

cdf2020BaseRoulanteRework

Generated by Doxygen 1.8.18



<b>1 Todo List</b>	<b>1</b>
<b>2 Module Index</b>	<b>3</b>
2.1 Modules	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Module Documentation</b>	<b>7</b>
4.1 motor_tim	7
4.1.1 Detailed Description	7
4.2 motor_a	8
4.2.1 Detailed Description	8
4.3 motor_b	9
4.3.1 Detailed Description	9
<b>5 File Documentation</b>	<b>11</b>
5.1 lowlevel/include/clock.h File Reference	11
5.1.1 Detailed Description	12
5.1.2 Macro Definition Documentation	12
5.1.2.1 RCC_CLOCK_FREQ_HZ	12
5.1.2.2 SYSTICK_FREQ_HZ	12
5.1.3 Function Documentation	12
5.1.3.1 clock_setup()	12
5.1.3.2 delay_ms()	12
5.2 lowlevel/include/gpio.h File Reference	13
5.2.1 Detailed Description	13
5.2.2 Function Documentation	13
5.2.2.1 gpio_setup_pin_af()	13
5.3 lowlevel/include/motor.h File Reference	14
5.3.1 Detailed Description	15
5.3.2 Macro Definition Documentation	15
5.3.2.1 PWM_PERIOD	15
5.3.2.2 PWM_PRESCALE	15
5.3.3 Enumeration Type Documentation	15
5.3.3.1 motor_sel	16
5.3.4 Function Documentation	16
5.3.4.1 motor_set()	16
5.3.4.2 motor_setup()	16
5.4 lowlevel/include/timer.h File Reference	16
5.4.1 Detailed Description	17
5.4.2 Function Documentation	17
5.4.2.1 timer_setup()	17
5.4.2.2 timer_setup_output_c()	18

5.4.2.3 <code>timer_start()</code> . . . . .	18
--	----

<b>Index</b>	<b>19</b>
--------------	-----------

## Chapter 1

## Todo List

Member `motor_set` (enum `motor_sel sel`, `int8_t value`)

we chose that 0 is forward and 1 is backward, it should be defined in a macro and adjustable for the motors



# Chapter 2

## Module Index

### 2.1 Modules

Here is a list of all modules:

motor_tim . . . . .	7
motor_a . . . . .	8
motor_b . . . . .	9





## Chapter 3

# File Index

### 3.1 File List

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

lowlevel/include/[clock.h](#)

This implements the setup of the system clock, acces fonction (debug) and temporal fonction (delay)

[11](#)

lowlevel/include/[gpio.h](#)

This implements the setup of a gpio pin

[13](#)

lowlevel/include/[motor.h](#)

This implements the functions required to pilot the propulsion motors of the robot

[14](#)

lowlevel/include/[timer.h](#)

This implements the functions required setup a timer and its output channel

[16](#)



## Chapter 4

# Module Documentation

### 4.1 motor\_tim

Internal timer used to pilot the motors.

#### Macros

- `#define MOTOR_TIM_RCC RCC_TIM3`
- `#define MOTOR_TIM TIM3`

#### 4.1.1 Detailed Description

Internal timer used to pilot the motors.

Two channels are used for the MOTOR\_A and MOTOR\_B

## 4.2 motor\_a

Definition for the MOTOR\_A.

### Macros

- #define **MOTOR\_A\_GPIO\_RCC\_EN** RCC\_GPIOA
- #define **MOTOR\_A\_PORT\_EN** GPIOA
- #define **MOTOR\_A\_PIN\_EN** GPIO4
- #define **MOTOR\_A\_AF** GPIO\_AF2
- #define **MOTOR\_A\_OC\_ID** TIM\_OC2
- #define **MOTOR\_A\_OC\_MODE** TIM\_OCM\_PWM1
- #define **MOTOR\_A\_GPIO\_RCC\_DIR** RCC\_GPIOA
- #define **MOTOR\_A\_PORT\_DIR** GPIOA
- #define **MOTOR\_A\_PIN\_DIR** GPIO3
- #define **MOTOR\_A\_INIT\_DIR** 0

### 4.2.1 Detailed Description

Definition for the MOTOR\_A.

EN stand for enable (output of the PWM signal)

We use OC\_ID to use a specific channel as a PWM\_output

DIR stand for direction (boolean value)

INIT\_DIR is the initial direction of the motor

## 4.3 motor\_b

Definition for the MOTOR\_B.

### Macros

- #define **MOTOR\_B\_GPIO\_RCC\_EN** RCC\_GPIOA
- #define **MOTOR\_B\_PORT\_EN** GPIOA
- #define **MOTOR\_B\_PIN\_EN** GPIO6
- #define **MOTOR\_B\_AF** GPIO\_AF2
- #define **MOTOR\_B\_OC\_ID** TIM\_OC1
- #define **MOTOR\_B\_OC\_MODE** TIM\_OCM\_PWM1
- #define **MOTOR\_B\_GPIO\_RCC\_DIR** RCC\_GPIOA
- #define **MOTOR\_B\_PORT\_DIR** GPIOA
- #define **MOTOR\_B\_PIN\_DIR** GPIO7
- #define **MOTOR\_B\_INIT\_DIR** 0

### 4.3.1 Detailed Description

Definition for the MOTOR\_B.

EN stand for enable (output of the PWM signal)

We use OC\_ID to use a specific channel as a PWM\_output

DIR stand for direction (boolean value)

INIT\_DIR is the initial direction of the motor



## Chapter 5

# File Documentation

### 5.1 lowlevel/include/clock.h File Reference

This implements the setup of the system clock, acces fonction (debug) and temporal fonction (delay)

```
#include <stdint.h>
```

#### Macros

- #define [RCC\\_CLOCK\\_FREQ\\_HZ](#) ( 64000000)
- #define [SYSTICK\\_FREQ\\_HZ](#) ( 100000)
- #define [SYSTICK\\_PERIOD](#) ([RCC\\_CLOCK\\_FREQ\\_HZ](#) / [SYSTICK\\_FREQ\\_HZ](#))
- #define [MICROS\\_SYSTICK\\_RATIO](#) (1000000 / [SYSTICK\\_FREQ\\_HZ](#))
- #define [MILLIS\\_TO\\_SYSTICK](#)(ms) (ms \* 1000 / [MICROS\\_SYSTICK\\_RATIO](#))
- #define [SYSTICK\\_TO\\_MILLIS](#)(ticks) (ticks \* [MICROS\\_SYSTICK\\_RATIO](#) / 1000)

#### Functions

- void [clock\\_setup](#) ()  
*This function setup the system clock.*
- uint32\_t [clock\\_get\\_systicks](#) ()  
*This function gets the number of systicks since starting.*
- uint32\_t [clock\\_get\\_uptime\\_ms](#) ()  
*This function gets the uptime in ms.*
- void [delay\\_ms](#) (uint32\_t ms)  
*This function implements a delay in ms.*

### 5.1.1 Detailed Description

This implements the setup of the system clock, acces fonction (debug) and temporal fonction (delay)

This file is part of cdfr2020BaseRoulanteRework

Date

06/2020

Licence :

Robotronik Phelma

Author

NPXav Benano Trukbidule

### 5.1.2 Macro Definition Documentation

#### 5.1.2.1 RCC\_CLOCK\_FREQ\_HZ

```
#define RCC_CLOCK_FREQ_HZ ( 64000000)
```

RCC Clock Frequency [Hz]

#### 5.1.2.2 SYSTICK\_FREQ\_HZ

```
#define SYSTICK_FREQ_HZ ( 100000)
```

Interruptions = 10kHz = 100us (beaucoup ?)

### 5.1.3 Function Documentation

#### 5.1.3.1 clock\_setup()

```
void clock_setup ( )
```

This function setup the system clock.

#### 5.1.3.2 delay\_ms()

```
void delay_ms (
    uint32_t ms )
```

This function implements a delay in ms.



## Parameters

<i>ms</i>	value of delay in ms
-----------	----------------------

## 5.2 lowlevel/include/gpio.h File Reference

This implements the setup of a gpio pin

```
#include <libopencm3/stm32/rcc.h>
#include <libopencm3/stm32/gpio.h>
```

### Functions

- void [gpio\\_setup\\_pin\\_af](#) (enum rcc\_periph\_clken rcc\_clken, uint32\_t gpio\_port, uint16\_t gpio\_pin, uint8\_t gpio\_altfun)

*This function setup a pin for an alternate function.*

#### 5.2.1 Detailed Description

This implements the setup of a gpio pin

This file is part of cdf2020BaseRoulanteRework

## Date

06/2020

## Licence :

Robotronik Phelma

## Author

NPXav Benano Trukbidule

#### 5.2.2 Function Documentation

##### 5.2.2.1 gpio\_setup\_pin\_af()

```
void gpio_setup_pin_af (
    enum rcc_periph_clken rcc_clken,
    uint32_t gpio_port,
    uint16_t gpio_pin,
    uint8_t gpio_altfun )
```

This function setup a pin for an alternate function.

## Parameters

<i>rcc_clken</i>	reset clock control for the pin (usually RCC_X with X the gpio_port)
<i>gpio_port</i>	port of the selected pin
<i>gpio_pin</i>	number of the selected pin
<i>gpio_altfun</i>	identifier for the alternate function (usually GPIO_AFX with X the number for altfun)

## 5.3 lowlevel/include/motor.h File Reference

This implements the functions required to pilot the propulsion motors of the robot

```
#include <libopencm3/stm32/timer.h>
#include "timer.h"
#include "gpio.h"
```

### Macros

- #define **PWM\_PRESCALE** (64)
- #define **PWM\_PERIOD** (20000)
- #define **MOTOR\_TIM\_RCC** RCC\_TIM3
- #define **MOTOR\_TIM** TIM3
- #define **MOTOR\_A\_GPIO\_RCC\_EN** RCC\_GPIOA
- #define **MOTOR\_A\_PORT\_EN** GPIOA
- #define **MOTOR\_A\_PIN\_EN** GPIO4
- #define **MOTOR\_A\_AF** GPIO\_AF2
- #define **MOTOR\_A\_OC\_ID** TIM\_OC2
- #define **MOTOR\_A\_OC\_MODE** TIM\_OCM\_PWM1
- #define **MOTOR\_A\_GPIO\_RCC\_DIR** RCC\_GPIOA
- #define **MOTOR\_A\_PORT\_DIR** GPIOA
- #define **MOTOR\_A\_PIN\_DIR** GPIO3
- #define **MOTOR\_A\_INIT\_DIR** 0
- #define **MOTOR\_B\_GPIO\_RCC\_EN** RCC\_GPIOA
- #define **MOTOR\_B\_PORT\_EN** GPIOA
- #define **MOTOR\_B\_PIN\_EN** GPIO6
- #define **MOTOR\_B\_AF** GPIO\_AF2
- #define **MOTOR\_B\_OC\_ID** TIM\_OC1
- #define **MOTOR\_B\_OC\_MODE** TIM\_OCM\_PWM1
- #define **MOTOR\_B\_GPIO\_RCC\_DIR** RCC\_GPIOA
- #define **MOTOR\_B\_PORT\_DIR** GPIOA
- #define **MOTOR\_B\_PIN\_DIR** GPIO7
- #define **MOTOR\_B\_INIT\_DIR** 0

### Enumerations

- enum **motor\_sel** { **MOTOR\_A**, **MOTOR\_B** }  
*enum of the two motors of the robot to choose which one will be piloted (with function motor\_set)*

## Functions

- void `motor_setup` ()  
*This function initialize the timers and GPIOs to pilot the propulsion motors in our setup by PWM + the GPIOs for the direction.*
- void `motor_set` (enum `motor_sel` sel, int8\_t value)  
*This function pilot the sel (MOTOR\_A or MOTOR\_B) with a value between -100(backward full speed) and +100 (forward full speed)*

### 5.3.1 Detailed Description

This implements the functions required to pilot the propulsion motors of the robot

This file is part of cdf2020BaseRoulanteRework

Date

06/2020

Licence :

Robotronik Phelma

Author

NPXav Benano Trukbidule

### 5.3.2 Macro Definition Documentation

#### 5.3.2.1 PWM\_PERIOD

```
#define PWM_PERIOD (20000)
```

We need a 50 Hz period ( $1000 / 20\text{ms} = 50$ ), thus divide 100000 by 50 = 20000 (us).

#### 5.3.2.2 PWM\_PRESCALE

```
#define PWM_PRESCALE (64)
```

Prescale 64000000 Hz system clock by 64 = 1000000 Hz.

### 5.3.3 Enumeration Type Documentation

### 5.3.3.1 motor\_sel

```
enum motor_sel
```

enum of the two motors of the robot to choose which one will be piloted (with function motor\_set)

## 5.3.4 Function Documentation

### 5.3.4.1 motor\_set()

```
void motor_set (
    enum motor_sel sel,
    int8_t value )
```

This function pilot the sel (MOTOR\_A or MOTOR\_B) with a value between -100(backward full speed) and +100 (forward full speed)

#### Parameters

<i>sel</i>	The motor that will be piloted (eg MOTOR_A)
<i>value</i>	value is between -100 and +100, controls the speed and direction of the motor sel (eg +54)

**Todo** we chose that 0 is forward and 1 is backward, it should be defined in a macro and adjustable for the motors

### 5.3.4.2 motor\_setup()

```
void motor_setup ( )
```

This function initialize the timers and GPIOs to pilot the propulsion motors in our setup by PWM + the GPIOs for the direction.

## 5.4 lowlevel/include/timer.h File Reference

This implements the functions required setup a timer and its output channel

```
#include <stdint.h>
#include <libopencm3/stm32/timer.h>
#include <libopencm3/stm32/rcc.h>
```

## Functions

- void [timer\\_setup](#) (enum rcc\_periph\_clken rcc\_clken, uint32\_t timer\_peripheral, uint32\_t prescaler, uint32\_t period)  
*This function setup an internal timer.*
- void [timer\\_setup\\_output\\_c](#) (uint32\_t timer\_peripheral, enum tim\_oc\_id oc\_id, enum tim\_oc\_mode oc\_mode, uint32\_t oc\_value)  
*This function configure an output channel for the timer.*
- void [timer\\_start](#) (uint32\_t timer\_peripheral)  
*This function start the timer.*

### 5.4.1 Detailed Description

This implements the functions required setup a timer and its output channel

This file is part of cdfr2020BaseRoulanteRework

Date

06/2020

Licence :

Robotronik Phelma

Author

NPXav Benano Trukbidule

### 5.4.2 Function Documentation

#### 5.4.2.1 timer\_setup()

```
void timer_setup (
    enum rcc_periph_clken rcc_clken,
    uint32_t timer_peripheral,
    uint32_t prescaler,
    uint32_t period )
```

This function setup an internal timer.

Parameters

<i>rcc_clken</i>	reset and clock control enable for the timer
<i>timer_peripheral</i>	timer selected
<i>prescaler</i>	prescaling the clock. Dividing the sys_clk by a factor
<i>period</i>	period of the timer in us

Generated by Doxygen

#### 5.4.2.2 timer\_setup\_output\_c()

```
void timer_setup_output_c (
    uint32_t timer_peripheral,
    enum tim_oc_id oc_id,
    enum tim_oc_mode oc_mode,
    uint32_t oc_value )
```

This function configure an output channel for the timer.

##### Parameters

<i>timer_peripheral</i>	selected timer
<i>oc_id</i>	selected channel
<i>oc_mode</i>	different mode used for the timer (here AF_2 for PWM)
<i>oc_value</i>	initial value of the duty cycle

#### 5.4.2.3 timer\_start()

```
void timer_start (
    uint32_t timer_peripheral )
```

This function start the timer.

##### Parameters

<i>timer_peripheral</i>	selected timer
-------------------------	----------------

# Index

clock.h  
    clock\_setup, [12](#)  
    delay\_ms, [12](#)  
    RCC\_CLOCK\_FREQ\_HZ, [12](#)  
    SYSTICK\_FREQ\_HZ, [12](#)  
clock\_setup  
    clock.h, [12](#)  
  
delay\_ms  
    clock.h, [12](#)  
  
gpio.h  
    gpio\_setup\_pin\_af, [13](#)  
gpio\_setup\_pin\_af  
    gpio.h, [13](#)  
  
lowlevel/include/clock.h, [11](#)  
lowlevel/include/gpio.h, [13](#)  
lowlevel/include/motor.h, [14](#)  
lowlevel/include/timer.h, [16](#)  
  
motor.h  
    motor\_sel, [15](#)  
    motor\_set, [16](#)  
    motor\_setup, [16](#)  
    PWM\_PERIOD, [15](#)  
    PWM\_PRESCALE, [15](#)  
motor\_a, [8](#)  
motor\_b, [9](#)  
motor\_sel  
    motor.h, [15](#)  
motor\_set  
    motor.h, [16](#)  
motor\_setup  
    motor.h, [16](#)  
motor\_tim, [7](#)  
  
PWM\_PERIOD  
    motor.h, [15](#)  
PWM\_PRESCALE  
    motor.h, [15](#)  
  
RCC\_CLOCK\_FREQ\_HZ  
    clock.h, [12](#)  
  
SYSTICK\_FREQ\_HZ  
    clock.h, [12](#)  
  
timer.h  
    timer\_setup, [17](#)  
    timer\_setup\_output\_c, [18](#)  
    timer\_start, [18](#)  
timer\_setup  
    timer.h, [17](#)  
timer\_setup\_output\_c  
    timer.h, [18](#)  
timer\_start  
    timer.h, [18](#)