libdorobo32

1.0.0

Generated by Doxygen 1.8.9.1

Fri May 6 2016 11:28:24

Contents

Chapter 1

Todo List

globalScope> **Global** **adc_init** **(void)**

Please implement me!

globalScope> **Global** **fft_init** **(void)**

Please implement me!

globalScope> **Global** **motor_init** **(void)**

Please implement me!

globalScope> **Global** **uart_init** **(void)**

Please implement me!

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Data Structure Documentation

4.1 motor_t Struct Reference

Data Fields

- GPIO_TypeDef * controlPORT1
- uint16_t controlPIN1
- GPIO_TypeDef * controlPORT2
- uint16_t controlPIN2
- uint32_t timerChannel
- int8_t speed

4.1.1 Detailed Description

Definition at line 18 of file motor.c.

4.1.2 Field Documentation

4.1.2.1 uint16_t controlPIN1

Definition at line 21 of file motor.c.

4.1.2.2 uint16_t controlPIN2

Definition at line 23 of file motor.c.

4.1.2.3 GPIO_TypeDef* controlPORT1

Definition at line 20 of file motor.c.

4.1.2.4 GPIO_TypeDef* controlPORT2

Definition at line 22 of file motor.c.

4.1.2.5 int8_t speed

Definition at line 25 of file motor.c.

4.1.2.6 uint32_t timerChannel

Definition at line 24 of file motor.c.

The documentation for this struct was generated from the following file:

- dorobo32/src/motor.c

4.2 pin_t Struct Reference

Data Fields

- GPIO_TypeDef * port
- uint16_t pin

4.2.1 Detailed Description

DoroboLib32 Digital IO

Funktionen für die Manipulation der digitalen IO Pins.

Copyright (c) 2016 Laurent Schröder, Claus Führer, Michael Hoffmann

Definition at line 15 of file digitial.c.

4.2.2 Field Documentation

4.2.2.1 uint16_t pin

Definition at line 18 of file digitial.c.

4.2.2.2 GPIO_TypeDef* port

Definition at line 17 of file digitial.c.

The documentation for this struct was generated from the following file:

- dorobo32/src/digitial.c

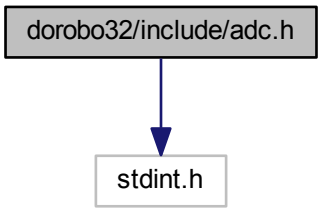
Chapter 5

File Documentation

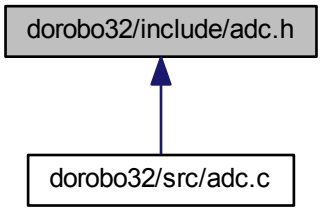
5.1 dorobo32/include/adc.h File Reference

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

Include dependency graph for adc.h:



This graph shows which files directly or indirectly include this file:



Enumerations

- enum DA_ADC_CHANNEL_E {
  DA_ADC_CHANNEL0, DA_ADC_CHANNEL1, DA_ADC_CHANNEL2, DA_ADC_CHANNEL3,
  DA_ADC_CHANNEL4, DA_ADC_CHANNEL5, DA_ADC_CHANNEL6, DA_ADC_CHANNEL7,
  DA_ADC_CHANNEL8, DA_ADC_CHANNEL9 }

Functions

- void adc_init (void)

  Adc initialization.
- uint32_t adc_get_value (enum DA_ADC_CHANNEL_E adc_channel_no)

  Get converted analog value from analog pin adc_channel_no.

5.1.1 Detailed Description

DoroboLib32 ADC (DA)

Functions for reading analog voltages using the analog digital converter.

5.1.2 Enumeration Type Documentation

5.1.2.1 enum DA_ADC_CHANNEL_E

Dorobo32 analog input channels

Enumerator

DA_ADC_CHANNEL0	analog channel 0
DA_ADC_CHANNEL1	analog channel 1
DA_ADC_CHANNEL2	analog channel 2
DA_ADC_CHANNEL3	analog channel 3
DA_ADC_CHANNEL4	analog channel 4
DA_ADC_CHANNEL5	analog channel 5
DA_ADC_CHANNEL6	analog channel 6
DA_ADC_CHANNEL7	analog channel 7
DA_ADC_CHANNEL8	analog channel 8
DA_ADC_CHANNEL9	analog channel 9

Definition at line 18 of file adc.h.

5.1.3 Function Documentation

5.1.3.1 uint32_t adc_get_value (enum DA_ADC_CHANNEL_E *adc_channel_no*)

Get converted analog value from analog pin *adc_channel_no*.

Parameters

adc_channel_no	The adc channel to be read.

Returns

Converted analog 12 bit value.

Definition at line 21 of file adc.c.

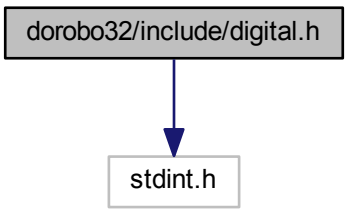
5.1.3.2 void adc_init (void)

Adc initialization.

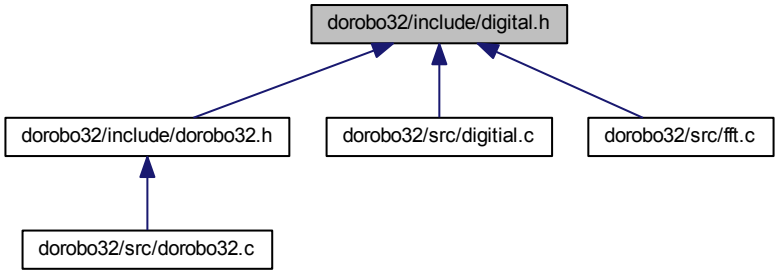
Todo Please implement me!

5.2 dorobo32/include/digital.h File Reference

`#include <stdint.h>`
Include dependency graph for digital.h:



This graph shows which files directly or indirectly include this file:



Enumerations

- enum DD_PINS_E {
  DD_PIN_PD14, DD_PIN_PD15, DD_PIN_PC8, DD_PIN_PC9,
  DD_PIN_PA8, DD_PIN_PC13, DD_PIN_PB11, DD_PIN_PB10,
  DD_PIN_PE14, DD_PIN_PE11, DD_PIN_PE9, DD_PIN_PB1,
  DD_PIN_PF10, DD_PIN_PF9, DD_PIN_PE1, DD_PIN_PE0,
  DD_PIN_PB3, DD_PIN_PA15 }
- enum DD_DIPS_E { DD_DIP1, DD_DIP2, DD_DIP3, DD_DIP4 }
- enum DD_PINLEVEL_E { DD_LEVEL_LOW, DD_LEVEL_HIGH }
- enum DD_PINCONFIG_E { DD_CFG_OUTPUT, DD_CFG_INPUT_PULLUP, DD_CFG_INPUT_PULLDOWN, DD_CFG_INPUT_NOPULL }

Functions

- void digital_init (void)

  Initialize the Dorobo32 Digital I/O (DD) module.
- void digital_configure_pin (enum DD_PINS_E pin_no, enum DD_PINCONFIG_E direction)

  Configure digital pin direction and pullup/pulldown resistors.
- void digital_set_pin (enum DD_PINS_E pin_no, enum DD_PINLEVEL_E level)

  Set pin level of pin pin_no.
- enum DD_PINLEVEL_E digital_get_pin (enum DD_PINS_E pin_no)

  Get signal level for a pin.
- enum DD_PINLEVEL_E digital_get_dip (enum DD_DIPS_E dip_no)

  Get position of a dip switch.

5.2.1 Detailed Description

DoroboLib32 Digital IO (DD)

Functions to read and set digitial IO pins.

5.2.2 Enumeration Type Documentation

5.2.2.1 enum DD_DIPS_E

Dorobo32 DIP switches

Enumerator

DD_DIP1 dip switch 1

DD_DIP2 dip switch 2

DD_DIP3 dip switch 3

DD_DIP4 dip switch 4

Definition at line 45 of file digital.h.

5.2.2.2 enum DD_PINCONFIG_E

Pin configuration

Enumerator

DD_CFG_OUTPUT pin configured as output

DD_CFG_INPUT_PULLUP pin configured as input with internal pullup enabled

DD_CFG_INPUT_PULLDOWN pin configured as input with internal pulldown enabled

DD_CFG_INPUT_NOPULL pin configured as input without internal pullup or pulldown resistor

Definition at line 65 of file digital.h.

5.2.2.3 enum DD_PINLEVEL_E

Pin signal levels

Enumerator

DD_LEVEL_LOW digital low

DD_LEVEL_HIGH digital high

Definition at line 56 of file digital.h.

5.2.2.4 enum DD_PINS_E

Dorobo32 pins available for digital i/o

Enumerator

DD_PIN_PD14 digital i/o pin D14

DD_PIN_PD15 digital i/o pin PD15

DD_PIN_PC8 digital i/o pin PC8

DD_PIN_PC9 digital i/o pin PC9

DD_PIN_PA8 digital i/o pin PA8

DD_PIN_PC13 digital i/o pin PC13

DD_PIN_PB11 digital i/o pin PB11

DD_PIN_PB10 digital i/o pin PB10

DD_PIN_PE14 digital i/o pin PE14

DD_PIN_PE11 digital i/o pin PE11

DD_PIN_PE9 digital i/o pin PE9

DD_PIN_PB1 digital i/o pin PB1

DD_PIN_PF10 digital i/o pin F10

DD_PIN_PF9 digital i/o pin F9

DD_PIN_PE1 digital i/o pin E1

DD_PIN_PE0 digital i/o pin E0

DD_PIN_PB3 digital i/o pin B3

DD_PIN_PA15 digital i/o pin A15

Definition at line 18 of file digital.h.

5.2.3 Function Documentation

5.2.3.1 void digital_configure_pin (enum DD_PINS_E *pin_no,* enum DD_PINCONFIG_E *direction*)

Configure digital pin direction and pullup/pulldown resistors.

Parameters

pin_no	Pin to configure
direction	Predefined configuration as defined in DD_PINCONFIG_E

Definition at line 30 of file digitial.c.

5.2.3.2 enum DD_PINLEVEL_E digital_get_dip (enum DD_DIPS_E *dip_no*)

Get position of a dip switch.

Parameters

dip_no	Dip to read from

Returns

Dip signal level as defined in DD_PINLEVEL_E

Definition at line 83 of file digitial.c.

5.2.3.3 enum DD_PINLEVEL_E digital_get_pin (enum DD_PINS_E *pin_no*)

Get signal level for a pin.

Parameters

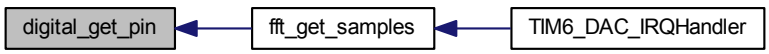
pin_no	Pin to read from

Returns

Pin signal level as defined in DD_PINLEVEL_E

Definition at line 76 of file digitial.c.

Here is the caller graph for this function:



5.2.3.4 void digital_init (void)

Initialize the Dorobo32 Digital I/O (DD) module.

Definition at line 25 of file digitial.c.

Here is the caller graph for this function:

5.2.3.5 void digital_set_pin (enum DD_PINS_E *pin_no*, enum DD_PINLEVEL_E *level*)

Set pin level of pin *pin_no*.

Parameters

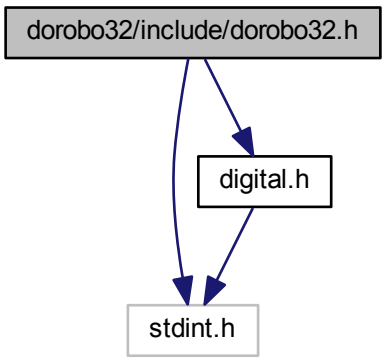
pin_no	Digital pin that is to be set
level	The desired pin level. Values can be DD_LEVEL_LOW or DD_LEVEL_HIGH

Definition at line 69 of file digitial.c.

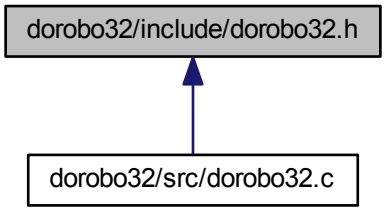
5.3 dorobo32/include/dorobo32.h File Reference

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

Include dependency graph for dorobo32.h:



This graph shows which files directly or indirectly include this file:



Functions

- void dorobo_init (void)

  Board initialization and module initialization for digital i/o and uart.

- void led_green (enum DD_PINLEVEL_E level)

  Set the green on-board led on or off.

- void led_red (enum DD_PINLEVEL_E level)

  Set the red on-board led on or off.

5.3.1 Detailed Description

DoroboLib32 central header

Initialization of the board and the most common modules and control of the on-board LEDs.

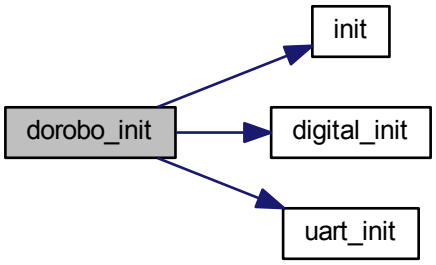
5.3.2 Function Documentation

5.3.2.1 void dorobo_init (void)

Board initialization and module initialization for digital i/o and uart.

Definition at line 19 of file dorobo32.c.

Here is the call graph for this function:



5.3.2.2 void led_green (enum DD_PINLEVEL_E *level*)

Set the green on-board led on or off.

Parameters

level	Level as defined in DD_PINLEVEL_E

Definition at line 26 of file dorobo32.c.

5.3.2.3 void led_red (enum DD_PINLEVEL_E *level*)

Set the red on-board led on or off.

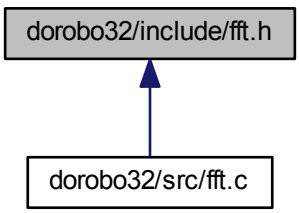
Parameters

level	Level as defined in DD_PINLEVEL_E

Definition at line 31 of file dorobo32.c.

5.4 dorobo32/include/fft.h File Reference

This graph shows which files directly or indirectly include this file:



Enumerations

- enum DFFT_FFT_FREQ_E { FFT_FREQ100, FFT_FREQ125 }

Functions

- void fft_init (void)
  FFT initialization.
- uint16_t fft_simple (enum DFFT_FFT_FREQ_E fft_freq, enum DD_PINS_E pin_no)
  Start sampling from digital pin_no and perform fft over the samples.

5.4.1 Detailed Description

DoroboLib32 FFT (DFFT)

Simple implementation of a fast fourier transform (fft) to detect 100 Hz and 125 Hz modulated signals.

5.4.2 Enumeration Type Documentation

5.4.2.1 enum DFFT_FFT_FREQ_E

The signals that can be detected by the function fft_simple()

Enumerator

FFT_FREQ100 100Hz

FFT_FREQ125 125Hz

Definition at line 18 of file fft.h.

5.4.3 Function Documentation

5.4.3.1 void fft_init (void)

FFT initialization.

Todo Please implement me!

5.4.3.2 uint16_t fft_simple (enum DFFT_FFT_FREQ_E *fft_freq*, enum DD_PINS_E *pin_no*)

Start sampling from digital *pin_no* and perform fft over the samples.

Parameters

fft_freq	The frequency that needs to be detected
pin_no	Digital pin that is used for sampling

Returns

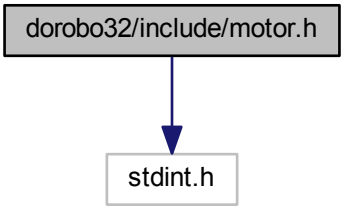
The compliance of the measured signal with the signal that is to be detected. A value above 1,200 indicates compliance.

Definition at line 45 of file fft.c.

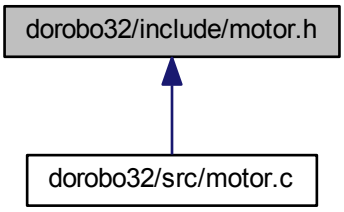
5.5 dorobo32/include/motor.h File Reference

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

Include dependency graph for motor.h:



This graph shows which files directly or indirectly include this file:



Enumerations

- enum DM_MOTORS_E { DM_MOTOR0, DM_MOTOR1, DM_MOTOR2, DM_MOTOR3 }

Functions

- void motor_init (void)

  Motor initialization.
- void motor_set (enum DM_MOTORS_E motor, int8_t speed)

  Set the speed for the given motor.

5.5.1 Detailed Description

DoroboLib32 Motor (DM)

Functions to set motor speed.

5.5.2 Enumeration Type Documentation

5.5.2.1 enum DM_MOTORS_E

Dorobo32 motors

Enumerator

DM_MOTOR0 Motor 0.

DM_MOTOR1 Motor 1.

DM_MOTOR2 Motor 2.

DM_MOTOR3 Motor 3.

Definition at line 19 of file motor.h.

5.5.3 Function Documentation

5.5.3.1 void motor_init (void)

Motor initialization.

Todo Please implement me!

5.5.3.2 void motor_set (enum DM_MOTORS_E *motor*, int8_t *speed*)

Set the speed for the given motor.

Warning: The user of this function is responsible for limiting the motor current by controlling the slope of speed changes.

Parameters

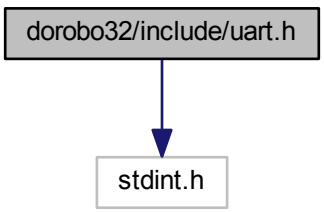
motor	Motor to control
speed	New motor speed and direction in percent (-100-100)

Definition at line 64 of file motor.c.

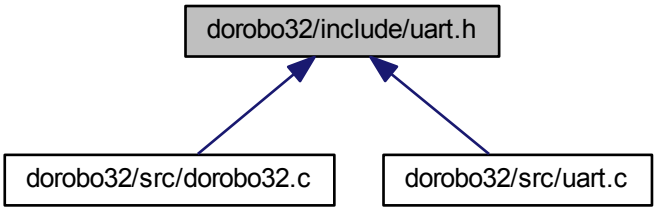
5.6 dorobo32/include/uart.h File Reference

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

Include dependency graph for uart.h:



This graph shows which files directly or indirectly include this file:



Enumerations

- enum DUART_UART_E { UART1, UART2 }

Functions

- void uart_init (void)

  UART initialization.
- void uart_send (enum DUART_UART_E uart, char ∗string)

  Send string via UART.
- void uart_send_buffer (enum DUART_UART_E uart, uint8_t ∗pbuffer, uint16_t size)

  Send buffer via UART.
- void uart_receive (enum DUART_UART_E uart, uint8_t ∗pbuffer, uint16_t size)

  Receive messages from UART.

5.6.1 Detailed Description

DoroboLib32 UART (DUART)

Functions for UART communication.

Copyright (c) 2016 Laurent Schröder, Claus Führer, Michael Hoffmann

5.6.2 Enumeration Type Documentation

5.6.2.1 enum DUART_UART_E

Dorobo32 uarts

Enumerator

UART1 UART1 instance.

UART2 UART2 instance.

Definition at line 19 of file uart.h.

5.6.3 Function Documentation

5.6.3.1 void uart_init (void)

UART initialization.

Todo Please implement me!

Here is the caller graph for this function:

5.6.3.2 void uart_receive (enum DUART_UART_E *uart*, uint8_t * *pbuffer*, uint16_t *size*)

Receive messages from UART.

Parameters

uart	UART instance to be used as defined in DUART_UART_E
**pbuffer*	Buffer to store received data
size	Number of bytes to be received

Definition at line 34 of file uart.c.

5.6.3.3 void uart_send (enum DUART_UART_E *uart*, char * *string*)

Send string via UART.

Parameters

uart	The UART instance to be used as defined in DUART_UART_E
string	char* string to be transmitted

Definition at line 20 of file uart.c.

5.6.3.4 void uart_send_buffer (enum DUART_UART_E *uart*, uint8_t * *pbuffer*, uint16_t *size*)

Send buffer via UART.

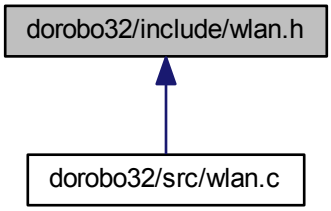
Parameters

uart	The UART instance to be used as defined in DUART_UART_E
**pbuffer*	Pointer to buffer
size	Number of character to be sent

Definition at line 27 of file uart.c.

5.7 dorobo32/include/wlan.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void wlan_init (void)

  Wlan initialization.

5.7.1 Detailed Description

DoroboLib32 WLAN (WI)

Functions for wlan init and std i/o stream redirection.

Copyright (c) 2016 Michael Hoffmann, Claus Führer

5.7.2 Function Documentation

5.7.2.1 void wlan_init (void)

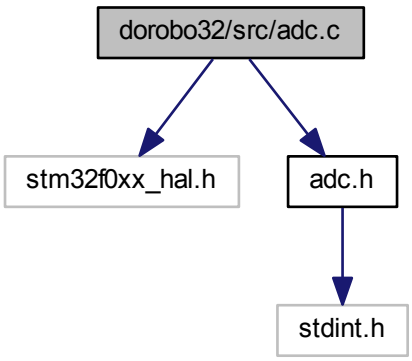
Wlan initialization.

Definition at line 17 of file wlan.c.

5.8 dorobo32/src/adc.c File Reference

```
#include "stm32f0xx_hal.h"
#include "adc.h"
```

Include dependency graph for adc.c:



Functions

- uint32_t adc_get_value (enum DA_ADC_CHANNEL_E adc_channel_no)

  Get converted analog value from analog pin adc_channel_no.

Variables

- ADC_HandleTypeDef hadc
- ADC_ChannelConfTypeDef sConfig

5.8.1 Function Documentation

5.8.1.1 uint32_t adc_get_value (enum DA_ADC_CHANNEL_E *adc_channel_no*)

Get converted analog value from analog pin *adc_channel_no*.

Parameters

adc_channel_no	The adc channel to be read.

Returns

Converted analog 12 bit value.

Definition at line 21 of file adc.c.

5.8.2 Variable Documentation

5.8.2.1 ADC_HandleTypeDef hadc

DoroboLib32 ADC (DA)

Functions for reading analog voltages using the analog digital converter.

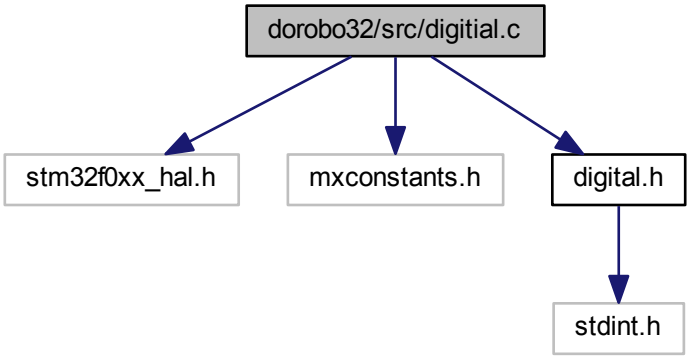
5.8.2.2 ADC_ChannelConfTypeDef sConfig

Definition at line 14 of file adc.c.

5.9 dorobo32/src/digitial.c File Reference

```
#include "stm32f0xx_hal.h"
#include "mxconstants.h"
#include "digital.h"
```

Include dependency graph for digitial.c:



Data Structures

- struct pin_t

Functions

- void digital_init (void)

  Initialize the Dorobo32 Digital I/O (DD) module.
- void digital_configure_pin (enum DD_PINS_E pin_no, enum DD_PINCONFIG_E direction)

  Configure digital pin direction and pullup/pulldown resistors.
- void digital_set_pin (enum DD_PINS_E pin_no, enum DD_PINLEVEL_E level)

  Set pin level of pin pin_no.
- enum DD_PINLEVEL_E digital_get_pin (enum DD_PINS_E pin_no)

  Get signal level for a pin.
- enum DD_PINLEVEL_E digital_get_dip (enum DD_DIPS_E dip_no)

  Get position of a dip switch.

5.9.1 Function Documentation

5.9.1.1 void digital_configure_pin (enum DD_PINS_E *pin_no,* enum DD_PINCONFIG_E *direction*)

Configure digital pin direction and pullup/pulldown resistors.

Parameters

pin_no	Pin to configure
direction	Predefined configuration as defined in DD_PINCONFIG_E

Definition at line 30 of file digitial.c.

5.9.1.2 enum DD_PINLEVEL_E digital_get_dip (enum DD_DIPS_E *dip_no*)

Get position of a dip switch.

Parameters

dip_no	Dip to read from

Returns

Dip signal level as defined in DD_PINLEVEL_E

Definition at line 83 of file digitial.c.

5.9.1.3 enum DD_PINLEVEL_E digital_get_pin (enum DD_PINS_E *pin_no*)

Get signal level for a pin.

Parameters

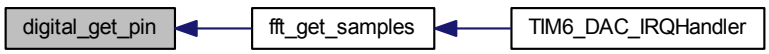
pin_no	Pin to read from

Returns

Pin signal level as defined in DD_PINLEVEL_E

Definition at line 76 of file digitial.c.

Here is the caller graph for this function:



5.9.1.4 void digital_init (void)

Initialize the Dorobo32 Digital I/O (DD) module.

Definition at line 25 of file digitial.c.

Here is the caller graph for this function:

5.9.1.5 void digital_set_pin (enum DD_PINS_E *pin_no*, enum DD_PINLEVEL_E *level*)

Set pin level of pin *pin_no*.

Parameters

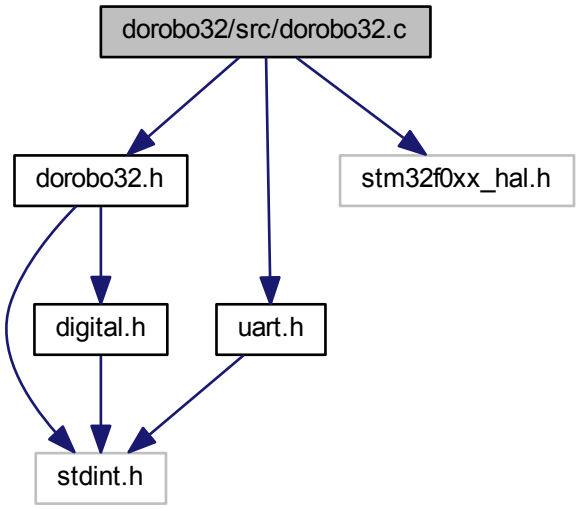
pin_no	Digital pin that is to be set
level	The desired pin level. Values can be DD_LEVEL_LOW or DD_LEVEL_HIGH

Definition at line 69 of file digitial.c.

5.10 dorobo32/src/dorobo32.c File Reference

```
#include "dorobo32.h"
#include "uart.h"
#include "stm32f0xx_hal.h"
```

Include dependency graph for dorobo32.c:



Functions

- void init ()
- void dorobo_init (void)

  Board initialization and module initialization for digital i/o and uart.
- void led_green (enum DD_PINLEVEL_E level)

  Set the green on-board led on or off.
- void led_red (enum DD_PINLEVEL_E level)

  Set the red on-board led on or off.

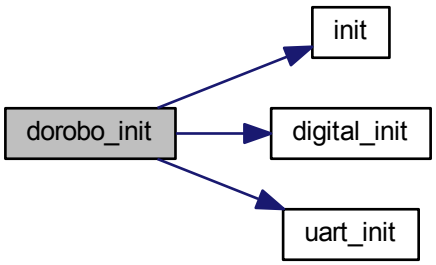
5.10.1 Function Documentation

5.10.1.1 void dorobo_init (void)

Board initialization and module initialization for digital i/o and uart.

Definition at line 19 of file dorobo32.c.

Here is the call graph for this function:



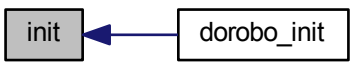
5.10.1.2 void init ( )

DoroboLib32 Sammel-Initialisierung

Initialisierung für die DoroboLib32 und die gebräuchlichsten Module.

Copyright (c) 2016 Laurent Schröder, Claus Führer, Michael Hoffmann

Here is the caller graph for this function:



5.10.1.3 void led_green (enum DD_PINLEVEL_E *level*)

Set the green on-board led on or off.

Parameters

level	Level as defined in DD_PINLEVEL_E

Definition at line 26 of file dorobo32.c.

5.10.1.4 void led_red (enum DD_PINLEVEL_E *level*)

Set the red on-board led on or off.

Parameters

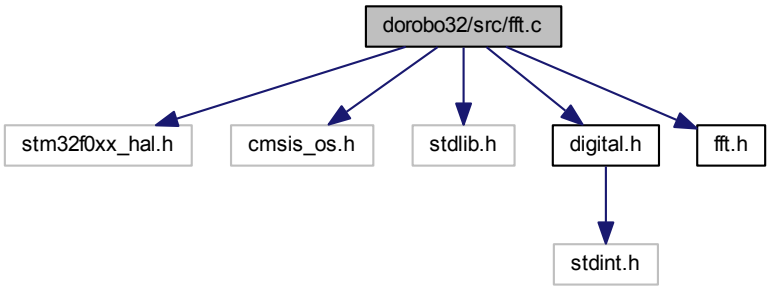
level	Level as defined in DD_PINLEVEL_E

Definition at line 31 of file dorobo32.c.

5.11 dorobo32/src/fft.c File Reference

```
#include "stm32f0xx_hal.h"
#include "cmsis_os.h"
#include <stdlib.h>
#include "digital.h"
#include "fft.h"
```

Include dependency graph for fft.c:



Macros

- #define SAMPLES 40

Functions

- uint16_t fft_simple (enum DFFT_FFT_FREQ_E efft_freq, enum DD_PINS_E pin_no)

  Start sampling from digital pin_no and perform fft over the samples.
- void fft_get_samples ()
- void TIM6_DAC_IRQHandler (void)

  This function handles TIM6 global and DAC channel underrun error interrupts.

Variables

- TIM_HandleTypeDef htim6
- uint8_t fft_sample_index = 0
- uint8_t fft_samples_ready = 0
- enum DD_PINS_E sample_pin

5.11.1 Macro Definition Documentation

5.11.1.1 #define SAMPLES 40

Definition at line 21 of file fft.c.

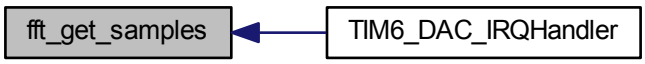
5.11.2 Function Documentation

5.11.2.1 void fft_get_samples ( )

Definition at line 85 of file fft.c.

Here is the call graph for this function:

Here is the caller graph for this function:



5.11.2.2 uint16_t fft_simple (enum DFFT_FFT_FREQ_E *fft_freq*, enum DD_PINS_E *pin_no*)

Start sampling from digital *pin_no* and perform fft over the samples.

Parameters

fft_freq	The frequency that needs to be detected

pin_no	Digital pin that is used for sampling

Returns

The compliance of the measured signal with the signal that is to be detected. A value above 1,200 indicates compliance.

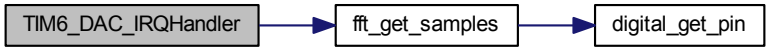
Definition at line 45 of file fft.c.

5.11.2.3 void TIM6_DAC_IRQHandler (void)

This function handles TIM6 global and DAC channel underrun error interrupts.

Definition at line 103 of file fft.c.

Here is the call graph for this function:



5.11.3 Variable Documentation

5.11.3.1 uint8_t fft_sample_index = 0

Definition at line 22 of file fft.c.

5.11.3.2 uint8_t fft_samples_ready = 0

Definition at line 23 of file fft.c.

5.11.3.3 TIM_HandleTypeDef htim6

DoroboLib32 FFT (DFFT)

Simple implementation of a fast fourier transform (fft) to detect 100 Hz and 125 Hz modulated signals.

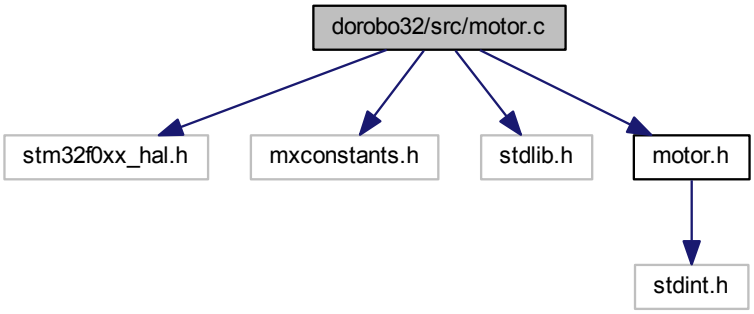
5.11.3.4 enum DD_PINS_E sample_pin

Definition at line 24 of file fft.c.

5.12 dorobo32/src/motor.c File Reference

```
#include "stm32f0xx_hal.h"
#include "mxconstants.h"
#include <stdlib.h>
#include "motor.h"
```

Include dependency graph for motor.c:



Data Structures

- struct motor_t

Functions

- void motor_set (enum DM_MOTORS_E motoren, int8_t speed)

  Set the speed for the given motor.

Variables

- TIM_HandleTypeDef htim3
- TIM_OC_InitTypeDef sConfigOC

5.12.1 Function Documentation

5.12.1.1 void motor_set (enum DM_MOTORS_E *motor,* int8_t *speed*)

Set the speed for the given motor.

Warning: The user of this function is responsible for limiting the motor current by controlling the slope of speed changes.

Parameters

motor	Motor to control
speed	New motor speed and direction in percent (-100-100)

Definition at line 64 of file motor.c.

5.12.2 Variable Documentation

5.12.2.1 TIM_HandleTypeDef htim3

DoroboLib32 Motor (DM)

Functions to set motor speed.

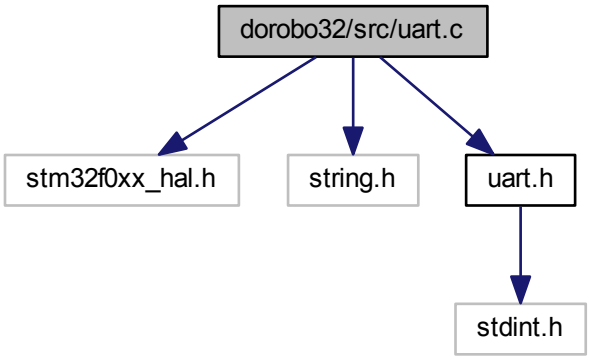
5.12.2.2 TIM_OC_InitTypeDef sConfigOC

Definition at line 16 of file motor.c.

5.13 dorobo32/src/uart.c File Reference

```
#include "stm32f0xx_hal.h"
#include "string.h"
#include "uart.h"
```

Include dependency graph for uart.c:



Functions

- void uart_send (enum DUART_UART_E euart, char *msg)

  Send string via UART.
- void uart_send_buffer (enum DUART_UART_E uart, uint8_t *pbuffer, uint16_t size)

  Send buffer via UART.
- void uart_receive (enum DUART_UART_E euart, uint8_t *pbuffer, uint16_t size)

  Receive messages from UART.
- void USART1_IRQHandler (void)
- void USART2_IRQHandler (void)

Variables

- UART_HandleTypeDef huart1
- UART_HandleTypeDef huart2

5.13.1 Function Documentation

5.13.1.1 void uart_receive (enum DUART_UART_E *uart*, uint8_t * *pbuffer*, uint16_t *size*)

Receive messages from UART.

Parameters

uart	UART instance to be used as defined in DUART_UART_E
∗pbuffer	Buffer to store received data
size	Number of bytes to be received

Definition at line 34 of file uart.c.

5.13.1.2 void uart_send (enum DUART_UART_E *uart,* char ∗ *string*)

Send string via UART.

Parameters

uart	The UART instance to be used as defined in DUART_UART_E
string	char∗ string to be transmitted

Definition at line 20 of file uart.c.

5.13.1.3 void uart_send_buffer (enum DUART_UART_E *uart,* uint8_t ∗ *pbuffer,* uint16_t *size*)

Send buffer via UART.

Parameters

uart	The UART instance to be used as defined in DUART_UART_E
∗pbuffer	Pointer to buffer
size	Number of character to be sent

Definition at line 27 of file uart.c.

5.13.1.4 void USART1_IRQHandler (void)

Definition at line 65 of file uart.c.

5.13.1.5 void USART2_IRQHandler (void)

Definition at line 70 of file uart.c.

5.13.2 Variable Documentation

5.13.2.1 UART_HandleTypeDef huart1

DoroboLib32 UART (DUART)

Functions for UART communication.

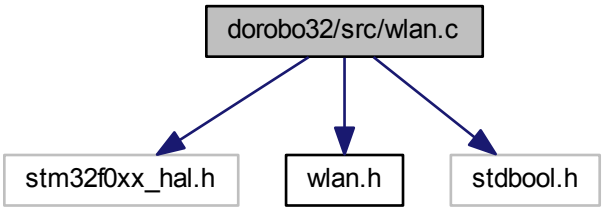
Copyright (c) 2016 Laurent Schröder, Claus Führer, Michael Hoffmann

5.13.2.2 UART_HandleTypeDef huart2

5.14 dorobo32/src/wlan.c File Reference

```
#include "stm32f0xx_hal.h"
#include "wlan.h"
#include <stdbool.h>
```

Include dependency graph for wlan.c:



Functions

- void wlan_init (void)

  Wlan initialization.

5.14.1 Function Documentation

5.14.1.1 void wlan_init (void)

Wlan initialization.

Definition at line 17 of file wlan.c.

Index