cdfr2020BaseRoulanteRework

Generated by Doxygen 1.8.18

Chapter 1

Todo List

Member motor_set (enum motor_sel sel, int8_t value)

we chosse 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:

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 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

ms	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

rcc_clken	reset clock control for the pin (usually RCC_X with X the gpio_port)
gpio_port	port of the selected pin
gpio_pin	number of the selected pin
gpio_altfun	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_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 cdfr2020BaseRoulanteRework

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 / 20ms = 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

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

Todo we chosse 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

rcc_clken	reset and clock control enable for the timer
timer_peripheral	timer selected
prescaler	prescaling the clock. Dividing the sys_clk by a factor
period	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 an output channel for the timer.

Parameters

timer_peripheral	selected timer
oc_id	selected channel
oc_mode	different mode used for the timer (here AF_2 for PWM)
oc_value	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

timer_peripheral	selected timer

Index