

cdf2020BaseRoulanteRework

Generated by Doxygen 1.8.18

1 Todo List	1
2 Module Index	3
2.1 Modules	3
3 File Index	5
3.1 File List	5
4 Module Documentation	7
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
5 File Documentation	11
5.1 lowlevel/include/clock.h File Reference	11
5.1.1 Detailed Description	11
5.1.2 Function Documentation	12
5.1.2.1 clock_setup()	12
5.1.2.2 delay_ms()	12
5.2 lowlevel/include/gpio.h File Reference	12
5.2.1 Detailed Description	12
5.2.2 Function Documentation	13
5.2.2.1 gpio_setup_pin_af()	13
5.3 lowlevel/include/motor.h File Reference	13
5.3.1 Detailed Description	14
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	15
5.3.4 Function Documentation	15
5.3.4.1 motor_set()	15
5.3.4.2 motor_setup()	16
5.4 lowlevel/include/timer.h File Reference	16
5.4.1 Detailed Description	16
5.4.2 Function Documentation	17
5.4.2.1 timer_setup()	17
5.4.2.2 timer_setup_output_c()	17
5.4.2.3 timer_start()	17
Index	19

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

[12](#)

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

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

[13](#)

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
- #define **MOTOR_A_INVERT_DIR** (-1)

4.2.1 Detailed Description

Definition for the MOTOR_A.

EN stands for enable (output of the PWM signal)

We use OC_ID to select a specific channel of the output comparator as a PWM_output

DIR stands for direction (boolean value)

INIT_DIR is the initial direction of the motor INVERT_DIR allows to define the forward direction in motor_set (must be 1 or -1) Pinmap used here: EN on PA4 (with TIM3_CH2), DIR on PA3

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
- #define **MOTOR_B_INVERT_DIR** (1)

4.3.1 Detailed Description

Definition for the MOTOR_B.

EN stands for enable (output of the PWM signal)

We use OC_ID to select a specific channel of the output comparator as a PWM_output

DIR stands for direction (boolean value)

INIT_DIR is the initial direction of the motor INVERT_DIR allows to define the forward direction in motor_set (must be 1 or -1) Pinmap used here: EN on PA6 (with TIM3_CH1), DIR on PA7

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>
```

Functions

- void `clock_setup` ()
This function setup the system clock.
- uint32_t `clock_get_systicks` ()
This function gets the number of systicks since starting.
- void `delay_ms` (uint32_t ms)
This function gets the uptime 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

PhenixRobotik NPXav Benano Trukbidule

5.1.2 Function Documentation

5.1.2.1 clock_setup()

```
void clock_setup ( )
```

This function setup the system clock.

5.1.2.2 delay_ms()

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

This function gets the uptime in 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 cdfr2020BaseRoulanteRework

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 (usualy 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 (usualy 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_A_INVERT_DIR** (-1)
- #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
- #define **MOTOR_B_INVERT_DIR** (1)

Enumerations

- enum [motor_sel](#) { **MOTOR_A**, **MOTOR_B** }
enum of the two motors used to identify them in some functions (like function motor_set)

Functions

- void [motor_setup](#) ()
This function initializes the timers (including the timer output comparator) and GPIOs to pilot by PWM the propulsion motors + the GPIOs for the direction.
- void [motor_set](#) (enum [motor_sel](#) sel, int8_t value)
This function pilots the sel (MOTOR_A or MOTOR_B) with a value between -100(backward full speed) and +100 (forward full speed). The forward direction depends on the sign of MOTOR_X_INVER_DIR.

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 used to identify them in some functions (like 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 pilots the sel (MOTOR_A or MOTOR_B) with a value between -100(backward full speed) and +100 (forward full speed). The forward direction depends on the sign of MOTOR_X_INVER_DIR.

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 initializes the timers (including the timer output comparator) and GPIOs to pilot by PWM the propulsion motors + 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 with the given parameters.
- 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 the output comparator of a channel for the timer specified.
- void [timer_start](#) (uint32_t timer_peripheral)
This function starts the given timer.

5.4.1 Detailed Description

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

This file is part of cdf2020BaseRoulanteRework

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 with the given parameters.

Parameters

<i>rcc_clken</i>	reset and clock control enable for the timer (clock tree)
<i>timer_peripheral</i>	timer selected
<i>prescaler</i>	the input frequency of the timer (sys_clk) is divided by this factor
<i>period</i>	period of the timer in us

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 the output comparator of a channel for the timer specified.

Parameters

<i>timer_peripheral</i>	selected timer
<i>oc_id</i>	selected channel of the output comparator
<i>oc_mode</i>	different mode used for the timer
<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 starts the given timer.

Parameters

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

Index

clock.h

clock_setup, [12](#)

delay_ms, [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, [12](#)

lowlevel/include/motor.h, [13](#)

lowlevel/include/timer.h, [16](#)

motor.h

motor_sel, [15](#)

motor_set, [15](#)

motor_setup, [16](#)

PWM_PERIOD, [15](#)

PWM_PRESCALE, [15](#)

motor_a, [8](#)

motor_b, [9](#)

motor_sel

motor.h, [15](#)

motor_set

motor.h, [15](#)

motor_setup

motor.h, [16](#)

motor_tim, [7](#)

PWM_PERIOD

motor.h, [15](#)

PWM_PRESCALE

motor.h, [15](#)

timer.h

timer_setup, [17](#)

timer_setup_output_c, [17](#)

timer_start, [17](#)

timer_setup

timer.h, [17](#)

timer_setup_output_c

timer.h, [17](#)

timer_start

timer.h, [17](#)