main.c File Reference

: Main program body More...

```
#include "main.h"
#include "adc.h"
#include "tim.h"
#include "usart.h"
#include "gpio.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
```

Macros

#define **UART_TX_BUFFER_SIZE** 64
Buffer size for transmission used in the shell communication.

#define **UART_RX_BUFFER_SIZE** 1
Buffer size for reception used in the shell communication.

#define **CMD_BUFFER_SIZE** 64
Buffer size for the shell command line.

#define **MAX_ARGS** 9
Maximum number of arguments for a command in the shell script.

#define **ASCII_CR** 0x0D
Defines CR, carriage return command in the shell.

#define **ASCII_DEL** 0x7F
Defines DEL, delete command in the shell

#define **SPEED_MAX** 512
Arbitrary number representing out maximum value for the motor speed.

#define **TIMCLOCK** 170000000
Definition of bus frequency for clock, useful to calculate time.

#define **PRESCALAR** 1
Definition of prescaler clock, useful to calculate time.

Functions

void **SystemClock_Config** (void)
System Clock Configuration. More...

void **HAL_TIM_IC_CaptureCallback** (TIM_HandleTypeDef *htim)

int **main** (void)
The application entry point. More...

void **HAL_UART_RxCpltCallback** (UART_HandleTypeDef *huart)

void **powerUpSequence** (void)

void **HAL_TIM_PeriodElapsedCallback** (TIM_HandleTypeDef *htim)

Period elapsed callback in non blocking mode. More...

void **Error_Handler** (void)

This function is executed in case of error occurrence. More...

Variables

const uint8_t **prompt** [] ="user@Nucleo-STM32G431>>"

Shell prompt text.

const uint8_t **started** []

Startup message when shell is initialized. More...

const uint8_t **newLine** [] ="\r\n"

Defines characters used to create a new line.

const uint8_t **cmdNotFound** [] ="Command not found\r\n"

Shell message for when a user input command is not implemented.

const uint8_t **help** []

Shell message for when the user types "help". More...

const uint8_t **pinout** []

Shell message that lists the pinout list of the microcontroller. More...

const uint8_t **powerOn** []

Shell message when powering on the motor. More...

const uint8_t **powerOff** []

Shell message when powering off the motor. More...

const char * **separators** = " ="

List of separators used to parse the strings in shell.

uint32_t **uartRxReceived**

Set to 1 when a new character is received on uart 2.

uint8_t **uartRxBuffer** [**UART_RX_BUFFER_SIZE**]

A buffer to store the received data from UART.

uint8_t **uartTxBuffer** [**UART_TX_BUFFER_SIZE**]

A buffer to store the UART data to be transmitted.

uint32_t **IC_Val1** = 0

First timer value.

uint32_t **IC_Val2** = 0

Second timer value.

uint32_t **Difference** = 0

Difference of the two timer values, to be able to know the direction of the rotation.

int **Is_First_Captured** = 0

Set to 1 when the first rising edge is captured.

int **Is_First_Captured_2** = 0

Set to 1 when the first rising edge is captured (version 2)

float **frequency** = 0

Declaration of variable to hold the frequency value.

Detailed Description

: Main program body

Attention

Function Documentation

◆ Error_Handler()

void Error_Handler (void)

This function is executed in case of error occurrence.

Return values

None

◆ HAL_TIM_PeriodElapsedCallback()

void HAL_TIM_PeriodElapsedCallback (TIM_HandleTypeDef * htim)

Period elapsed callback in non blocking mode.

Note

This function is called when TIM6 interrupt took place, inside HAL_TIM_IRQHandler(). It makes a direct call to HAL_IncTick() to increment a global variable "uwTick" used as application time base.

Parameters

htim : TIM handle

Return values

None

◆ main()

int main (void)

The application entry point.

Return values

int

< Initializes a cmd list to store the characters sent via shell

< Initializes an index used to parse through the shell characters

< Initializes a variable to hold the arguments of a function

< Initializes a variable to hold the argc value

< Initializes a variable to hold the tokens from strtok function

< Initializes a variable to verify if shell has finished user input

< Initializes a variable to hold the speed value

< Initializes a variable to hold the value in the compare register channel 1

< Initializes a variable to hold the value in the compare register channel 2

< Initializes a variable to hold the value form the ADC conversion for the current

◆ powerUpSequence()

void powerUpSequence (void)

Sends the required sequence to power up the motor

Parameters

`[in]` **None**

`[out]` **None**

◆ 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

Variable Documentation

◆ help

const uint8_t help[]

Initial value:

```
=
        "\r\n*-----------------------------*"
        "\r\n| Help Menu |"
        "\r\n*-----------------------------*"
        "\r\n*set PA5 1 : Turns ON the LED*"
        "\r\n*set PA5 0 : Turns OFF the LED*"
        "\r\n*get : Prints the current and the frequency in the motor*"
        "\r\n*pinout : Prints the pinout list*"
        "\r\n*start : Starts the motor*"
        "\r\n*speed x : Sets the speed of the motor to x (-512 < x < 512)*"
        "\r\n"
```

Shell message for when the user types "help".

◆ pinout

const uint8_t pinout[]

Initial value:

```
=
        "\r\n*-----------------------------*"
        "\r\n| Pinout List |"
        "\r\n*-----------------------------*"
        "\r\n*PA0: ADC_CURRENT*"
        "\r\n*PA5: LED*"
        "\r\n*PA8: TIM1_CH1*"
        "\r\n*PA9: TIM1_CH2*"
        "\r\n*PA11: TIM1_CH1N*"
        "\r\n*PA12: TIM1_CH2N*"
        "\r\n*PC3: ISO_RESET*"
        "\r\n"
```

Shell message that lists the pinout list of the microcontroller.

◆ powerOff

const uint8_t powerOff[]

Initial value:

```
=
        "\r\n*-----------------------------*"
        "\r\n| Motor OFF |"
        "\r\n"
```

Shell message when powering off the motor.

◆ powerOn

const uint8_t powerOn[]

Initial value:

```
=
        "\r\n*-----------------------------*"
        "\r\n| Motor ON |"
        "\r\n"
```

Shell message when powering on the motor.

◆ started

const uint8_t started[]

Initial value:

```
=
        "\r\n*-----------------------------*"
        "\r\n| Welcome on Nucleo-STM32G431 |"
        "\r\n*-----------------------------*"
        "\r\n"
```

Startup message when shell is initialized.