
Description	Specify LCD writing selection.		

Name	LCD_Config_Bit_Mode_e		
Type	Enumeration		
Range	DATA_PINS_4	0	Description: Write operation to LCD occurs using 4 pins only
	DATA_PINS_8	1	Description: Write operation to LCD occurs using 8 pins
Description	Specify LCD writing selection.		

Name	LCD_State_e		
Type	Enumeration		
Range	FREE	0	Description: LCD is free to write new data
	BUSY	1	Description: Write operation to LCD not finished yet
Description	Specify LCD writing selection.		

○ Configuration Data Types:

Name	LCD_cfg_s		
Type	Structure		
Elements	configBitMode	Type: LCD_Config_Bit_Mode_e	Description: Specify number of pins used in writing data to the LCD.
Configuration time	Linking		
Description	LCD configuration structure.		

○ API Tables:

Function name	LCD_Init			
Arguments LCD_Init_State_e	Input	a1: LCDConfig	Type: LCD_cfg_s	
	Output	N/A		
	Input/Output	N/A		
Return	LCD_Init_State_e			
Synchronous	Yes			
Reentrant	No			
Description	This function initializes the LCD hardware.			

Function name	LCD_Write	
Arguments	Input	N/A
	Output	N/A
	Input/Output	N/A
Return	LCD_Write_State_e	
Synchronous	Yes	
Reentrant	No	
Description	This function writes a byte (character or command) to the LCD hardware.	

○ Sequence Diagram:

- RobotControl_Init calls LCD_Init
- RobotControl_MoveUpdate calls LCD_Write

- DCMotorDriver.
 - Description: This module controls the DCMotorDriver hardware which controls the speed and direction of DC motors.
 - Constrains:
 - HW: N/A.
 - SW: N/A.
 - APIs Data Types:

Name	channelNumber
Type	uint8_t
Range	0:3
Description	Specify DCMotorDriver channel number.

Name	DCMotorDriver_Direction_e		
Type	Enumeration		
Range	CLOCKWISE_DIRECTION	0	Description: Moving clockwise
	COUNTER_CLOCKWISE_DIRECTION	1	Description: Moving counter clockwise
Description	Specify DC motor direction.		

Name	Speed
Type	uint8_t
Range	0:100
Description	Specify motor speed.

- Configuration Data Types:
- API Tables:

Function name	DCMotorDriver_Init	
Arguments	Input	N/A
	Output	N/A
	Input/Output	N/A
Return	ERROR_OK	0
	ERROR_NOK	1
Synchronous	yes	

Reentrant	no
Description	This function initializes DCMotor driver which controls the speed and direction of DC motors.

Function name	DCMotorDriver_MotorMove				
Arguments	Input	a1: channelNumber	Type: uint8_t		
		a2: direction	Type: DCMotorDriver_Direction_e		
		a3: speed	Type: uint8_t		
	Output	N/A			
	Input/Output	N/A			
Return	ERROR_OK	0			
	ERROR_NOK	1			
Synchronous	yes				
Reentrant	no				
Description	This function controls the speed and direction of DC motors.				

- Sequence Diagram:
 - Steering_Init calls DCMotorDriver_Init
 - Steering_Command calls DCMotorDriver_MotorMove

3. Application Layer.

- Steering.
 - Description: This module sends motor control information to the DC motor hardware.
 - Constrains:
 - HW: N/A.
 - SW: N/A.
- APIs Data Types:

Name	Steering_MovingState_e		
Type	Enumeration		
Range	STOP_MOVE	0	Description: Stop moving
	START_MOVE	1	Description: Start moving
Description	Specify DCMotor Steering start or stop		

	state.
--	--------

Name	Steering_Direction_e		
Type	Enumeration		
Range	FORWARD_DIRECTION	0	Description: Moving forward
	BACKWARD_DIRECTION	1	Description: Moving backward
	RIGHT_DIRECTION	2	Moving right
	LEFT_DIRECTION	3	Moving left
Description	Specify DCMotor Steering direction.		

Name	Speed
Type	uint8_t
Range	0:100
Description	Specify DCMotor Steering speed.

- Configuration Data Types: N/A.
- API Tables:

Function name	Steering_Init	
Arguments	Input	N/A
	Output	N/A
	Input/Output	N/A
Return	ERROR_OK	0
	ERROR_NOK	1
Synchronous	Yes	
Reentrant	No	
Description	This function initializes Steering control module.	

Function name	Steering_Command		
Arguments	Input	a1: movingState	Type: Steering_MovingState_e
		a2: direction	Type: Steering_Direction_e

		a3: speed	Type: uint8_t	
	Output	N/A		
	Input/Output	N/A		
Return	ERROR_OK	0		
	ERROR_NOK	1		
Synchronous	Yes			
Reentrant	No			
Description	This function sends Steering command to the motor hardware.			

- Sequence Diagram:
 - RobotControl_Init calls Steering_Init
 - RobotControl_MoveUpdate calls Steering_Command
- RobotControl.
 - Description: This module controls the robot movement.
 - Constrains:
 - HW: N/A.
 - SW: RobotControl_MoveUpdate API performs a sequence of jobs depending on a state machine.
 - APIs Data Types.

Name	Robot_Control_State_e		
Type	Enumeration		
Range	ROBOT_INIT	0	Description: Initialize robot movement
	ROBOT_MOVE_FORWARD	1	Description: Move robot forward
	ROBOT_MOVE_BACKWARD	2	Description: Move robot backward
	ROBOT_MOVE_RIGHT	3	Description: Move robot right
	ROBOT_MOVE_LEFT	4	Description: Move robot left
Description	Specify robot movement state.		

- Configuration Data Types: N/A.
- API Tables:

Function name	RobotControl_Init	
Arguments	Input	N/A
	Output	N/A
	Input/Output	N/A
Return	ERROR_OK	0
	ERROR_NOK	1
Synchronous	Yes	
Reentrant	No	
Description	This function initializes the robot system.	

Function name	RobotControl_MoveUpdate	
Arguments	Input	N/A
	Output	N/A
	Input/Output	N/A
Return	ERROR_OK	0
	ERROR_NOK	1
Synchronous	Yes	
Reentrant	No	
Description	This function controls the robot system	

- Sequence Diagram: N/A.