

WABL Documentation

Generated by Doxygen 1.8.6

Tue May 19 2015 23:26:40

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	comm.h File Reference	3
2.1.1	Detailed Description	3
2.2	encoder.h File Reference	3
2.2.1	Detailed Description	3
2.2.2	Function Documentation	3
2.2.2.1	encoder_get_x	3
2.2.2.2	encoder_get_x_dot	4
2.2.2.3	encoder_update	4
2.3	io_defs.h File Reference	4
2.3.1	Detailed Description	4
2.4	lqr.h File Reference	4
2.4.1	Detailed Description	4
2.4.2	Function Documentation	4
2.4.2.1	lqr	4
2.5	main.c File Reference	5
2.5.1	Detailed Description	5
2.6	modules.h File Reference	5
2.6.1	Detailed Description	5
2.7	motor.h File Reference	6
2.7.1	Detailed Description	6
2.7.2	Function Documentation	6
2.7.2.1	get_current_value	6
2.7.2.2	motor_pid	6
2.8	orient.h File Reference	6
2.8.1	Detailed Description	7
2.8.2	Function Documentation	7
2.8.2.1	orient_calibrate	7
2.8.2.2	orient_get_phi	7
2.8.2.3	orient_get_phi_dot	7
2.8.2.4	orient_get_phi_dot_raw	8
2.8.2.5	orient_get_phi_raw	8
2.8.2.6	orient_init	8
2.8.2.7	orient_read_reg16	8
2.8.2.8	orient_update	8
2.8.2.9	orient_write_reg	8
2.9	safety.h File Reference	8
2.9.1	Detailed Description	9

2.9.2	Function Documentation	9
2.9.2.1	safety_battery_critical	9
2.10	sound.h File Reference	9
2.10.1	Detailed Description	9
2.10.2	Function Documentation	9
2.10.2.1	sound_play_melody	9
2.11	uart.h File Reference	9
2.11.1	Detailed Description	10
Index		11

Chapter 1

File Index

1.1 File List

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

comm.h	3
encoder.h	3
io_defs.h	4
lqr.h	4
main.c	5
modules.h	5
motor.h	6
orient.h	6
safety.h	8
sound.h	9
uart.h	9

Chapter 2

File Documentation

2.1 comm.h File Reference

2.1.1 Detailed Description

Module for handling terminal serial communication and commands

Author

Stephen Papierski stephenpapierski@gmail.com

Date

2015-05-13 17:02:22 2015-05-13 18:20:43

2.2 encoder.h File Reference

Functions

- void [encoder_init](#) (void)
Setup encoder pins, must be called once.
- void [encoder_update](#) (uint8_t delta_t_ms)
Calculate new position (x) and velocity (x_dot) values.
- double [encoder_get_x](#) (unsigned char motor)
Get position (x) for specified motor in millimeters.
- double [encoder_get_x_dot](#) (unsigned char motor)
Get velocity (x_dot) for specified motor in millimeters/second.

2.2.1 Detailed Description

Motor quadrature encoder module

Author

Stephen Papierski stephenpapierski@gmail.com

Date

2015-04-28 20:19:15 2015-05-19 22:13:38

2.2.2 Function Documentation

2.2.2.1 double encoder_get_x (unsigned char *motor*)

Get position (x) for specified motor in millimeters.

Parameters

<i>motor</i>	MOTOR1 or MOTOR2
--------------	------------------

Returns

position (x) in millimeters

2.2.2.2 double encoder_get_x_dot (unsigned char *motor*)

Get velocity (x_dot) for specified motor in millimeters/second.

Parameters

<i>motor</i>	MOTOR1 or MOTOR2
--------------	------------------

Returns

velocity (x) in millimeters/second

2.2.2.3 void encoder_update (uint8_t *delta_t_ms*)

Calculate new position (x) and velocity (x_dot) values.

Parameters

<i>delta_t_ms</i>	Time difference since last call in milliseconds
-------------------	---

2.3 io_defs.h File Reference

2.3.1 Detailed Description

Author

Stephen Papierski stephenpapierski@gmail.com

Date

2015-03-23 01:20:09 2015-04-28 09:19:03

2.4 lqr.h File Reference

#include "lqr.h"

Functions

- float [lqr](#) (float positionRef_mm, float position_mm, float velocity_mm_s, float angle_mrad, float a_velocity_mrad_s)
Calculate new torque reference from system states.

2.4.1 Detailed Description

WABL Linear Quadratic Regulator

Author

Stephen Papierski stephenpapierski@gmail.com

Date

2015-04-28 14:23:33 2015-05-19 20:45:58

2.4.2 Function Documentation

2.4.2.1 float lqr (float *positionRef_mm*, float *position_mm*, float *velocity_mm_s*, float *angle_mrad*, float *a_velocity_mrad_s*)

Calculate new torque reference from system states.

Parameters

<i>positionRef_mm</i>	Position to track to in millimeters
<i>position_mm</i>	System state: position (x) in millimeters
<i>velocity_mm_s</i>	System state: velocity (x_dot) in millimeters/second
<i>angle_mrad</i>	System state: pendulum angle (phi) in milliradians
<i>a_velocity_mrad-_s</i>	System state: pendulum angular velocity (phi_dot) in milliradians/second

Returns

New torque reference in N*m

2.5 main.c File Reference

```
#include <avr/io.h>
#include <util/delay.h>
#include <pololu/orangutan.h>
#include <stdint.h>
#include <string.h>
#include <stdio.h>
#include "io_defs.h"
#include "modules.h"
#include "test_code/test.h"
```

Macros

- #define **F_CPU** 2000000UL

Functions

- int **main** (void)
- **ISR** (TIMER1_COMPA_vect)

2.5.1 Detailed Description

WABL main loop

Author

Stephen Papierski stephenpapierski@gmail.com

Date

2015-03-22 20:08:47 2015-05-19 22:12:10

2.6 modules.h File Reference

```
#include "motor/motor.h"
#include "orientation/orient.h"
#include "uart/uart.h"
#include "lqr/lqr.h"
#include "encoder/encoder.h"
#include "safety/safety.h"
#include "sound/sound.h"
#include "communication/comm.h"
#include "../libs/i2cmaster/i2cmaster.h"
#include "../libs/avr-systimer/tmr.h"
#include "../libs/ringbuffer/ringbuffer.h"
```

2.6.1 Detailed Description

Author

Stephen Papierski stephenpapierski@gmail.com

Date

2015-03-23 01:09:46 2015-05-19 21:12:04

2.7 motor.h File Reference

Functions

- double [motor_pid](#) (double torqueRef, unsigned char motor)
Calculates new motor voltage.
- double [get_current_value](#) (double raw)
Translate the raw current reading to amps.

2.7.1 Detailed Description

PID motor controller module

Author

Andrew Krock

Date

2015-03-26 03:49:13 2015-05-19 20:54:20

2.7.2 Function Documentation

2.7.2.1 double [get_current_value](#) (double raw)

Translate the raw current reading to amps.

Parameters

<i>raw</i>	Raw current reading
------------	---------------------

Returns

Current in amps

2.7.2.2 double [motor_pid](#) (double *torqueRef*, unsigned char *motor*)

Calculates new motor voltage.

Parameters

<i>torqueRef</i>	Torque reference from lqr() in N*m
<i>motor</i>	MOTOR1 or MOTOR2

Returns

Motor value that corresponds to a duty cycle [-255 255]

2.8 orient.h File Reference

Macros

- #define **GYRO_ADDR** (0b1101010 << 1)
- #define **GYRO_LOW_ODR** 0x39
- #define **GYRO_CTRL_REG1** 0x20
- #define **GYRO_CTRL_REG4** 0x23
- #define **GYRO_X_L** 0x28
- #define **GYRO_X_H** 0x29
- #define **ACCL_ADDR** (0b0011110 << 1)

- #define **ACCL_CTRL_REG1** 0x20
- #define **ACCL_CTRL_REG2** 0x21
- #define **ACCL_CTRL_REG5** 0x24
- #define **ACCL_CTRL_REG6** 0x25
- #define **ACCL_CTRL_REG7** 0x26
- #define **ACCL_Y_L** 0x2A
- #define **ACCL_Y_H** 0x2B
- #define **ACCL_Z_L** 0x2C
- #define **ACCL_Z_H** 0x2D

Functions

- void [orient_init](#) (void)
- void [orient_calibrate](#) (void)
- void [orient_update](#) (uint8_t delta_t_ms)
Update orientation values.
- double [orient_get_phi_dot_raw](#) (void)
Get raw angular velocity.
- double [orient_get_phi_dot](#) (void)
Get angular velocity.
- double [orient_get_phi_raw](#) (void)
Get raw angle.
- double [orient_get_phi](#) (void)
Get angle.
- void [orient_write_reg](#) (uint8_t dev_addr, uint8_t reg_addr, uint8_t value)
Write value to register.
- int16_t [orient_read_reg16](#) (uint8_t dev_addr, uint8_t reg_addr_l)
Read and combine high and low register.

2.8.1 Detailed Description

Orientation module for use with the Pololu AltIMU-10 v4

Author

Stephen Papierski stephenpapierski@gmail.com

Date

2015-03-24 10:19:26 2015-05-19 20:57:03

2.8.2 Function Documentation

2.8.2.1 void orient_calibrate (void)

Calibrate phi to account for gyro drift

2.8.2.2 double orient_get_phi (void)

Get angle.

Returns

Angle (phi) in milliradians

2.8.2.3 double orient_get_phi_dot (void)

Get angular velocity.

Returns

Angular velocity (phi_dot) in milliradians/second

2.8.2.4 double orient_get_phi_dot_raw (void)

Get raw angular velocity.

Returns

Angular velocity in gyro raw units

2.8.2.5 double orient_get_phi_raw (void)

Get raw angle.

Returns

Angle (phi) in gyro raw units

2.8.2.6 void orient_init (void)

Initialize orientation sensor over the i2c bus

2.8.2.7 int16_t orient_read_reg16 (uint8_t dev_addr, uint8_t reg_addr_l)

Read and combine high and low register.

Parameters

dev_addr	I2C device address
reg_addr_l	Address of the the low register

Returns

Value of low register combined with high register

2.8.2.8 void orient_update (uint8_t delta_t_ms)

Update orientation values.

Parameters

delta_t_ms	Time since calling function last in milliseconds
------------	--

2.8.2.9 void orient_write_reg (uint8_t dev_addr, uint8_t reg_addr, uint8_t value)

Write value to register.

Parameters

dev_addr	I2C device address
reg_addr	Register address
value	Value to write to register

2.9 safety.h File Reference

```
#include <pololu/orangutan.h>
```

Macros

- #define **BATLEV** ADC6
- #define **CRITICAL_VOLTAGE_MV** 10000
Battery's critical voltage.

Functions

- uint8_t **safety_battery_critical** (void)
Checks if battery voltage is at its critical level.

2.9.1 Detailed Description

Author

Stephen Papierski stephenpapierski@gmail.com

Date

2015-05-10 17:01:17 2015-05-19 21:03:00

2.9.2 Function Documentation

2.9.2.1 uint8_t safety_battery_critical (void)

Checks if battery voltage is at its critical level.

Returns

True if battery level is critical, false if not

2.10 sound.h File Reference

Typedefs

- typedef enum melody_e **melody_e**

Enumerations

- enum **melody_e** { **BATTERY_CRITICAL**, **BATTERY_LOW** }

Functions

- void [sound_play_melody](#) (melody_e melody)
Play melody from program memory.

2.10.1 Detailed Description

Sound module

Author

Stephen Papierski stephenpapierski@gmail.com

Date

2015-05-10 19:45:59 2015-05-19 21:05:03

2.10.2 Function Documentation

2.10.2.1 void sound_play_melody (melody_e melody)

Play melody from program memory.

Parameters

<i>melody</i>	Melody in program memory
---------------	--------------------------

2.11 uart.h File Reference

```
#include <pololu/orangutan.h>
```

Macros

- #define **XBEE** UART0

Functions

- void [uart_init](#) (void)
Sets up baud rate for UART communication.

2.11.1 Detailed Description

Author

Stephen Papierski stephenpapierski@gmail.com

Date

2015-04-28 09:17:16 2015-05-19 21:06:30

Index

comm.h, 3	
encoder.h, 3	
encoder_get_x, 3	
encoder_get_x_dot, 4	
encoder_update, 4	
encoder_get_x	
encoder.h, 3	
encoder_get_x_dot	
encoder.h, 4	
encoder_update	
encoder.h, 4	
get_current_value	
motor.h, 6	
io_defs.h, 4	
lqr	
lqr.h, 4	
lqr.h, 4	
lqr, 4	
main.c, 5	
modules.h, 5	
motor.h, 6	
get_current_value, 6	
motor_pid, 6	
motor_pid	
motor.h, 6	
orient.h, 6	
orient_calibrate, 7	
orient_get_phi, 7	
orient_get_phi_dot, 7	
orient_get_phi_dot_raw, 7	
orient_get_phi_raw, 8	
orient_init, 8	
orient_read_reg16, 8	
orient_update, 8	
orient_write_reg, 8	
orient_calibrate	
orient.h, 7	
orient_get_phi	
orient.h, 7	
orient_get_phi_dot	
orient.h, 7	
orient_get_phi_dot_raw	
orient.h, 7	
orient_get_phi_raw	
orient.h, 8	
orient_init	
orient.h, 8	
orient_read_reg16	
orient.h, 8	
orient_update	
orient.h, 8	
orient_write_reg	
orient.h, 8	
safety.h, 8	
safety_battery_critical, 9	
safety_battery_critical	
safety.h, 9	
sound.h, 9	
sound_play_melody, 9	
sound_play_melody	
sound.h, 9	
uart.h, 9	