Aeropendulum

Generated by Doxygen 1.8.13

Contents

Chapter 1

Aeropendulum

Repositorio para el proyecto final de la materia sistemas de control

Arbol de proyecto

Se espera que el proyecto siga la siguente estructura:

```
+-- README.md
+-- .gitignore
|
+-- firmware
|       +-- Makefile
|       +-- libs
|       |       +-- cmsis
|       |       |       +-- inc
|       |       |       |       +-- (cmsis headers)
|       |       |       +-- src
|       |       |       |       +-- (cmsis sources)
|       |       +-- hal
|       |       |       +-- inc
|       |       |       |       +-- (hal headers)
|       |       |       +-- src
|       |       |       |       +-- (hal sources)
|       +-- scripts
|       |       +-- openocd
|       |       |       +-- stm32f4discovery.cfg
|       |       |       +-- gdbinit
|       +-- aero
|       |       +-- inc
|       |       |       +-- main.h
|       |       |       +-- FreeRTOSConfig.h
|       |       |       +-- (tasks and/or files headers)
|       |       +-- src
|       |       |       +-- main.c
|       |       |       +-- FreeRTOSConfig.c
|       |       |       +-- (tasks and/or files headers)
+-- mechanical
|       +-- 3DPrints
|       |       +-- freecad
|       |       |       +-- *.fcstd
|       |       +-- stl
|       |       |       +-- *.stl
|       +-- blueprints
|       |       +-- *.*
```

Lineamientos de código

- Se trabaja con tabs equivalentes a 8 espacios.
- Variables y parámetros tienen tipos de datos completos.
- Expresiones para constantes #define están encerradas en paréntesis. Si es posible, se busca utilizar:
- Nombres en *MAYÚSCULAS* para identificar registros o instrucciones del procesador.
- Nombres en *camelCase* para identificar funciónes o rutinas de interrupciones.
- Prefijos Namespace_ para agrupar funciónes relacionadas (ej. cuando creamos una librería para periféricos).

Compilación

Linux:

```
cd firmware
make
```

Prequisitos

- arm-none-eabi-∗
- GNU Make

Target Download

Linux

```
cd firmware
make download
```

Prequisitos

- arm-none-eabi-∗
- GNU Make
- OpenOCD

Debug

Linux

```
cd firmware
make server_open
```

open new terminal

```
cd firmware
make server_connect
```

Prequisitos

- arm-none-eabi-∗
- GNU Make
- OpenOCD

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Chapter 3

File Index

3.1 File List

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

Chapter 4

Module Documentation

4.1 Included dependencies

4.2 Function primitives

Functions

- uint8_t GPIO_ModeSet (uint8_t port, uint8_t pin, uint8_t mode)

  Set GPIO Mode (with MODER reg.)
- uint8_t GPIO_SetPullUpPullDown (uint8_t port, uint8_t pin, uint8_t pupd)

  Set GPIO Pull up/down config (with PUPDR reg.)
- uint8_t GPIO_OutSpeed (uint8_t port, uint8_t pin, uint8_t speed)

  Set GPIO Out Speed (with OSPEEDR reg.)
- uint8_t GPIO_OutData (uint8_t port, uint8_t pin, uint8_t data)

  Set GPIO Out Data (with ODR reg.)
- uint8_t GPIO_SetAlternateFunction (uint8_t port, uint8_t pin, uint8_t af)

  Set GPIO Alternate Function (with AFRL and AFRH reg.)
- uint8_t EXTI0_IRQHandler (void)

  Handle External interrupt 0.
- uint8_t RCC_GPIOPortSetClock (uint8_t port, uint8_t state)

  Set GPIOx port clock.

4.2.1 Detailed Description

4.2.2 Function Documentation

4.2.2.1 EXTI0_IRQHandler()

```
uint8_t EXTI0_IRQHandler (
            void  )
```

Handle External interrupt 0.

Return values

0	if success, -1 if error

4.2.2.2 GPIO_ModeSet()

```
uint8_t GPIO_ModeSet (
            uint8_t port,
            uint8_t pin,
            uint8_t mode )
```

Set GPIO Mode (with MODER reg.)

Parameters

port	The GPIO Port (A=0, I=8)
pin	The GPIO pin number
mode	The selected mode: 0b00 = Input 0b01 = Output 0b10 = Alternate function 0b11 = Analog mode

Return values

0	if success

4.2.2.3 GPIO_OutData()

```
uint8_t GPIO_OutData (
            uint8_t port,
            uint8_t pin,
            uint8_t data )
```

Set GPIO Out Data (with ODR reg.)

Parameters

port	The GPIO Port (A=0, I=8)
pin	The GPIO pin number
dir	Output data: 0b00 = In 0b01 = Out

Return values

0	if success

4.2.2.4 GPIO_OutSpeed()

```
uint8_t GPIO_OutSpeed (
            uint8_t port,
            uint8_t pin,
            uint8_t speed )
```

Set GPIO Out Speed (with OSPEEDR reg.)

Parameters

port	The GPIO Port (A=0, I=8)
pin	The GPIO pin number
speed	The Speed: 0b00 = Low (limit 2MHz) 0b01 = Medium (limit 10MHz) 0b10 = High (limit 50MHz) 0b11 = Very High (limit 100MHz)

Return values

0	if success

4.2.2.5 GPIO_SetAlternateFunction()

```
uint8_t GPIO_SetAlternateFunction (
            uint8_t port,
            uint8_t pin,
            uint8_t af )
```

Set GPIO Alternate Function (with AFRL and AFRH reg.)

Parameters

port	The GPIO Port (A=0, I=8)
pin	The GPIO pin number
af	The Alternate function number

Return values

0	if success

4.2.2.6 GPIO_SetPullUpPullDown()

```
uint8_t GPIO_SetPullUpPullDown (
            uint8_t port,
            uint8_t pin,
            uint8_t pupd )
```

Set GPIO Pull up/down config (with PUPDR reg.)

Parameters

port	The GPIO Port (A=0, I=8)
pin	The GPIO pin number
pupd	The Configuration: 0b00 = No pull 0b01 = Pull-Up 0b10 = Pull-Down 0b11 = Reserved

Return values

0	if success

4.2.2.7 RCC_GPIOPortSetClock()

```
uint8_t RCC_GPIOPortSetClock (
            uint8_t port,
            uint8_t state )
```

Set GPIOx port clock.

Parameters

port	represents the x port of GPIO A=0 ,I=8
state	1=enable clock, 0=disable clock

Return values

0	if success, 1 if error

4.3 Included headers

4.4 GPIO Functions

Functions

- uint8_t GPIO_ModeSet (uint8_t port, uint8_t pin, uint8_t mode)
  Set GPIO Mode (with MODER reg.)
- uint8_t GPIO_SetPullUpPullDown (uint8_t port, uint8_t pin, uint8_t pupd)
  Set GPIO Pull up/down config (with PUPDR reg.)
- uint8_t GPIO_OutSpeed (uint8_t port, uint8_t pin, uint8_t speed)
  Set GPIO Out Speed (with OSPEEDR reg.)
- uint8_t GPIO_SetAlternateFunction (uint8_t port, uint8_t pin, uint8_t af)
  Set GPIO Alternate Function (with AFRL and AFRH reg.)
- uint8_t GPIO_OutData (uint8_t port, uint8_t pin, uint8_t data)
  Set GPIO Out Data (with ODR reg.)

4.4.1 Detailed Description

4.4.2 Function Documentation

4.4.2.1 GPIO_ModeSet()

```
uint8_t GPIO_ModeSet (
            uint8_t port,
            uint8_t pin,
            uint8_t mode )
```

Set GPIO Mode (with MODER reg.)

Parameters

port	The GPIO Port (A=0, I=8)
pin	The GPIO pin number
mode	The selected mode: 0b00 = Input 0b01 = Output 0b10 = Alternate function 0b11 = Analog mode

Return values

0	if success

4.4.2.2 GPIO_OutData()

```
uint8_t GPIO_OutData (
            uint8_t port,
```

```
            uint8_t pin,
            uint8_t data )
```

Set GPIO Out Data (with ODR reg.)

Parameters

port	The GPIO Port (A=0, I=8)
pin	The GPIO pin number
dir	Output data: 0b00 = In 0b01 = Out

Return values

0	if success

4.4.2.3 GPIO_OutSpeed()

```
uint8_t GPIO_OutSpeed (
            uint8_t port,
            uint8_t pin,
            uint8_t speed )
```

Set GPIO Out Speed (with OSPEEDR reg.)

Parameters

port	The GPIO Port (A=0, I=8)
pin	The GPIO pin number
speed	The Speed: 0b00 = Low (limit 2MHz) 0b01 = Medium (limit 10MHz) 0b10 = High (limit 50MHz) 0b11 = Very High (limit 100MHz)

Return values

0	if success

4.4.2.4 GPIO_SetAlternateFunction()

```
uint8_t GPIO_SetAlternateFunction (
            uint8_t port,
            uint8_t pin,
            uint8_t af )
```

Set GPIO Alternate Function (with AFRL and AFRH reg.)

Parameters

port	The GPIO Port (A=0, I=8)
pin	The GPIO pin number
af	The Alternate function number

Return values

0	if success

4.4.2.5 GPIO_SetPullUpPullDown()

```
uint8_t GPIO_SetPullUpPullDown (
            uint8_t port,
            uint8_t pin,
            uint8_t pupd )
```

Set GPIO Pull up/down config (with PUPDR reg.)

Parameters

port	The GPIO Port (A=0, I=8)
pin	The GPIO pin number
pupd	The Configuration: 0b00 = No pull 0b01 = Pull-Up 0b10 = Pull-Down 0b11 = Reserved

Return values

0	if success

4.5 Interrupt Service Routines

Functions

- uint8_t EXTI0_IRQHandler (void)

  Handle External interrupt 0.

4.5.1 Detailed Description

4.5.2 Function Documentation

4.5.2.1 EXTI0_IRQHandler()

```
uint8_t EXTI0_IRQHandler (
            void  )
```

Handle External interrupt 0.

Return values

0	if success, -1 if error

4.6 RCC functions

Functions

- uint8_t RCC_GPIOPortSetClock (uint8_t port, uint8_t state)

  Set GPIOx port clock.

4.6.1 Detailed Description

4.6.2 Function Documentation

4.6.2.1 RCC_GPIOPortSetClock()

```
uint8_t RCC_GPIOPortSetClock (
            uint8_t port,
            uint8_t state )
```

Set GPIOx port clock.

Parameters

port	represents the x port of GPIO A=0 ,I=8
state	1=enable clock, 0=disable clock

Return values

0	if success, 1 if error

4.7 GPIO Constants

Macros

- #define **LD4_GPIO_PIN** (1 << 12) /* PD12 */
- #define **LD4_MODE_OUT** (1 << 24) /* General Purpose Input for PD12 */
- #define **LD5_GPIO_PIN** (1 << 14) /* PD14 */
- #define **PD14_AFRH_AF2** ((uint32_t)0x02000000) /* PD14 to TIM4 */
- #define **PD14_OSPEEDR_VHS** ((uint32_t)0x30000000) /* PD14 High Speed (100Mhz and beyond) */
- #define **LD5_MODE_ALT** ((uint32_t)0x20000000) /* General Purpose Input for PD14 */

4.7.1 Detailed Description

4.8 TIM Constants

Collaboration diagram for TIM Constants:

Modules

- PWM Constants

Macros

- #define **TIM4_ARR** ((uint16_t)0x20CF) /∗ Auto reload register = 84000000 / 10000 - 1 = 8399 set for 10kHz PWM∗/
- #define **TIM_PSCReloadMode_Immediate** ((uint16_t)0x0001)

4.8.1 Detailed Description

4.9 PWM Constants

Collaboration diagram for PWM Constants:

Macros

- #define **TIM4_CCMR2_OC3M** ((uint16_t)0x0070) /∗ PWM mode 2 ∗/
- #define **TIM4_CCER_CC3E** ((uint16_t)0x0001 << 8)
- #define **TIM4_CCER_CC3P** ((uint16_t)0x0002 << 8) /∗ Polarity Low ∗/
- #define **TIM4_CCR3** ((uint16_t)0x00FF) /∗ 25% duty TODO: implement a macro ∗/
- #define **TIM4_OCPreload_Enable** ((uint16_t)0x0008)

4.9.1 Detailed Description

4.10 Function definitions

Functions

- uint8_t GPIO_Init (void)

  Enables and sets peripherals.
- uint8_t TIM3_Init (void)

  Init sequence for TIM3.
- uint8_t TIM4_Init (void)

  Init sequence for TIM4.
- uint8_t PWM_Init (void)

  Init sequence for PWM.
- uint8_t PWM_SetDuty (uint8_t duty)

  Change pwm duty value.
- uint8_t blinkLed4 (void)

  Creates an infinite loop where the LD4 blinks.

4.10.1 Detailed Description

4.10.2 Function Documentation

4.10.2.1 blinkLed4()

```
uint8_t blinkLed4 (
            void  )
```

Creates an infinite loop where the LD4 blinks.

Return values

0	if success

4.10.2.2 GPIO_Init()

```
uint8_t GPIO_Init (
            void  )
```

Enables and sets peripherals.

Return values

0	if success.

4.10.2.3 PWM_Init()

```
uint8_t PWM_Init (
            void  )
```

Init sequence for PWM.

Return values

0	if success.

4.10.2.4 PWM_SetDuty()

```
uint8_t PWM_SetDuty (
            uint8_t duty )
```

Change pwm duty value.

Parameters

duty	integer between 0 and 100 corresponding to the duty percentage.

Return values

0	if success

4.10.2.5 TIM3_Init()

```
uint8_t TIM3_Init (
            void  )
```

Init sequence for TIM3.

Return values

0	if success.

4.10.2.6 TIM4_Init()

```
uint8_t TIM4_Init (
             void  )
```

Init sequence for TIM4.

Return values

0	if success.

Chapter 5

File Documentation

5.1 main.cpp File Reference

main file, entry point and pin definitions.

```
#include "main.h"
```
Include dependency graph for main.cpp:

5.2 main.h File Reference

main header

```
#include <cstdio>
#include "stm32f4xx_HAL.h"
```
Include dependency graph for main.h: This graph shows which files directly or indirectly include this file:

Macros

- #define **LD4_GPIO_PIN** (1 << 12) /* PD12 */
- #define **LD4_MODE_OUT** (1 << 24) /* General Purpose Input for PD12 */
- #define **LD5_GPIO_PIN** (1 << 14) /* PD14 */
- #define **PD14_AFRH_AF2** ((uint32_t)0x02000000) /* PD14 to TIM4 */
- #define **PD14_OSPEEDR_VHS** ((uint32_t)0x30000000) /* PD14 High Speed (100Mhz and beyond) */
- #define **LD5_MODE_ALT** ((uint32_t)0x20000000) /* General Purpose Input for PD14 */
- #define **TIM4_ARR** ((uint16_t)0x20CF) /* Auto reload register = 84000000 / 10000 - 1 = 8399 set for 10kHz PWM*/
- #define **TIM_PSCReloadMode_Immediate** ((uint16_t)0x0001)
- #define **TIM4_CCMR2_OC3M** ((uint16_t)0x0070) /* PWM mode 2 */
- #define **TIM4_CCER_CC3E** ((uint16_t)0x0001 << 8)
- #define **TIM4_CCER_CC3P** ((uint16_t)0x0002 << 8) /* Polarity Low */
- #define **TIM4_CCR3** ((uint16_t)0x00FF) /* 25% duty TODO: implement a macro */
- #define **TIM4_OCPreload_Enable** ((uint16_t)0x0008)

5.2.1 Detailed Description

main header

Author

Marco Miretti

See also

https://github.com/MarcoMiretti

Copyright

Date

2019/10/30

5.3 stm32f4xx_HAL.c File Reference

main file, entry point and pin definitions.

5.3.1 Detailed Description

main file, entry point and pin definitions.

Author

Marco Miretti

See also

https://github.com/MarcoMiretti

Copyright

Date

2019/11/13

5.4 stm32f4xx_HAL.h File Reference

principal HAL header file

```
#include <assert.h>
#include "stm32f4xx.h"
#include "stm32f4xx_HAL_RCC.h"
#include "stm32f4xx_HAL_GPIO.h"
```

Include dependency graph for stm32f4xx_HAL.h: This graph shows which files directly or indirectly include this file:

5.4.1 Detailed Description

principal HAL header file

Author

Marco Miretti

See also

https://github.com/MarcoMiretti

Copyright

License: gpl-3.0 (see LICENSE.txt)

Date

2019/11/13

5.5 stm32f4xx_HAL_GPIO.c File Reference

GPIO_HAL source file.

```
#include "stm32f4xx_HAL.h"
```

Include dependency graph for stm32f4xx_HAL_GPIO.c:

Functions

- uint8_t GPIO_ModeSet (uint8_t port, uint8_t pin, uint8_t mode)
  Set GPIO Mode (with MODER reg.)
- uint8_t GPIO_SetPullUpPullDown (uint8_t port, uint8_t pin, uint8_t pupd)
  Set GPIO Pull up/down config (with PUPDR reg.)
- uint8_t GPIO_OutSpeed (uint8_t port, uint8_t pin, uint8_t speed)
  Set GPIO Out Speed (with OSPEEDR reg.)
- uint8_t GPIO_SetAlternateFunction (uint8_t port, uint8_t pin, uint8_t af)
  Set GPIO Alternate Function (with AFRL and AFRH reg.)
- uint8_t GPIO_OutData (uint8_t port, uint8_t pin, uint8_t data)
  Set GPIO Out Data (with ODR reg.)

5.5.1 Detailed Description

GPIO_HAL source file.

Author

Marco Miretti

See also

https://github.com/MarcoMiretti

Copyright

License: gpl-3.0 (see LICENSE.txt)

Date

2019/11/13

5.6 stm32f4xx_HAL_GPIO.h File Reference

GPIO_HAL header file.

```
#include "stm32f4xx_HAL.h"
```

Include dependency graph for stm32f4xx_HAL_GPIO.h: This graph shows which files directly or indirectly include this file:

Functions

- uint8_t GPIO_ModeSet (uint8_t port, uint8_t pin, uint8_t mode)

  Set GPIO Mode (with MODER reg.)
- uint8_t GPIO_SetPullUpPullDown (uint8_t port, uint8_t pin, uint8_t pupd)

  Set GPIO Pull up/down config (with PUPDR reg.)
- uint8_t GPIO_OutSpeed (uint8_t port, uint8_t pin, uint8_t speed)

  Set GPIO Out Speed (with OSPEEDR reg.)
- uint8_t GPIO_OutData (uint8_t port, uint8_t pin, uint8_t data)

  Set GPIO Out Data (with ODR reg.)
- uint8_t GPIO_SetAlternateFunction (uint8_t port, uint8_t pin, uint8_t af)

  Set GPIO Alternate Function (with AFRL and AFRH reg.)

5.6.1 Detailed Description

GPIO_HAL header file.

Author

Marco Miretti

See also

https://github.com/MarcoMiretti

Copyright

License: gpl-3.0 (see LICENSE.txt)

Date

2019/11/13

5.7 stm32f4xx_HAL_ISR.c File Reference

ISR_HAL source file.

```
#include "stm32f4xx_HAL_ISR.h"
```

Include dependency graph for stm32f4xx_HAL_ISR.c:

Functions

- uint8_t EXTI0_IRQHandler (void)

  Handle External interrupt 0.

5.7.1 Detailed Description

ISR_HAL source file.

Author

Marco Miretti

See also

https://github.com/MarcoMiretti

Copyright

License: gpl-3.0 (see LICENSE.txt)

Date

2019/11/15

5.8 stm32f4xx_HAL_ISR.h File Reference

ISR_HAL header file.

```
#include "stm32f4xx_HAL.h"
```

Include dependency graph for stm32f4xx_HAL_ISR.h: This graph shows which files directly or indirectly include this file:

Functions

- uint8_t EXTI0_IRQHandler (void)

  Handle External interrupt 0.

5.8.1 Detailed Description

ISR_HAL header file.

Author

Marco Miretti

See also

https://github.com/MarcoMiretti

Copyright

License: gpl-3.0 (see LICENSE.txt)

Date

2019/11/15

5.9 stm32f4xx_HAL_RCC.c File Reference

RCC_HAL source file.

```
#include "stm32f4xx_HAL_RCC.h"
```

Include dependency graph for stm32f4xx_HAL_RCC.c:

Functions

- uint8_t RCC_GPIOPortSetClock (uint8_t port, uint8_t state)

  Set GPIOx port clock.

5.9.1 Detailed Description

RCC_HAL source file.

Author

Marco Miretti

See also

https://github.com/MarcoMiretti

Copyright

License: gpl-3.0 (see LICENSE.txt)

Date

2019/11/13

5.10 stm32f4xx_HAL_RCC.h File Reference

RCC_HAL header file.

```
#include "stm32f4xx_HAL.h"
```

Include dependency graph for stm32f4xx_HAL_RCC.h: This graph shows which files directly or indirectly include this file:

Functions

- uint8_t RCC_GPIOPortSetClock (uint8_t port, uint8_t state)

  Set GPIOx port clock.

5.10.1 Detailed Description

RCC_HAL header file.

Author

Marco Miretti

See also

https://github.com/MarcoMiretti

Copyright

License: gpl-3.0 (see LICENSE.txt)

Date

2019/11/13

Index