Microproject

Generated by Doxygen 1.9.1

Chapter 1

File Index

1.1 File List

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

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

void	

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

void	

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

void	

Returns

void

2.3.2.2 Doors_Init()

```
void Doors_Init (
            void  )
```

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

Parameters

void	

Returns

void

2.3.2.3 initDoorsPWM()

```
void initDoorsPWM (
            void  )
```

Initialize PWM Channels 1&2 of TIM1.

Parameters

void	

Returns

void

2.3.2.4 OpenDoors()

```
void OpenDoors (
            void  )
```

Set servo angle to open position.

Parameters

void	

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

uint16_t	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 Maria 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

void	

Returns

void

2.4.2.2 Doors_Init()

```
void Doors_Init (
            void  )
```

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

Parameters

void	

Returns

void

2.4.2.3 initDoorsPWM()

```
void initDoorsPWM (
            void  )
```

Initialize PWM Channels 1&2 of TIM1.

Parameters

void	

Returns

void

2.4.2.4 OpenDoors()

```
void OpenDoors (
            void  )
```

Set servo angle to open position.

Parameters

void	

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

uint16_t	Servo, uint8_t 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

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

void	

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

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

void	

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

uint16_t	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

None	

Returns

None

2.7.2.8 TurnOffLEDPWM()

```
void TurnOffLEDPWM (
             void  )
```

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

Parameters

None	

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

None	

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

None	

Returns

None

2.8.2.5 LED_Init()

```
void LED_Init (
            void  )
```

Initialize the LED hardware.

Parameters

None	

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

uint16_t	LED, uint8_t 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

void	

Returns

void

2.9.2.2 liftDown()

```
void liftDown (
            void  )
```

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

Parameters

void	

Returns

void

2.9.2.3 liftFloor()

```
int32_t liftFloor (
            void  )
```

Retrieve the floor where the lift is.

Parameters

void	

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

void	

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

void	

Returns

void

2.10.2.2 liftDown()

```
void liftDown (
            void  )
```

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

Parameters

void	

Returns

void

2.10.2.3 liftFloor()

```
int32_t liftFloor (
            void  )
```

Retrieve the floor where the lift is.

Parameters

void	

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

void	

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

void	

Returns

None

2.10.2.6 liftUp()

```
void liftUp (
            void  )
```

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

Parameters

void	

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

None	

2.11.2.2 HAL_GPIO_EXTI_Callback()

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

EXTI Callback Function.

Parameters

void	

Returns

None

2.11.2.3 HAL_TIM_PeriodElapsedCallback()

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

TIM3 (5s) Callback Function.

Parameters

void	

Returns

None

2.11.2.4 main()

```
int main (
            void  )
```

The application entry point.

Return values

int	

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

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

None	

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

void	

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

void	

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 movememt.

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 movememt.

Parameters

void	

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

void	

Returns

None

2.16.2.2 TIM3_Init()

```
void TIM3_Init (
             void  )
```

TIM3 Init Function.

Parameters

void	

Returns

None

2.16.2.3 TIM4_Init()

```
void TIM4_Init (
             void  )
```

TIM4 Init Function.

Parameters

void	

Returns

None

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

2.17 timer.h File Reference

This 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

void	

Returns

None

2.17.2.2 TIM3_Init()

```
void TIM3_Init (
            void  )
```

TIM3 Init Function.

Parameters

void	

Returns

None

2.17.2.3 TIM4_Init()

```
void TIM4_Init (
            void  )
```

TIM4 Init Function.

Parameters

void	

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

void	

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

void	

Returns

None

Index