

Microproject

Generated by Doxygen 1.9.1

1 File Index	1
1.1 File List	1
2 File Documentation	3
2.1 button.c File Reference	3
2.1.1 Detailed Description	3
2.1.2 Function Documentation	3
2.1.2.1 Button_Init()	3
2.2 button.h File Reference	4
2.2.1 Detailed Description	4
2.2.2 Function Documentation	4
2.2.2.1 Button_Init()	4
2.3 doors.c File Reference	5
2.3.1 Detailed Description	5
2.3.2 Function Documentation	5
2.3.2.1 CloseDoors()	5
2.3.2.2 Doors_Init()	6
2.3.2.3 initDoorsPWM()	6
2.3.2.4 OpenDoors()	6
2.3.2.5 Servo_PWM_Angle()	8
2.4 doors.h File Reference	8
2.4.1 Detailed Description	9
2.4.2 Function Documentation	9
2.4.2.1 CloseDoors()	9
2.4.2.2 Doors_Init()	9
2.4.2.3 initDoorsPWM()	10
2.4.2.4 OpenDoors()	10
2.4.2.5 Servo_PWM_Angle()	10
2.5 gpio.c File Reference	11
2.5.1 Detailed Description	11
2.5.2 Function Documentation	11
2.5.2.1 MX_GPIO_Init()	11
2.6 gpio.h File Reference	12
2.6.1 Detailed Description	12
2.6.2 Function Documentation	12
2.6.2.1 MX_GPIO_Init()	12
2.7 led.c File Reference	13
2.7.1 Detailed Description	13
2.7.2 Function Documentation	14
2.7.2.1 FirstFloor()	14
2.7.2.2 GroundFloor()	14
2.7.2.3 initLEDGPIO()	14

2.7.2.4 initLEDsPWM()	15
2.7.2.5 LED_Init()	15
2.7.2.6 LED_PWM_Percent()	15
2.7.2.7 Moving()	16
2.7.2.8 TurnOffLEDPWM()	16
2.8 led.h File Reference	16
2.8.1 Detailed Description	17
2.8.2 Function Documentation	17
2.8.2.1 FirstFloor()	17
2.8.2.2 GroundFloor()	18
2.8.2.3 initLEDGPIO()	18
2.8.2.4 initLEDsPWM()	18
2.8.2.5 LED_Init()	19
2.8.2.6 LED_PWM_Percent()	19
2.8.2.7 Moving()	20
2.8.2.8 TurnOffLEDPWM()	20
2.9 lift.c File Reference	20
2.9.1 Detailed Description	21
2.9.2 Function Documentation	21
2.9.2.1 lift_Init()	21
2.9.2.2 liftDown()	22
2.9.2.3 liftFloor()	22
2.9.2.4 liftIsMoving()	22
2.9.2.5 liftStop()	23
2.9.2.6 liftUp()	23
2.10 lift.h File Reference	23
2.10.1 Detailed Description	24
2.10.2 Function Documentation	24
2.10.2.1 lift_Init()	24
2.10.2.2 liftDown()	24
2.10.2.3 liftFloor()	25
2.10.2.4 liftIsMoving()	25
2.10.2.5 liftStop()	25
2.10.2.6 liftUp()	27
2.11 main.c File Reference	27
2.11.1 Detailed Description	28
2.11.2 Function Documentation	28
2.11.2.1 Error_Handler()	28
2.11.2.2 HAL_GPIO_EXTI_Callback()	28
2.11.2.3 HAL_TIM_PeriodElapsedCallback()	29
2.11.2.4 main()	29
2.11.2.5 SystemClock_Config()	29

2.12 main.h File Reference	30
2.12.1 Detailed Description	31
2.12.2 Function Documentation	31
2.12.2.1 Error_Handler()	31
2.13 motor.c File Reference	32
2.13.1 Detailed Description	32
2.13.2 Function Documentation	32
2.13.2.1 Motor_Init()	32
2.14 motor.h File Reference	33
2.14.1 Detailed Description	33
2.14.2 Function Documentation	33
2.14.2.1 Motor_Init()	33
2.15 SysTick_Handler.c File Reference	34
2.15.1 Detailed Description	34
2.15.2 Function Documentation	34
2.15.2.1 SysTick_Handler()	34
2.16 timer.c File Reference	34
2.16.1 Detailed Description	35
2.16.2 Function Documentation	35
2.16.2.1 TIM1_Init()	35
2.16.2.2 TIM3_Init()	36
2.16.2.3 TIM4_Init()	36
2.17 timer.h File Reference	36
2.17.1 Detailed Description	37
2.17.2 Function Documentation	37
2.17.2.1 TIM1_Init()	37
2.17.2.2 TIM3_Init()	38
2.17.2.3 TIM4_Init()	38
2.18 uart.c File Reference	38
2.18.1 Detailed Description	39
2.18.2 Function Documentation	39
2.18.2.1 USART2_Init()	39
2.19 uart.h File Reference	39
2.19.1 Detailed Description	40
2.19.2 Function Documentation	40
2.19.2.1 USART2_Init()	40

Index

41

Chapter 1

File Index

1.1 File List

Here is a list of all documented files with brief descriptions:

button.c	Button Functions	3
button.h	This file contains all the function prototypes for the button.c file	4
doors.c	Door Functions	5
doors.h	This file contains all the function prototypes for the doors.c file	8
gpio.c	This file provides code for the configuration of all used GPIO pins	11
gpio.h	This file contains all the function prototypes for the gpio.c file	12
led.c	Led Functions	13
led.h	This file contains all the function prototypes for the led.c file	16
lift.c	Lift Functions	20
lift.h	This file contains all the function prototypes for the lift.c file	23
main.c	Main File	27
main.h	: Header for main.c file. This file contains the common defines of the application	30
motor.c	Motor Functions	32
motor.h	This file contains all the function prototypes for the motor.c file	33
SysTick_Handler.c	SysTick Handler. This function, along with other interrupt handlers is defined at <code>stm32f4xx_it.c</code> . In order to address it, we have created this file. This file is not included on the project	34
timer.c	Timer(s) Initialization	34
timer.h	This file contains all the function prototypes for the timer.c file	36

uart.c	UART implementation	38
uart.h	This file contains all the function prototypes for the uart.c file	39

Chapter 2

File Documentation

2.1 button.c File Reference

Button Functions.

```
#include "button.h"
```

Functions

- void [Button_Init](#) (void)
Initialize the button hardware.

2.1.1 Detailed Description

Button Functions.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.1.2 Function Documentation

2.1.2.1 Button_Init()

```
void Button_Init (  
    void )
```

Initialize the button hardware.

Parameters

<code>void</code>	
-------------------	--

Returns

`void`

2.2 button.h File Reference

This file contains all the function prototypes for the [button.c](#) file.

```
#include "main.h"
```

Functions

- void [Button_Init](#) (void)
Initialize the button hardware.

2.2.1 Detailed Description

This file contains all the function prototypes for the [button.c](#) file.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.2.2 Function Documentation

2.2.2.1 Button_Init()

```
void Button_Init (  
    void )
```

Initialize the button hardware.

Parameters

<code>void</code>	
-------------------	--

Returns

void

2.3 doors.c File Reference

Door Functions.

```
#include "doors.h"
#include "timer.h"
```

Functions

- void [initDoorsPWM](#) (void)
Initialize PWM Channels 1&2 of TIM1.
- void [Servo_PWM_Angle](#) (uint16_t Servo, uint8_t angle)
Modify the servo angle by setting the value of the CCRx register.
- void [OpenDoors](#) (void)
Set servo angle to open position.
- void [CloseDoors](#) (void)
Set servo angle to closed position.
- void [Doors_Init](#) ()
Initialize the door hardware and set the initial angle to "Open".

2.3.1 Detailed Description

Door Functions.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.3.2 Function Documentation

2.3.2.1 CloseDoors()

```
void CloseDoors (
    void )
```

Set servo angle to closed position.

Parameters

<i>void</i>	
-------------	--

Returns

void

2.3.2.2 Doors_Init()

```
void Doors_Init (
    void )
```

Initialize the door hardware and set the initial angle to "Open".

Parameters

<i>void</i>	
-------------	--

Returns

void

2.3.2.3 initDoorsPWM()

```
void initDoorsPWM (
    void )
```

Initialize PWM Channels 1&2 of TIM1.

Parameters

<i>void</i>	
-------------	--

Returns

void

2.3.2.4 OpenDoors()

```
void OpenDoors (
    void )
```

Set servo angle to open position.

Parameters

<code>void</code>	
-------------------	--

Returns

`void`

2.3.2.5 Servo_PWM_Angle()

```
void Servo_PWM_Angle (
    uint16_t Servo,
    uint8_t angle )
```

Modify the servo angle by setting the value of the CCRx register.

Parameters

<code>uint16_t</code>	Servo, uint8_t angle
-----------------------	----------------------

Returns

`void`

2.4 doors.h File Reference

This file contains all the function prototypes for the [doors.c](#) file.

```
#include "main.h"
```

Functions

- void [initDoorsPWM](#) (void)
Initialize PWM Channels 1&2 of TIM1.
- void [Servo_PWM_Angle](#) (uint16_t Servo, uint8_t angle)
Modify the servo angle by setting the value of the CCRx register.
- void [OpenDoors](#) (void)
Set servo angle to open position.
- void [CloseDoors](#) (void)
Set servo angle to closed position.
- void [Doors_Init](#) (void)
Initialize the door hardware and set the initial angle to "Open".

2.4.1 Detailed Description

This file contains all the function prototypes for the [doors.c](#) file.

Author

Alejandro López Rodríguez and Ana María Casanova López

Date

06/06/2021

2.4.2 Function Documentation

2.4.2.1 CloseDoors()

```
void CloseDoors (
    void )
```

Set servo angle to closed position.

Parameters

<i>void</i>	
-------------	--

Returns

void

2.4.2.2 Doors_Init()

```
void Doors_Init (
    void )
```

Initialize the door hardware and set the initial angle to "Open".

Parameters

<i>void</i>	
-------------	--

Returns

void

2.4.2.3 initDoorsPWM()

```
void initDoorsPWM (  
    void )
```

Initialize PWM Channels 1&2 of TIM1.

Parameters

<i>void</i>	
-------------	--

Returns

void

2.4.2.4 OpenDoors()

```
void OpenDoors (  
    void )
```

Set servo angle to open position.

Parameters

<i>void</i>	
-------------	--

Returns

void

2.4.2.5 Servo_PWM_Angle()

```
void Servo_PWM_Angle (  
    uint16_t Servo,  
    uint8_t angle )
```

Modify the servo angle by setting the value of the CCRx register.

Parameters

<i>uint16_t</i>	Servo, <i>uint8_t</i> angle
-----------------	-----------------------------

Returns

void

2.5 gpio.c File Reference

This file provides code for the configuration of all used GPIO pins.

```
#include "gpio.h"
```

Functions

- void [MX_GPIO_Init](#) (void)
GPIO Initialization (Non-used ones) -> Autogenerated (Non-modifiable)

2.5.1 Detailed Description

This file provides code for the configuration of all used GPIO pins.

Attention

© Copyright (c) 2021 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

2.5.2 Function Documentation

2.5.2.1 MX_GPIO_Init()

```
void MX_GPIO_Init (  
    void )
```

GPIO Initialization (Non-used ones) -> Autogenerated (Non-modifiable)

Parameters

<code>void</code>	
-------------------	--

Returns

None Configure pins as Analog Input Output EVENT_OUT EXTI Free pins are configured automatically as Analog (this feature is enabled through the Code Generation settings)

2.6 gpio.h File Reference

This file contains all the function prototypes for the [gpio.c](#) file.

```
#include "main.h"
```

Functions

- void [MX_GPIO_Init](#) (void)
GPIO Initialization (Non-used ones) -> Autogenerated (Non-modifiable)

2.6.1 Detailed Description

This file contains all the function prototypes for the [gpio.c](#) file.

Attention

© Copyright (c) 2021 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

2.6.2 Function Documentation

2.6.2.1 MX_GPIO_Init()

```
void MX_GPIO_Init (  
    void )
```

GPIO Initialization (Non-used ones) -> Autogenerated (Non-modifiable)

Parameters

<code>void</code>	
-------------------	--

Returns

None Configure pins as Analog Input Output EVENT_OUT EXTI Free pins are configured automatically as Analog (this feature is enabled through the Code Generation settings)

2.7 led.c File Reference

Led Functions.

```
#include "led.h"
#include "timer.h"
```

Functions

- void `initLEDGPIO` (void)
Configure the floor LEDs.
- void `initLEDsPWM` (void)
Initialize TIM4 PWMs.
- void `LED_PWM_Percent` (uint16_t LED, uint8_t value)
Control the duty cycle of the TIM4 PWM signal.
- void `TurnOffLEDPWM` (void)
Set TIM4 PWM duty cycle to 0 -> Turn off red and orange LEDs.
- void `Moving` (void)
Turn OFF green and blue LEDs to indicate that the lift is moving.
- void `GroundFloor` (void)
Turn ON blue LED to indicate that the lift is in the ground floor.
- void `FirstFloor` (void)
Turn ON green LED to indicate that the lift is in the first floor.
- void `LED_Init` (void)
Initialize the LED hardware.

2.7.1 Detailed Description

Led Functions.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.7.2 Function Documentation

2.7.2.1 FirstFloor()

```
void FirstFloor (
    void )
```

Turn ON green LED to indicate that the lift is in the first floor.

Parameters

None	
------	--

Returns

None

2.7.2.2 GroundFloor()

```
void GroundFloor (
    void )
```

Turn ON blue LED to indicate that the lift is in the ground floor.

Parameters

None	
------	--

Returns

None

2.7.2.3 initLEDGPIO()

```
void initLEDGPIO (
    void )
```

Configure the floor LEDs.

Parameters

void	
------	--

Returns

void

2.7.2.4 initLEDsPWM()

```
void initLEDsPWM (
    void )
```

Initialize TIM4 PWMs.

Parameters

None	
------	--

Returns

None

2.7.2.5 LED_Init()

```
void LED_Init (
    void )
```

Initialize the LED hardware.

Parameters

None	
------	--

Returns

None

2.7.2.6 LED_PWM_Percent()

```
void LED_PWM_Percent (
    uint16_t LED,
    uint8_t value )
```

Control the duty cycle of the TIM4 PWM signal.

Parameters

<code>uint16_t</code>	LED, uint8_t value
-----------------------	--------------------

Returns

None

2.7.2.7 Moving()

```
void Moving (
            void )
```

Turn OFF green and blue LEDs to indicate that the lift is moving.

Parameters

<i>None</i>	
-------------	--

Returns

None

2.7.2.8 TurnOffLEDPWM()

```
void TurnOffLEDPWM (
            void )
```

Set TIM4 PWM duty cycle to 0 -> Turn off red and orange LEDs.

Parameters

<i>None</i>	
-------------	--

Returns

None

2.8 led.h File Reference

This file contains all the function prototypes for the [led.c](#) file.

```
#include "main.h"
```

Functions

- void [initLEDGPIO](#) (void)
Configure the floor LEDs.
- void [initLEDsPWM](#) (void)
Initialize TIM4 PWMs.
- void [LED_PWM_Percent](#) (uint16_t LED, uint8_t value)
Control the duty cycle of the TIM4 PWM signal.
- void [TurnOffLEDPWM](#) (void)
Set TIM4 PWM duty cycle to 0 -> Turn off red and orange LEDs.
- void [Moving](#) (void)
Turn OFF green and blue LEDs to indicate that the lift is moving.
- void [GroundFloor](#) (void)
Turn ON blue LED to indicate that the lift is in the ground floor.
- void [FirstFloor](#) (void)
Turn ON green LED to indicate that the lift is in the first floor.
- void [LED_Init](#) (void)
Initialize the LED hardware.

2.8.1 Detailed Description

This file contains all the function prototypes for the [led.c](#) file.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.8.2 Function Documentation

2.8.2.1 FirstFloor()

```
void FirstFloor (
    void )
```

Turn ON green LED to indicate that the lift is in the first floor.

Parameters

<i>None</i>	
-------------	--

Returns

None

2.8.2.2 GroundFloor()

```
void GroundFloor (
    void )
```

Turn ON blue LED to indicate that the lift is in the ground floor.

Parameters

None	
------	--

Returns

None

2.8.2.3 initLEDGPIO()

```
void initLEDGPIO (
    void )
```

Configure the floor LEDs.

Parameters

void	
------	--

Returns

void

2.8.2.4 initLEDsPWM()

```
void initLEDsPWM (
    void )
```

Initialize TIM4 PWMs.

Parameters

<i>None</i>	
-------------	--

Returns

None

2.8.2.5 LED_Init()

```
void LED_Init (
    void )
```

Initialize the LED hardware.

Parameters

<i>None</i>	
-------------	--

Returns

None

2.8.2.6 LED_PWM_Percent()

```
void LED_PWM_Percent (
    uint16_t LED,
    uint8_t value )
```

Control the duty cycle of the TIM4 PWM signal.

Parameters

<i>uint16_t</i>	LED, <i>uint8_t</i> value
-----------------	---------------------------

Returns

None

2.8.2.7 Moving()

```
void Moving (
    void )
```

Turn OFF green and blue LEDs to indicate that the lift is moving.

Parameters

None	
------	--

Returns

None

2.8.2.8 TurnOffLEDPWM()

```
void TurnOffLEDPWM (
    void )
```

Set TIM4 PWM duty cycle to 0 -> Turn off red and orange LEDs.

Parameters

None	
------	--

Returns

None

2.9 lift.c File Reference

Lift Functions.

```
#include "main.h"
#include "timer.h"
#include "button.h"
#include "led.h"
#include "doors.h"
#include "motor.h"
#include "uart.h"
#include "string.h"
```

Functions

- void `lift_Init` (void)
Main Init -> Initialize the lift by calling the required sub-functions.
- uint32_t `liftIsMoving` (void)
Retrieve whether the lift is moving or stopped.
- int32_t `liftFloor` (void)
Retrieve the floor where the lift is.
- void `liftUp` (void)
Control the lift to go up and indicate it via the UART.
- void `liftDown` (void)
Control the lift to go down and indicate it via the UART.
- void `liftStop` (void)
Stop the lift, open the doors and send the current floor via UART.

2.9.1 Detailed Description

Lift Functions.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.9.2 Function Documentation

2.9.2.1 `lift_Init()`

```
void lift_Init (  
    void )
```

Main Init -> Initialize the lift by calling the required sub-functions.

Parameters

<code>void</code>	
-------------------	--

Returns

`void`

2.9.2.2 liftDown()

```
void liftDown (
    void )
```

Control the lift to go down and indicate it via the UART.

Parameters

<i>void</i>	
-------------	--

Returns

void

2.9.2.3 liftFloor()

```
int32_t liftFloor (
    void )
```

Retrieve the floor where the lift is.

Parameters

<i>void</i>	
-------------	--

Returns

0 ground floor, 1 first floor, -1 moving

2.9.2.4 liftIsMoving()

```
uint32_t liftIsMoving (
    void )
```

Retrieve whether the lift is moving or stopped.

Parameters

<i>void</i>	
-------------	--

Returns

1 if it is moving, 0 if it is stopped

2.9.2.5 liftStop()

```
void liftStop (
    void )
```

Stop the lift, open the doors and send the current floor via UART.

Parameters

void	
------	--

Returns

None

2.9.2.6 liftUp()

```
void liftUp (
    void )
```

Control the lift to go up and indicate it via the UART.

Parameters

void	
------	--

Returns

void

2.10 lift.h File Reference

This file contains all the function prototypes for the [lift.c](#) file.

```
#include "main.h"
```

Functions

- void [lift_Init](#) (void)
Main Init -> Initialize the lift by calling the required sub-functions.
- void [liftUp](#) (void)

- Control the lift to go up and indicate it via the UART.*
- void `liftDown` (void)
- Control the lift to go down and indicate it via the UART.*
- void `liftStop` (void)
- Stop the lift, open the doors and send the current floor via UART.*
- uint32_t `liftIsMoving` (void)
- Retrieve whether the lift is moving or stopped.*
- int32_t `liftFloor` (void)
- Retrieve the floor where the lift is.*

2.10.1 Detailed Description

This file contains all the function prototypes for the `lift.c` file.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.10.2 Function Documentation

2.10.2.1 lift_Init()

```
void lift_Init (
    void )
```

Main Init -> Initialize the lift by calling the required sub-functions.

Parameters

<code>void</code>	
-------------------	--

Returns

`void`

2.10.2.2 liftDown()

```
void liftDown (
    void )
```

Control the lift to go down and indicate it via the UART.

Parameters

<i>void</i>	
-------------	--

Returns

void

2.10.2.3 liftFloor()

```
int32_t liftFloor (  
    void )
```

Retrieve the floor where the lift is.

Parameters

<i>void</i>	
-------------	--

Returns

0 ground floor, 1 first floor, -1 moving

2.10.2.4 liftIsMoving()

```
uint32_t liftIsMoving (  
    void )
```

Retrieve whether the lift is moving or stopped.

Parameters

<i>void</i>	
-------------	--

Returns

1 if it is moving, 0 if it is stopped

2.10.2.5 liftStop()

```
void liftStop (  
    void )
```

Stop the lift, open the doors and send the current floor via UART.

Parameters

<code>void</code>	
-------------------	--

Returns

None

2.10.2.6 liftUp()

```
void liftUp (  
    void )
```

Control the lift to go up and indicate it via the UART.

Parameters

<code>void</code>	
-------------------	--

Returns

void

2.11 main.c File Reference

Main File.

```
#include "main.h"  
#include "gpio.h"  
#include "timer.h"  
#include "uart.h"  
#include "led.h"  
#include "button.h"  
#include "lift.h"  
#include "doors.h"
```

Functions

- void [SystemClock_Config](#) (void)
System Clock Configuration.
- int [main](#) (void)
The application entry point.
- void [HAL_GPIO_EXTI_Callback](#) (uint16_t GPIO_Pin)
EXTI Callback Function.
- void [HAL_TIM_PeriodElapsedCallback](#) (TIM_HandleTypeDef *htim)
TIM3 (5s) Callback Function.
- void [Error_Handler](#) (void)
This function is executed in case of error occurrence.

Variables

- volatile bool **state** = true
- bool **buttonPressFlag** = false
- bool **timer5sEndFlag** = false
- bool **timer100msEndFlag** = false
- uint8_t **step_positions** [8] = {0x08,0x0C,0x04,0x06,0x02,0x03,0x01,0x09}
- uint8_t **step_index** = 0
- NUMBER_FLOOR **current_floor** = GROUND_FLOOR
- bool **DirUP** = true
- bool **DirDOWN** = false
- uint8_t **UART2_rxBuffer** [12] = {0}

2.11.1 Detailed Description

Main File.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.11.2 Function Documentation

2.11.2.1 Error_Handler()

```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

Return values

<i>None</i>	
-------------	--

2.11.2.2 HAL_GPIO_EXTI_Callback()

```
void HAL_GPIO_EXTI_Callback (
    uint16_t GPIO_Pin )
```

EXTI Callback Function.

Parameters

<i>void</i>	
-------------	--

Returns

None

2.11.2.3 HAL_TIM_PeriodElapsedCallback()

```
void HAL_TIM_PeriodElapsedCallback (
    TIM_HandleTypeDef * htim )
```

TIM3 (5s) Callback Function.

Parameters

<i>void</i>	
-------------	--

Returns

None

2.11.2.4 main()

```
int main (
    void )
```

The application entry point.

Return values

<i>int</i>	
------------	--

2.11.2.5 SystemClock_Config()

```
void SystemClock_Config (
    void )
```

System Clock Configuration.

Return values

None	
------	--

Configure the main internal regulator output voltage

Initializes the RCC Oscillators according to the specified parameters in the RCC_OscInitTypeDef structure.

Initializes the CPU, AHB and APB buses clocks

2.12 main.h File Reference

: Header for [main.c](#) file. This file contains the common defines of the application.

```
#include "stm32f4xx_hal.h"
#include <stdbool.h>
```

Macros

- `#define PC14_OSC32_IN_Pin` GPIO_PIN_14
- `#define PC14_OSC32_IN_GPIO_Port` GPIOC
- `#define PC15_OSC32_OUT_Pin` GPIO_PIN_15
- `#define PC15_OSC32_OUT_GPIO_Port` GPIOC
- `#define PH0_OSC_IN_Pin` GPIO_PIN_0
- `#define PH0_OSC_IN_GPIO_Port` GPIOH
- `#define PH1_OSC_OUT_Pin` GPIO_PIN_1
- `#define PH1_OSC_OUT_GPIO_Port` GPIOH
- `#define BOOT1_Pin` GPIO_PIN_2
- `#define BOOT1_GPIO_Port` GPIOB
- `#define SWDIO_Pin` GPIO_PIN_13
- `#define SWDIO_GPIO_Port` GPIOA
- `#define SWCLK_Pin` GPIO_PIN_14
- `#define SWCLK_GPIO_Port` GPIOA
- `#define SWO_Pin` GPIO_PIN_3
- `#define SWO_GPIO_Port` GPIOB
- `#define LD3_Pin` GPIO_PIN_13
- `#define LD3_GPIO_Port` GPIOD
- `#define LD4_Pin` GPIO_PIN_12
- `#define LD4_GPIO_Port` GPIOD
- `#define LD5_Pin` GPIO_PIN_14
- `#define LD5_GPIO_Port` GPIOD
- `#define LD6_Pin` GPIO_PIN_15
- `#define LD6_GPIO_Port` GPIOD
- `#define Servo1_Pin` GPIO_PIN_9
- `#define Servo1_GPIO_Port` GPIOE
- `#define Servo2_Pin` GPIO_PIN_11
- `#define Servo2_GPIO_Port` GPIOE

Enumerations

- enum **NUMBER_FLOOR** { **GROUND_FLOOR** , **FIRST_FLOOR** , **MOVING** }

Functions

- void [Error_Handler](#) (void)

This function is executed in case of error occurrence.

Variables

- bool **buttonPressFlag**
- bool **timer100msEndFlag**
- bool **timer5sEndFlag**
- uint8_t **step_positions** [8]
- uint8_t **step_index**
- **NUMBER_FLOOR** **current_floor**
- bool **isMoving**
- bool **DirUP**
- bool **DirDOWN**

2.12.1 Detailed Description

: Header for [main.c](#) file. This file contains the common defines of the application.

Attention

© Copyright (c) 2021 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

2.12.2 Function Documentation

2.12.2.1 Error_Handler()

```
void Error_Handler (  
    void )
```

This function is executed in case of error occurrence.

Return values

<i>None</i>	
-------------	--

2.13 motor.c File Reference

Motor Functions.

```
#include "motor.h"
```

Functions

- void [Motor_Init](#) (void)
Initialize the motor hardware.

2.13.1 Detailed Description

Motor Functions.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.13.2 Function Documentation

2.13.2.1 Motor_Init()

```
void Motor_Init (
    void )
```

Initialize the motor hardware.

Parameters

<i>void</i>	
-------------	--

Returns

None

2.14 motor.h File Reference

This file contains all the function prototypes for the [motor.c](#) file.

```
#include "main.h"
```

Functions

- void [Motor_Init](#) (void)
Initialize the motor hardware.

2.14.1 Detailed Description

This file contains all the function prototypes for the [motor.c](#) file.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.14.2 Function Documentation

2.14.2.1 Motor_Init()

```
void Motor_Init (  
    void )
```

Initialize the motor hardware.

Parameters

<i>void</i>	
-------------	--

Returns

None

2.15 SysTick_Handler.c File Reference

SysTick Handler. This function, along with other interrupt handlers is defined at stm32f4xx_it.c. In order to address it, we have created this file. This file is not included on the project.

Functions

- void [SysTick_Handler](#) (void)

Increases a variable each ms and if the lift is moving, updates the stepper output according to the direction of the movement.

2.15.1 Detailed Description

SysTick Handler. This function, along with other interrupt handlers is defined at stm32f4xx_it.c. In order to address it, we have created this file. This file is not included on the project.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.15.2 Function Documentation

2.15.2.1 SysTick_Handler()

```
void SysTick_Handler (  
    void )
```

Increases a variable each ms and if the lift is moving, updates the stepper output according to the direction of the movement.

Parameters

<i>void</i>	
-------------	--

Returns

void

2.16 timer.c File Reference

Timer(s) Initialization.


```
#include "timer.h"
```

Functions

- void [TIM1_Init](#) (void)
TIM1 Init Function.
- void [TIM3_Init](#) (void)
TIM3 Init Function.
- void [TIM4_Init](#) (void)
TIM4 Init Function.

Variables

- TIM_HandleTypeDef **htim1**
- TIM_HandleTypeDef **htim3**
- TIM_HandleTypeDef **htim4**

2.16.1 Detailed Description

Timer(s) Initialization.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.16.2 Function Documentation

2.16.2.1 TIM1_Init()

```
void TIM1_Init (  
    void )
```

TIM1 Init Function.

Parameters

<i>void</i>	
-------------	--

Returns

None

2.16.2.2 TIM3_Init()

```
void TIM3_Init (
    void )
```

TIM3 Init Function.

Parameters

<i>void</i>	
-------------	--

Returns

None

2.16.2.3 TIM4_Init()

```
void TIM4_Init (
    void )
```

TIM4 Init Function.

Parameters

<i>void</i>	
-------------	--

Returns

None

TIM4 GPIO Configuration PD13 -----> TIM4_CH2 PD14 -----> TIM4_CH3

2.17 timer.h File ReferenceThis file contains all the function prototypes for the [timer.c](#) file.

```
#include "main.h"
```

Functions

- void [TIM1_Init](#) (void)
TIM1 Init Function.
- void [TIM3_Init](#) (void)
TIM3 Init Function.
- void [TIM4_Init](#) (void)
TIM4 Init Function.

Variables

- TIM_HandleTypeDef **htim1**
- TIM_HandleTypeDef **htim3**
- TIM_HandleTypeDef **htim4**

2.17.1 Detailed Description

This file contains all the function prototypes for the [timer.c](#) file.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.17.2 Function Documentation

2.17.2.1 TIM1_Init()

```
void TIM1_Init (  
                void )
```

TIM1 Init Function.

Parameters

<i>void</i>	
-------------	--

Returns

None

2.17.2.2 TIM3_Init()

```
void TIM3_Init (
    void )
```

TIM3 Init Function.

Parameters

<i>void</i>	
-------------	--

Returns

None

2.17.2.3 TIM4_Init()

```
void TIM4_Init (
    void )
```

TIM4 Init Function.

Parameters

<i>void</i>	
-------------	--

Returns

None

TIM4 GPIO Configuration PD13 ----> TIM4_CH2 PD14 ----> TIM4_CH3

2.18 uart.c File Reference

UART implementation.

```
#include "uart.h"
```

Functions

- void [USART2_Init](#) (void)
USART2 Initialization.

Variables

- UART_HandleTypeDef **huart2**

2.18.1 Detailed Description

UART implementation.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.18.2 Function Documentation

2.18.2.1 USART2_Init()

```
void USART2_Init (
    void )
```

USART2 Initialization.

Parameters

<i>void</i>	
-------------	--

Returns

None

2.19 uart.h File Reference

This file contains all the function prototypes for the [uart.c](#) file.

```
#include "main.h"
```

Functions

- void [USART2_Init](#) (void)
USART2 Initialization.

Variables

- UART_HandleTypeDef **huart2**

2.19.1 Detailed Description

This file contains all the function prototypes for the [uart.c](#) file.

Author

Alejandro López Rodríguez and Ana Maria Casanova López

Date

06/06/2021

2.19.2 Function Documentation

2.19.2.1 USART2_Init()

```
void USART2_Init (
    void )
```

USART2 Initialization.

Parameters

<i>void</i>	
-------------	--

Returns

None

Index

- button.c, [3](#)
 - Button_Init, [3](#)
- button.h, [4](#)
 - Button_Init, [4](#)
- Button_Init
 - button.c, [3](#)
 - button.h, [4](#)
- CloseDoors
 - doors.c, [5](#)
 - doors.h, [9](#)
- doors.c, [5](#)
 - CloseDoors, [5](#)
 - Doors_Init, [6](#)
 - initDoorsPWM, [6](#)
 - OpenDoors, [6](#)
 - Servo_PWM_Angle, [8](#)
- doors.h, [8](#)
 - CloseDoors, [9](#)
 - Doors_Init, [9](#)
 - initDoorsPWM, [10](#)
 - OpenDoors, [10](#)
 - Servo_PWM_Angle, [10](#)
- Doors_Init
 - doors.c, [6](#)
 - doors.h, [9](#)
- Error_Handler
 - main.c, [28](#)
 - main.h, [31](#)
- FirstFloor
 - led.c, [14](#)
 - led.h, [17](#)
- gpio.c, [11](#)
 - MX_GPIO_Init, [11](#)
- gpio.h, [12](#)
 - MX_GPIO_Init, [12](#)
- GroundFloor
 - led.c, [14](#)
 - led.h, [18](#)
- HAL_GPIO_EXTI_Callback
 - main.c, [28](#)
- HAL_TIM_PeriodElapsedCallback
 - main.c, [29](#)
- initDoorsPWM
 - doors.c, [6](#)
- doors.h, [10](#)
- initLEDGPIO
 - led.c, [14](#)
 - led.h, [18](#)
- initLEDsPWM
 - led.c, [15](#)
 - led.h, [18](#)
- led.c, [13](#)
 - FirstFloor, [14](#)
 - GroundFloor, [14](#)
 - initLEDGPIO, [14](#)
 - initLEDsPWM, [15](#)
 - LED_Init, [15](#)
 - LED_PWM_Percent, [15](#)
 - Moving, [16](#)
 - TurnOffLEDPWM, [16](#)
- led.h, [16](#)
 - FirstFloor, [17](#)
 - GroundFloor, [18](#)
 - initLEDGPIO, [18](#)
 - initLEDsPWM, [18](#)
 - LED_Init, [19](#)
 - LED_PWM_Percent, [19](#)
 - Moving, [19](#)
 - TurnOffLEDPWM, [20](#)
- LED_Init
 - led.c, [15](#)
 - led.h, [19](#)
- LED_PWM_Percent
 - led.c, [15](#)
 - led.h, [19](#)
- lift.c, [20](#)
 - lift_Init, [21](#)
 - liftDown, [21](#)
 - liftFloor, [22](#)
 - liftIsMoving, [22](#)
 - liftStop, [23](#)
 - liftUp, [23](#)
- lift.h, [23](#)
 - lift_Init, [24](#)
 - liftDown, [24](#)
 - liftFloor, [25](#)
 - liftIsMoving, [25](#)
 - liftStop, [25](#)
 - liftUp, [27](#)
- lift_Init
 - lift.c, [21](#)
 - lift.h, [24](#)
- liftDown

- lift.c, [21](#)
- lift.h, [24](#)
- liftFloor
 - lift.c, [22](#)
 - lift.h, [25](#)
- liftIsMoving
 - lift.c, [22](#)
 - lift.h, [25](#)
- liftStop
 - lift.c, [23](#)
 - lift.h, [25](#)
- liftUp
 - lift.c, [23](#)
 - lift.h, [27](#)
- main
 - main.c, [29](#)
- main.c, [27](#)
 - Error_Handler, [28](#)
 - HAL_GPIO_EXTI_Callback, [28](#)
 - HAL_TIM_PeriodElapsedCallback, [29](#)
 - main, [29](#)
 - SystemClock_Config, [29](#)
- main.h, [30](#)
 - Error_Handler, [31](#)
- motor.c, [32](#)
 - Motor_Init, [32](#)
- motor.h, [33](#)
 - Motor_Init, [33](#)
- Motor_Init
 - motor.c, [32](#)
 - motor.h, [33](#)
- Moving
 - led.c, [16](#)
 - led.h, [19](#)
- MX_GPIO_Init
 - gpio.c, [11](#)
 - gpio.h, [12](#)
- OpenDoors
 - doors.c, [6](#)
 - doors.h, [10](#)
- Servo_PWM_Angle
 - doors.c, [8](#)
 - doors.h, [10](#)
- SystemClock_Config
 - main.c, [29](#)
- SysTick_Handler
 - SysTick_Handler.c, [34](#)
- SysTick_Handler.c, [34](#)
 - SysTick_Handler, [34](#)
- TIM1_Init
 - timer.c, [35](#)
 - timer.h, [37](#)
- TIM3_Init
 - timer.c, [36](#)
 - timer.h, [37](#)
- TIM4_Init
 - timer.c, [36](#)
 - timer.h, [38](#)
- timer.c, [34](#)
 - TIM1_Init, [35](#)
 - TIM3_Init, [36](#)
 - TIM4_Init, [36](#)
- timer.h, [36](#)
 - TIM1_Init, [37](#)
 - TIM3_Init, [37](#)
 - TIM4_Init, [38](#)
- TurnOffLEDPWM
 - led.c, [16](#)
 - led.h, [20](#)
- uart.c, [38](#)
 - USART2_Init, [39](#)
- uart.h, [39](#)
 - USART2_Init, [40](#)
- USART2_Init
 - uart.c, [39](#)
 - uart.h, [40](#)