RTOS Project

AUTHOR: Ahmed Sayed
Alaa Youssif

Version V 2.1

My Project

Brief

This file contains schleduler Tasks of system which user can chose to display shape by LEDs or Write a String by LCD Based on FreeRTOS

Details

Four Tasks are explained in the SoftwareContext

SoftwareContext

This service initialize MCU,PORT,DIO,Dispatcher then start a round Robin between MangerTask, ActionTask, Led_Shape_Task and LCD_String_Task.

Creation

Creation	
Name	Ahmed Sayed - Alaa Youssif
Date	13/9/2023
File	RTOS System Project
Version	V2.1

Refrences

Refrences	
First refrence	Reference manual STM32F103xx
Second refrence	Programming manual Cortex®-M3
Third refrence	freeRTOS.org
Fourth refrence	plantuml.com

File Index

File List

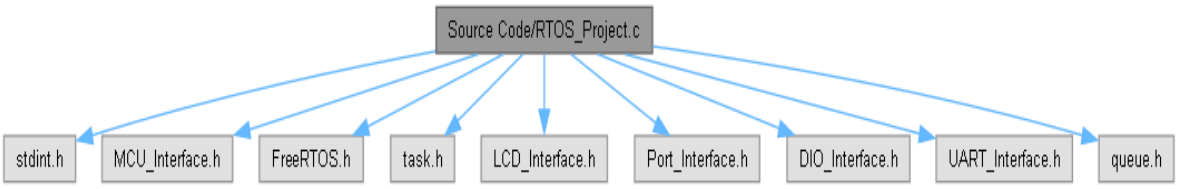
Here is a list of all files with brief descriptions:

File Documentation

Source Code/RTOS_Project.c File Reference

```
#include <stdint.h>
#include "MCU_Interface.h"
#include "FreeRTOS.h"
#include "task.h"
#include "LCD_Interface.h"
#include "Port_Interface.h"
#include "DIO_Interface.h"
#include "UART_Interface.h"
#include "queue.h"
```

Include dependency graph for RTOS_Project.c:



Functions

- void MangmentTask_Func (void *pvParameters)
- void ActionTask_Func (void *pvParameters)
- void LCDTask_Func (void *pvParameters)
- void LEDTask_Func (void *pvParameters)
- void vTaskStartScheduler (void)

Variables

- TaskHandle_t ManageHandler
- TaskHandle_t ActionHandler
- TaskHandle_t LCDHandler
- TaskHandle_t LEDHandler
- QueueHandle_t MainQueue

Function Documentation

void ActionTask_Func (void * pvParameters)

Service_Name

ActionTask_Func

Service_ID_hex

0x02

Sync_Async

Synchronous

Renterancy

Reentrant

Parameters_in

Parameters

in	void	*pvParameters—>A value that is passed as the paramater to the created task.
out		None
in-out		None

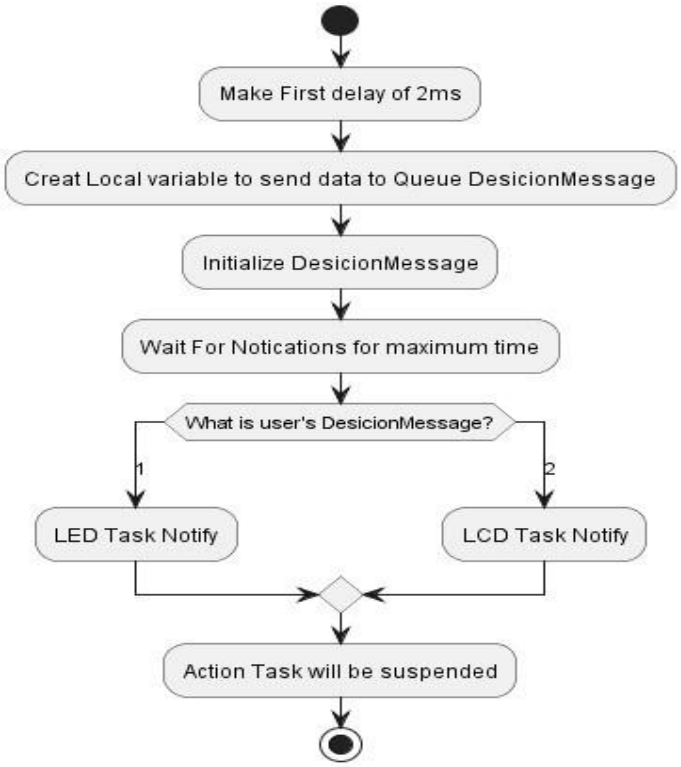
Return_value

None

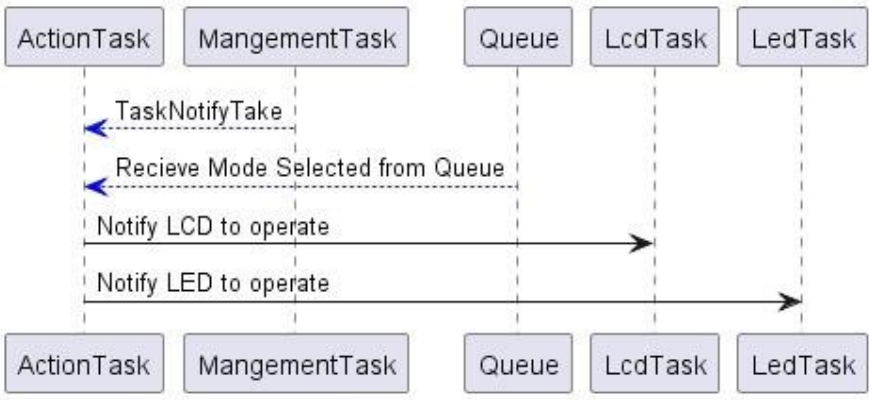
Description

It receives decision messages from a queue and performing specific tasks based on the received message &notifying either the LEDHandler or LCDHandler tasks.

Activity_diagram



Sequence_diagram



Here is the caller graph for this function:

void LCDTask_Func (void * pvParameters)

Service_Name

LCDTask_Func

Service_ID_hex

0x04

Sync_Async

Synchronous

Renterancy

Reentrant

Parameters_in

Parameters

in	void ptr	*pvParameters—>A value that is passed as the paramater to the created task.
out		None
in-out		None

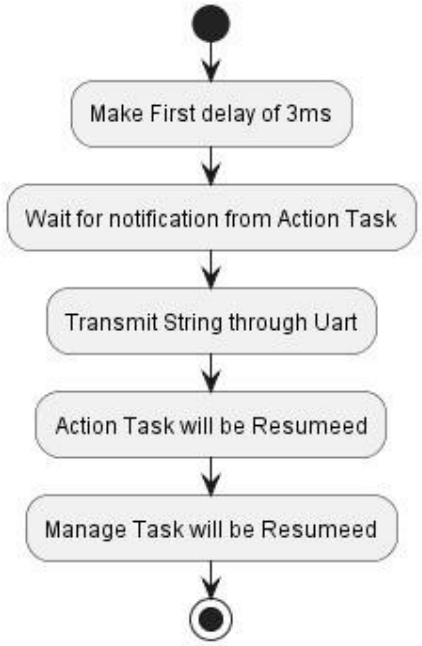
Return_value

None

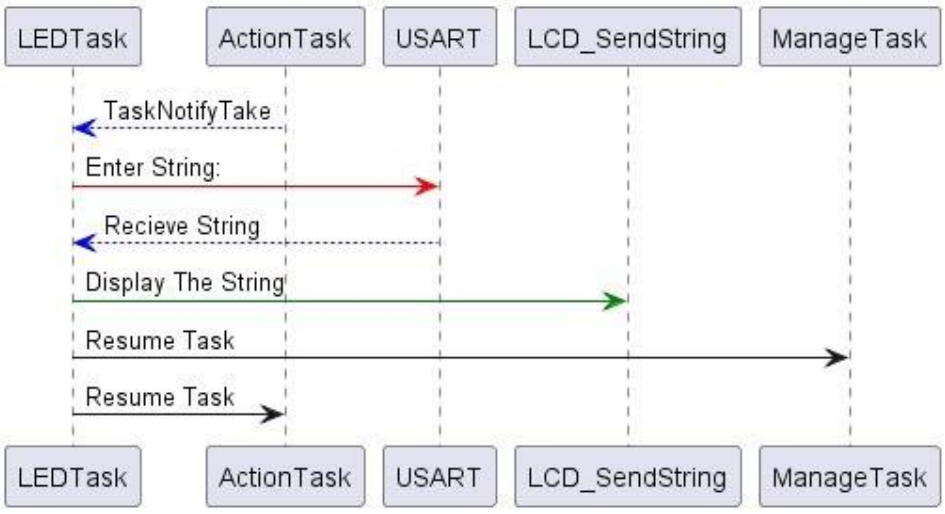
Description

This function, LCDTask_Func,Take User's choice to LCD operate

Activity_diagram



Sequence_diagram



Here is the caller graph for this function:

void LEDTask_Func (void * pvParameters)

Service_Name

LEDTask_Func

Service_ID_hex

0x03

Sync_Async

Synchronous

Renterancy

Reentrant

Parameters_in

Parameters

in	void ptr	*pvParameters—>A value that is passed as the paramater to the created task.
out		None
in-out		None

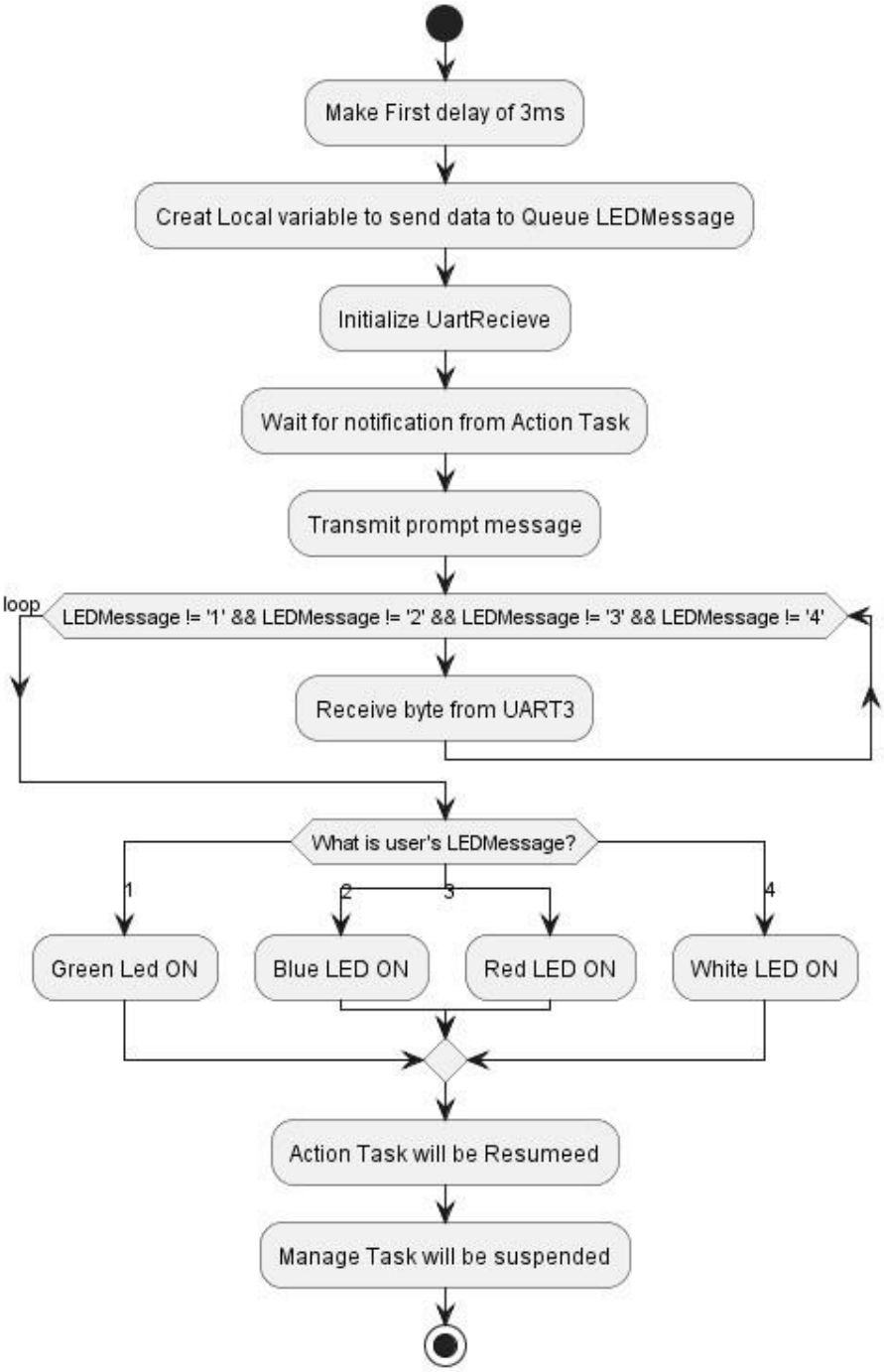
Return_value

None

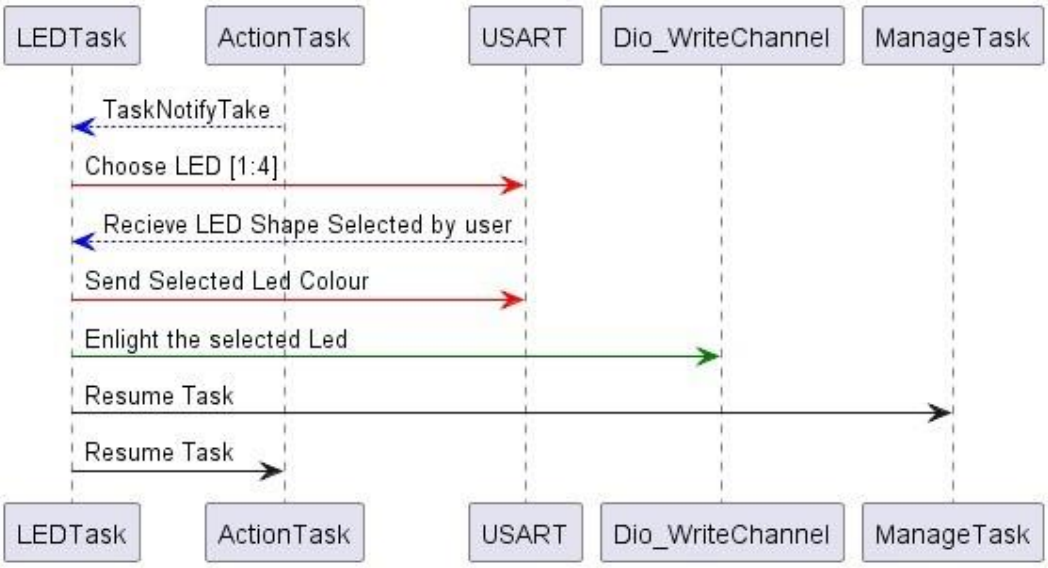
Description

This function, LEDTask_Func,Take User's choice Led to Led operate

Activity_diagram



Sequence_diagram



Here is the caller graph for this function:

void MangmentTask_Func (void * pvParameters)

Service_Name

MangmentTask_Func

Service_ID_hex

0x01

Sync_Async

Synchronous

Renterancy

Reentrant

Parameters_in

Parameters

in	void ptr	*pvParameters—>A value that is passed as the paramater to the created task.
out		None
in-out		None

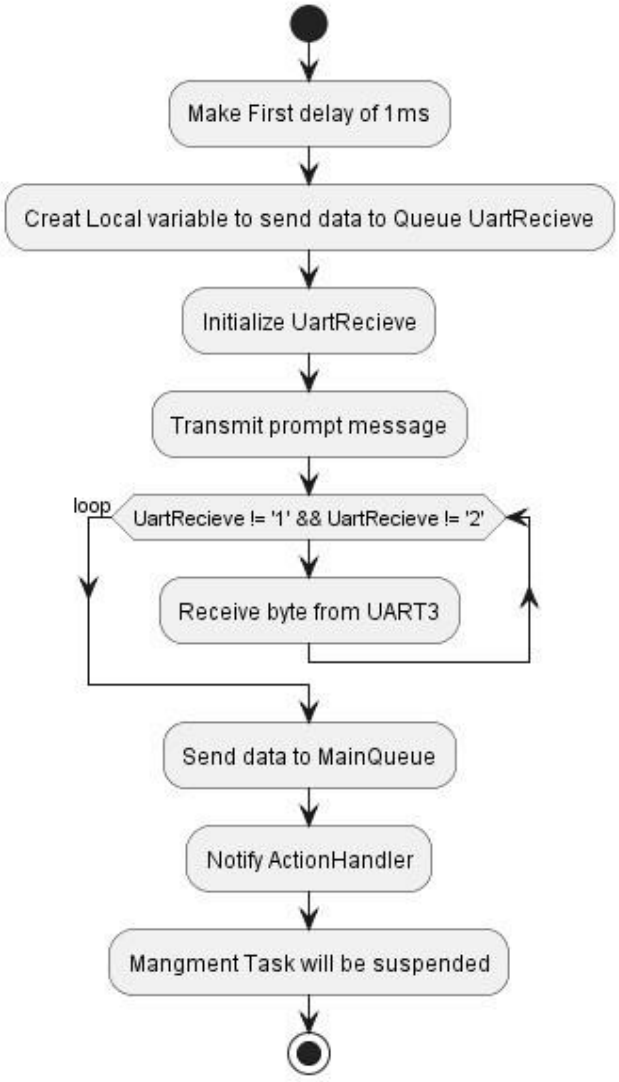
Return_value

None

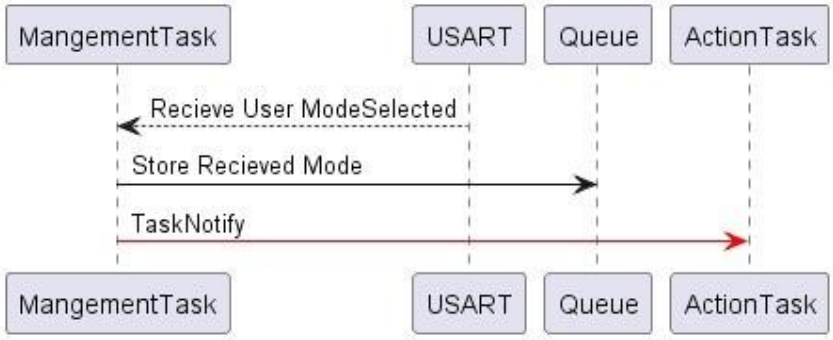
Description

The MangmentTask_Func function is a task function that prompts the user to choose between LED Mode and LCD Mode, sends the chosen option to a queue, and notifies the Action Task to start processing. The function then suspends itself and runs indefinitely until interrupted by an external event, such as a notification from the Action Task.

Activity_diagram



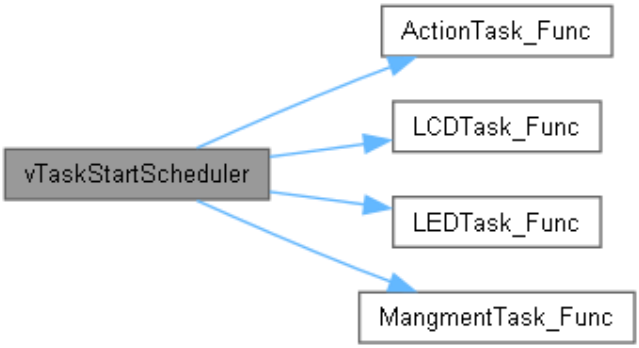
Sequence_diagram



Here is the caller graph for this function:

void vTaskStartScheduler (void)

Here is the call graph for this function:



Variable Documentation

TaskHandle_t ActionHandler

TaskHandle_t LCDHandler

TaskHandle_t LEDHandler

QueueHandle_t MainQueue

TaskHandle_t ManageHandler