



RT-Motion USB - User API Reference Manual

v2.00

Sait Izmit

Philips Applied Technologies

Contents

1	RT-Motion USB - User API Main Page	1
1.1	Introduction	1
1.2	Functional Overview	1
1.3	USB Communication	2
1.4	Firmware Error	2
1.5	I/O List	2
1.6	Analog I/O	3
1.7	PC Software	3
2	RT-Motion USB - User API File Index	5
2.1	RT-Motion USB - User API File List	5
3	RT-Motion USB - User API File Documentation	7
3.1	rtmotion_usb_conf.h File Reference	8

Chapter 1

RT-Motion USB - User API Main Page

1.1 Introduction

RT-Motion USB is a 2-axes motion control board with integrated drive units and extensive I/O. It is especially designed and developed to be used in high-speed USB-based distributed motion control applications. The electronics are supported by real-time software architecture, which is implemented both on firmware and PC-side driver levels.

RT-Motion USB motion control boards contain a 32-bit processor. An onboard CPLD is also included in the design for interfacing with incremental motor encoders. Two 150 watt DC motor amplifiers are used to drive two brushed DC motors. Five 10-bit and two 16-bit analog inputs together with 16 bit bidirectional digital I/O pins are also present on the RT-Motion USB motion control boards.

1.2 Functional Overview

The firmware can execute a single hard real-time task. The user should be careful when choosing the execution frequency of the hard real-time task. Enough processing bandwidth should be left also for other software modules.

USB communication is handled by the communication layer, which will be explained in more detail in the next section. There is also a non-real-time task scheduler present in the firmware. Tasks such as thermal and slip watchdogs can be scheduled using this scheduler. There is also a error management module, which will be explained in more detail in the coming sections.

RT-Motion USB board contains also three LEDs. Even though the user can switch these LEDs from the user API, their main indications are as follows:

1. Green LED: On after initialization, blinking whenever non-real-time scheduler is running.
2. Orange LED: Off after initialization, indicates how much processing power the hard real-time task is using. It should never be fully on. There should be enough processor bandwidth for other software modules.
3. Red LED: Off after initialization, indicates firmware error state.

1.3 USB Communication

RT-Motion USB communicates with the central PC through two communication channels over USB. These channels are:

1. Configuration Channel: Handles configuration related messages to serve user configuration requests.
2. Real-Time Data Channel: Handles real-time data communication. Transfers input data of RT-Motion USB to the user while outputs user defined data using board outputs.

The communication layer is executed with a lower priority than hard real-time task. Therefore enough processor bandwidth should always be made available by the hard real-time task for USB communication.

Over USB, all transactions are initiated by the PC. Real-time data write (PC to Firmware) transactions also trigger the firmware to fill RT-Motion USB outgoing buffer for the next read (Firmware to PC) transaction. Therefore, it is advised to execute the control loop with one sample delayed actuation signal. Message indexing is possible for real-time data verification. The user can increment the index before each write transaction. When a read transaction is performed, the firmware will also return the last received index. By comparing the two, user can verify correct message is being sent and received.

1.4 Firmware Error

There is an error management module present in the RT-Motion USB firmware. There are four errors states:

1. No Error State: Firmware is fully functional.
2. Non-Fatal Error State: Error Code logged and Red LED is turned-on.
3. Halt Error State: Error Code Logged and system is halted (all outputs are disabled). Red LED blinking task is set active.
4. Fatal Error State: Firmware is terminated.

Using the user API, user can always read back from the firmware, the error count, the error state of the firmware and the last 4 error codes that are logged. User can also reset these values via the user API. Whenever Halt Error occurs, error needs to be cleared for output access. Safety critical tasks such as thermal and slip coupling watchdog tasks will generate halt errors.

1.5 I/O List

RT-Motion USB motion controller boards contain the following I/O:

1. 16 bit digital bi-directional I/O (3.3V range, 5V input tolerant)
2. 1x10-bit 5ch analog inputs (0-3V), 2x16-bit analog inputs (0-3.3V)
3. 2x16-bit analog outputs (0-2.7V) (used to drive amplifiers)
4. 2 brushed DC motor amplifier outputs (each 150 watt max.)
5. 2 differential encoder input
6. 3 LEDs

1.6 Analog I/O

Analog inputs 0 to 4 correspond to the 10-bit analog converter channels while analog inputs 5 and 6 correspond to the 16-bit analog converter inputs. The range for the 10-bit analog inputs is in between 0 and 3 volts while that range is in between 0 and 3.3 volts for the 16-bit analog inputs. All the inputs are connected through an anti-aliasing filter to the converters. These anti-aliasing filters are implemented as 1st order low-pass filters with a corner frequency of 500 Hz.

The analog output can generate a voltage value between 0 and 2.7 volts. The output behavior of the D/A converter is pretty linear with a small non-linear region close to zero. When the amplifiers are disabled, the firmware accepts 16-bit unsigned short integer (range 0 to 65535) as input for the D/A outputs while it accepts 16-bit signed short integer (range -32768 to 32767) when amplifiers are enabled. The sign information is used to set the direction of the amplifiers.

1.7 PC Software

The PC communicates with RT-Motion USB motion controller board(s) via USB using the standard Linux USB driver stack. RT-Motion USB Linux device driver is implemented as a native Linux driver running in kernel space. Motion control user applications, on the other hand, should be implemented as real-time Xenomai task(s) in the user space.

Chapter 2

RT-Motion USB - User API File Index

2.1 RT-Motion USB - User API File List

Here is a list of all files with brief descriptions:

[rtmotion_usb_conf.h](#) (RT-Motion USB - User API Header File) 8

Chapter 3

RT-Motion USB - User API File Documentation

3.1 rtmotion_usb_conf.h File Reference

RT-Motion USB - User API Header File.

```
#include <unistd.h>
#include <stdio.h>
#include <asm/types.h>
```

Defines

- #define [ADC_NUM](#) 5
- #define [DADC_NUM](#) 2
- #define [DDAC_NUM](#) 2
- #define [ERR_UNSUPPORTED](#) -2
- #define [ERR_INVALID_BOARD_VERSION](#) -5
- #define [ERR_INVALID_AMPCH](#) -10
- #define [ERR_INVALID_PWMSOURCE](#) -15
- #define [ERR_INVALID_PID_REF](#) -20
- #define [ERR_INVALID_PID_SENS](#) -21
- #define [ERR_INVALID_PID_ACT](#) -22
- #define [ERR_MAX_NONRT_TASKS](#) -40
- #define [ERR_SCHED_ACTIVE](#) -41
- #define [ERR_SCHED_INVALIDTICK](#) -42
- #define [ERR_MOTOR_THERMAL](#) -60
- #define [ERR_MOTOR_SLIP](#) -70
- #define [ENCODER_NUM](#) 2
- #define [ENC_CH_X](#) 0
- #define [ENC_CH_Y](#) 1
- #define [PID_CNT_MAX](#) 10
- #define [PID_ANALOG_REF](#) 0
- #define [PID_ENCX_SENS](#) 0
- #define [PID_ENCY_SENS](#) 1
- #define [PID_ONBOARDAMPX_ACT](#) 0
- #define [PID_ONBOARDAMPY_ACT](#) 1
- #define [MAX_ONBOARD_AMP_NUM](#) 2
- #define [ONBOARD_AMP_X](#) 0
- #define [ONBOARD_AMP_Y](#) 1
- #define [AMP_SLOWDECAY](#) 0
- #define [AMP_MIX15DECAY](#) 1
- #define [AMP_MIX48DECAY](#) 2
- #define [AMP_FASTDECAY](#) 3
- #define [AMPENABLEARM](#) 0
- #define [AMPENABLECPLD](#) 1
- #define [SWPWMSOURCE](#) 0
- #define [HWPWMSOURCE](#) 1
- #define [NOPWMSOURCE](#) 2
- #define [MAX_NONRT_TASK](#) 20
- #define [SCHED_TICK](#) 1
- #define [MAX_CURRMS_SEC](#) 20

- #define [MAX_ERR_STORED](#) 4
- #define [ERROR_STATE_NOERR](#) 0
- #define [ERROR_STATE_ERR](#) 1
- #define [ERROR_STATE_HALTERR](#) 2
- #define [ERROR_STATE_FATALERR](#) 3

Functions

- int [rtm_usb_init](#) ()
- int [rtm_usb_terminate](#) ()
- int [rtm_usb_EncEN](#) (unsigned char dev_nr, unsigned char ch, unsigned char chA, unsigned char chB, unsigned char index)
- unsigned char [rtm_usb_EncDis](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_Enc_SetFreq](#) (unsigned char dev_nr, unsigned int value)
- unsigned char [rtm_usb_EncAlg](#) (unsigned char dev_nr, unsigned char alg)
- int [rtm_usb_ADC_EN](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_ADC_Dis](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_DAC_EN](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_DAC_Dis](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_DigIO_SetMask](#) (unsigned char dev_nr, unsigned int mask)
- int [rtm_usb_DigIO_GetMask](#) (unsigned char dev_nr, unsigned int *mask)
- int [rtm_usb_Amp_EN](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_Amp_Dis](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_Amp_DecMode](#) (unsigned char dev_nr, unsigned char ch, unsigned char mode)
- int [rtm_usb_Amp_Sleep](#) (unsigned char dev_nr, unsigned char ch, unsigned char on_off)
- int [rtm_usb_Amp_Blank](#) (unsigned char dev_nr, unsigned char ch, unsigned char on_off)
- int [rtm_usb_Amp_ExtMode](#) (unsigned char dev_nr, unsigned char ch, unsigned char on_off)
- int [rtm_usb_Amp_SetEnableSigPath](#) (unsigned char dev_nr, unsigned char ch, unsigned char path)
- int [rtm_usb_Amp_SetPwmThreshold](#) (unsigned char dev_nr, unsigned char ch, unsigned short int value)
- int [rtm_usb_Amp_SetPwmThresholdGain](#) (unsigned char dev_nr, unsigned char ch, unsigned short int Gmult, unsigned short int Gdiv)
- int [rtm_usb_Amp_SetPwmSrc](#) (unsigned char dev_nr, unsigned char ch, unsigned char source)
- int [rtm_usb_Amp_EnablePwm](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_Amp_DisablePwm](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_Amp_SetSwPwmFreq](#) (unsigned char dev_nr, unsigned char ch, unsigned char freq)
- int [rtm_usb_Amp_SetSwPwmSigSrc](#) (unsigned char dev_nr, unsigned char ch, unsigned char source)
- int [rtm_usb_LED0](#) (unsigned char dev_nr, signed char status)
- int [rtm_usb_LED1](#) (unsigned char dev_nr, signed char status)
- int [rtm_usb_LED2](#) (unsigned char dev_nr, signed char status)
- int [rtm_usb_Version](#) (unsigned char dev_nr, unsigned char *version_str)
- int [rtm_usb_BoardVersion](#) (unsigned char dev_nr, unsigned char *BoardVersion)
- int [rtm_usb_CpldImageVersion](#) (unsigned char dev_nr, unsigned char *CpldImageVersion)
- int [rtm_usb_get_dev_nr](#) (int busnum, const char *devpath)
- int [rtm_usb_Dsend](#) (unsigned char dev_nr, void *data, unsigned int cnt)
- *int [rtm_usb_Dread](#) (unsigned char dev_nr, unsigned int cnt)
- int [rtm_usb_Dread_get](#) (unsigned char dev_nr, void *data, unsigned int cnt)
- unsigned char [rtm_usb_Dsend_ready](#) (unsigned char dev_nr)
- unsigned char [rtm_usb_Dread_ready](#) (unsigned char dev_nr)

- int [rtm_usb_Read_as5046](#) (unsigned char dev_nr, unsigned int *ret_data, unsigned char ch)
- int [rtm_usb_HCTL2032_En](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_HCTL2032_Dis](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_HCTL2032_Reset](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_HCTL2032_SetRes](#) (unsigned char dev_nr, unsigned char res)
- int [rtm_usb_RtmCount_SetSource](#) (unsigned char dev_nr, unsigned char source, unsigned char port)
- int [rtm_usb_RtmCount_EnCh](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_RtmCount_DisCh](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_RtmCount_ResetCh](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_RtmCount_SetRes](#) (unsigned char dev_nr, unsigned char res)
- int [rtm_usb_RtmCount_EnCycRead](#) (unsigned char dev_nr, unsigned char RtMode, unsigned int T_usec)
- int [rtm_usb_RtmCount_DisCycRead](#) (unsigned char dev_nr)
- int [rtm_usb_RtmCount_Read](#) (unsigned char dev_nr, unsigned char ch, unsigned int *EncValue)
- int [rtm_usb_Pid_En](#) (unsigned char dev_nr, unsigned int freq)
- int [rtm_usb_Pid_Dis](#) (unsigned char dev_nr)
- int [rtm_usb_Pid_RegCont](#) (unsigned char dev_nr, unsigned short int SensIndex, unsigned short int ActIndex, unsigned short int RefIndex, float Kp, float Td, float Ti, float Ff, float Nd)
- int [rtm_usb_Pid_EncConst](#) (unsigned char dev_nr, float ConstEncX, float ConstEncY)
- int [rtm_usb_Pid_DeleteAll](#) (unsigned char dev_nr)
- int [rtm_usb_Pid_NoActive](#) (unsigned char dev_nr, unsigned char *ContNr)
- int [rtm_usb_dac7624_SetPort](#) (unsigned char dev_nr, unsigned char ch)
- int [rtm_usb_Sched_Enable](#) (unsigned char dev_nr, unsigned char status)
- int [rtm_usb_Sched_ClrList](#) (unsigned char dev_nr)
- int [rtm_usb_Sched_SetTick](#) (unsigned char dev_nr, unsigned int tick)
- int [rtm_usb_Thermal_SchedNonRt](#) (unsigned char dev_nr, unsigned int period)
- int [rtm_usb_Thermal_Config](#) (unsigned char dev_nr, unsigned char ch, unsigned char Tsec, unsigned short int CurRmsLimit)
- int [rtm_usb_Err_Read](#) (unsigned char dev_nr, unsigned char *ErrState, signed short int *ErrBuffer)
- int [rtm_usb_Err_Clr](#) (unsigned char dev_nr)
- int [rtm_usb_Slip_SchedNonRt](#) (unsigned char dev_nr, unsigned int period)
- int [rtm_usb_Slip_Config](#) (unsigned char dev_nr, unsigned char ch, unsigned char src, signed short int AbsEncNum, signed short int AbsEncDen, signed short int EncNum, signed short int EncDen, unsigned short int limit)

3.1.1 Detailed Description

RT-Motion USB - User API Header File.

Author:

Sait Izmit

Copyright (c) 2008 Koninklijke Philips Electronics N.V. All rights reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

In particular, under no circumstances is this software to be combined with any Open Source Software in any way or placed under an Open Source License of any type without the express written permission of Koninklijke Philips Electronics N.V.

This source code and any compilation or derivative thereof is the proprietary information of Koninklijke Philips Electronics N.V. and is confidential in nature.

This source code is developed and maintained by: Philips Applied Technologies.

2008.11.26 v1.18 GH changed sys/types.h back to asm/types.h since __s32 is not defined by sys/types.h

3.1.2 Define Documentation

3.1.2.1 #define ADC_NUM 5

Number of 10-Bit ADC Channels.

3.1.2.2 #define AMP_FASTDECAY 3

Fast Decay Mode.

3.1.2.3 #define AMP_MIX15DECAY 1

Mixed Decay Mode (15).

3.1.2.4 #define AMP_MIX48DECAY 2

Mixed Decay Mode (48).

3.1.2.5 #define AMP_SLOWDECAY 0

Slow Decay Mode.

3.1.2.6 #define AMPENABLEARM 0

Enable signal to amplifier is directly from the processor.

3.1.2.7 #define AMPENABLECPLD 1

Enable signal to amplifier is from the CPLD PWM unit.

3.1.2.8 #define DADC_NUM 2

Number of 16-Bit ADC Channels.

3.1.2.9 #define DDAC_NUM 2

Number of 16-Bit DAC Channels.

3.1.2.10 #define ENC_CH_X 0

Index for Encoder Channel X.

3.1.2.11 #define ENC_CH_Y 1

Index for Encoder Channel Y.

3.1.2.12 #define ENCODER_NUM 2

Number of Encoder Channels.

3.1.2.13 #define ERR_INVALID_AMPCH -10

Invalid Onboard Amplifier Channel.

3.1.2.14 #define ERR_INVALID_BOARD_VERSION -5

Invalid RT-Motion USB board version.

3.1.2.15 #define ERR_INVALID_PID_ACT -22

Invalid PID Actuation Function.

3.1.2.16 #define ERR_INVALID_PID_REF -20

Invalid PID Reference Function.

3.1.2.17 #define ERR_INVALID_PID_SENS -21

Invalid PID Sense Function.

3.1.2.18 #define ERR_INVALID_PWMSOURCE -15

Invalid Amplifier PWM Source.

3.1.2.19 #define ERR_MAX_NONRT_TASKS -40

Maximum number of non-real-time tasks exceeded.

3.1.2.20 #define ERR_MOTOR_SLIP -70

Motor Slip Coupling Limit Reached.

3.1.2.21 #define ERR_MOTOR_THERMAL -60

Motor Thermal Time Constant Limit Reached.

3.1.2.22 #define ERR_SCHED_ACTIVE -41

Scheduler is active, cannot perform the operation.

3.1.2.23 #define ERR_SCHED_INVALIDTICK -42

Invalid scheduler tick.

3.1.2.24 #define ERR_UNSUPPORTED -2

Not Supported for the Current Version.

3.1.2.25 #define ERROR_STATE_ERR 1

Error state.

3.1.2.26 #define ERROR_STATE_FATALERR 3

Fatal error state.

3.1.2.27 #define ERROR_STATE_HALTERR 2

Halt error state.

3.1.2.28 #define ERROR_STATE_NOERR 0

No error state.

3.1.2.29 #define HWPWMSOURCE 1

HW-based PWM Source.

3.1.2.30 #define MAX_CURRMS_SEC 20

Maximum window size in seconds for current RMS calculation.

3.1.2.31 #define MAX_ERR_STORED 4

The number of error codes stored.

3.1.2.32 #define MAX_NONRT_TASK 20

Maximum number of non-RT tasks that can be scheduled.

3.1.2.33 #define MAX_ONBOARD_AMP_NUM 2

Maximum number of onboard amplifiers available.

3.1.2.34 #define NOPWMSOURCE 2

No Amplifier PWMing.

3.1.2.35 #define ONBOARD_AMP_X 0

Index for onboard amplifier 0.

3.1.2.36 #define ONBOARD_AMP_Y 1

Index for onboard amplifier 1.

3.1.2.37 #define PID_ANALOG_REF 0

Index for Analog Reference Function.

3.1.2.38 #define PID_CNT_MAX 10

Maximum number of active PID controllers.

3.1.2.39 #define PID_ENC_X_SENS 0

Index for Encoder X Sense Function.

3.1.2.40 #define PID_ENC_Y_SENS 1

Index for Encoder Y Sense Function.

3.1.2.41 #define PID_ONBOARDAMPX_ACT 0

Index for Onboard Amplifier X Actuation Function.

3.1.2.42 #define PID_ONBOARDAMPY_ACT 1

Index for Onboard Amplifier Y Actuation Function.

3.1.2.43 #define SCHED_TICK 1

Schedulat Tick is set to 1 ms by default.

3.1.2.44 #define SWPWMSOURCE 0

SW-based PWM Source.

3.1.3 Function Documentation

3.1.3.1 int rtm_usb_ADC_Dis (unsigned char *dev_nr*, unsigned char *ch*)

Disable D/A Converter.

Parameters:

dev_nr RT-Motion USB device number

ch RT-Motion USB A/D Channel

Channels 0-4: 10-bit ADC

Channels 5-6: 16-bit ADC

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.2 int rtm_usb_ADC_EN (unsigned char *dev_nr*, unsigned char *ch*)

Enables A/D Conversion.

Parameters:

dev_nr RT-Motion USB device number

ch RT-Motion USB A/D Channel

Channels 0-4: 10-bit ADC

Channels 5-6: 16-bit ADC

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.3 int rtm_usb_Amp_Blank (unsigned char *dev_nr*, unsigned char *ch*, unsigned char *on_off*)

Set Amplifier Blank Pin (1: ON, 0: OFF).

Parameters:

dev_nr RT-Motion USB device number

ch RT-Motion USB Amplifier Channel 0 or 1

on_off RT-Motion USB Amplifier Blank Pin 0 or 1

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.4 int rtm_usb_Amp_DecMode (unsigned char *dev_nr*, unsigned char *ch*, unsigned char *mode*)

Choose Amplifier Decay Mode (Channels 0-1).

Parameters:

dev_nr RT-Motion USB device number

ch RT-Motion USB Amplifier Channel 0 or 1

mode RT-Motion USB Amplifier Decay Mode

Mode 0: Slow Decay Mode

Mode 1: Mixed Decay Mode (15)

Mode 2: Mixed Decay Mode (48)

Mode 3: Fast Decay Mode

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.5 int rtm_usb_Amp_Dis (unsigned char *dev_nr*, unsigned char *ch*)

Disable Amplifier (Channels 0-1).

Parameters:

dev_nr RT-Motion USB device number

ch RT-Motion USB Amplifier Channel 0 or 1

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.6 int rtm_usb_Amp_DisablePwm (unsigned char *dev_nr*, unsigned char *ch*)

Disable Amplifier PWM.

This function is used for amplifier linearization

Parameters:

dev_nr Device Number

ch Amplifier Channel 0 or 1

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.7 int rtm_usb_Amp_EN (unsigned char *dev_nr*, unsigned char *ch*)

Enables Amplifier (Channels 0-1).

Parameters:

dev_nr RT-Motion USB device number

ch RT-Motion USB Amplifier Channel 0 or 1

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.8 int rtm_usb_Amp_EnablePwm (unsigned char *dev_nr*, unsigned char *ch*)

Enable Amplifier PWM.

This function is used for amplifier linearization

Parameters:

dev_nr Device Number

ch Amplifier Channel 0 or 1

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.9 int rtm_usb_Amp_ExtMode (unsigned char *dev_nr*, unsigned char *ch*, unsigned char *on_off*)

Set Amplifier ExtMode Pin (1: ON, 0: OFF).

Parameters:

dev_nr RT-Motion USB device number

ch RT-Motion USB Amplifier Channel 0 or 1

on_off RT-Motion USB Amplifier ExtMode Pin 0 or 1

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.10 int rtm_usb_Amp_SetEnableSigPath (unsigned char *dev_nr*, unsigned char *ch*, unsigned char *path*)

Set Amplifier Enable Pin Physical Signal Path.

Parameters:

dev_nr Device Number

ch Amplifier Channel 0 or 1

path PWM signal source

0: Directly Connected to the Processor

1: Connected via PWM unit

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.11 int rtm_usb_Amp_SetPwmSrc (unsigned char *dev_nr*, unsigned char *ch*, unsigned char *source*)

Set Amplifier PWM Source.

This function is used for amplifier linearization

Parameters:

dev_nr Device Number

ch Amplifier Channel 0 or 1

source PWM source

0: SW PWM

1: HW PWM

2: No PWM

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.12 int rtm_usb_Amp_SetPwmThreshold (unsigned char *dev_nr*, unsigned char *ch*, unsigned short int *value*)

Set Amplifier PWM Treshhold Value.

This function is used for amplifier linearization

Parameters:

dev_nr Device Number

ch Amplifier Channel 0 or 1

value PWM Trashhold Value (max 0x7FFF)

Return values:

<0> for Success
<Negative Values> for Errors

3.1.3.13 int rtm_usb_Amp_SetPwmThresholdGain (unsigned char *dev_nr*, unsigned char *ch*, unsigned short int *Gmult*, unsigned short int *Gdiv*)

Set Gain for Amplifier PWM Treshold Region.

This function is used for amplifier linearization

Parameters:

dev_nr Device Number
ch Amplifier Channel 0 or 1
Gmult Amplifier Reference Multiplier
Gdiv Amplifier Reference Divider

Return values:

<0> for Success
<Negative Values> for Errors

3.1.3.14 int rtm_usb_Amp_SetSwPwmFreq (unsigned char *dev_nr*, unsigned char *ch*, unsigned char *freq*)

Set Amplifier SW-based PWM Frequency.

This function is used for amplifier linearization

Parameters:

dev_nr Device Number
ch Amplifier Channel 0 or 1
freq PWM frequency reference
(0: 5/10kHz, 1:10/20kHz, 2: 20/40kHz (Phase/Enable))

Return values:

<0> for Success
<Negative Values> for Errors

3.1.3.15 int rtm_usb_Amp_SetSwPwmSigSrc (unsigned char *dev_nr*, unsigned char *ch*, unsigned char *source*)

Set Amplifier SW-based PWM Source.

This function is used for amplifier linearization

Parameters:

dev_nr Device Number
ch Amplifier Channel 0 or 1
source PWM source
 (0: phase, 1: enable)

Return values:

<0> for Success
 <Negative Values> for Errors

3.1.3.16 int rtm_usb_Amp_Sleep (unsigned char *dev_nr*, unsigned char *ch*, unsigned char *on_off*)

Choose Amplifier Sleep Mode (1: ON, 0: OFF).

Parameters:

dev_nr RT-Motion USB device number
ch RT-Motion USB Amplifier Channel 0 or 1
on_off RT-Motion USB Amplifier Sleep Mode
 0: Wake up
 1: Sleep

Return values:

<0> for Success
 <Negative Values> for Errors

3.1.3.17 int rtm_usb_BoardVersion (unsigned char *dev_nr*, unsigned char * *BoardVersion*)

Request RT-Motion USB Board Version.

Parameters:

dev_nr RT-Motion USB device number
BoardVersion Pointer to board version

Return values:

<0> for Success
 <Negative Values> for Errors

3.1.3.18 int rtm_usb_CpldImageVersion (unsigned char *dev_nr*, unsigned char * *CpldImageVersion*)

Request RT-Motion USB CPLD Image Version.

Parameters:

dev_nr RT-Motion USB device number

CpldImageVersion Pointer to CPLD Image Version

Return values:

<0> for Success
<Negative Values> for Errors

3.1.3.19 int rtm_usb_dac7624_SetPort (unsigned char *dev_nr*, unsigned char *ch*)

Set Digital I/O Port for DAC7624 (Ports 0-1).

Parameters:

dev_nr RT-Motion USB device number
ch Digital I/O Port 0 or 1
0: 0-15 bits (Port 0)
1: 16-31 bits(Port 1) - Doesn't exist on board version 2

Return values:

<0> for Success
<Negative Values> for Errors

3.1.3.20 int rtm_usb_DAC_Dis (unsigned char *dev_nr*, unsigned char *ch*)

Disables D/A Converter (Channels 0-1).

Parameters:

dev_nr RT-Motion USB device number
ch RT-Motion USB D/A Channel

Return values:

<0> for Success
<Negative Values> for Errors

3.1.3.21 int rtm_usb_DAC_EN (unsigned char *dev_nr*, unsigned char *ch*)

Enables D/A Converter (Channels 0-1).

Parameters:

dev_nr RT-Motion USB device number
ch RT-Motion USB D/A Channel

Return values:

<0> for Success
<Negative Values> for Errors

3.1.3.22 int rtm_usb_DigIO_GetMask (unsigned char *dev_nr*, unsigned int * *mask*)

Get RT-Motion USB Digital I/O Mask.

Parameters:

dev_nr RT-Motion USB device number
mask Pointer to Digital I/O Mask
0 is input, 1 is output

Return values:

<0> for Success
<Negative Values> for Errors

3.1.3.23 int rtm_usb_DigIO_SetMask (unsigned char *dev_nr*, unsigned int *mask*)

Set RT-Motion USB Digital I/O Mask.

Parameters:

dev_nr RT-Motion USB device number
mask RT-Motion USB Digital I/O Mask
0 is input, 1 is output

Return values:

<0> for Success
<Negative Values> for Errors

3.1.3.24 *int rtm_usb_Dread (unsigned char *dev_nr*, unsigned int *cnt*)

Request Read Data through Real-Time Communication Channel.

Parameters:

dev_nr RT-Motion USB device number
cnt USB data size package in number of bytes

Return values:

<0> for Success
<Negative Values> for Errors

3.1.3.25 int rtm_usb_Dread_get (unsigned char *dev_nr*, void * *data*, unsigned int *cnt*)

Get Requested Read Data through Real-Time Communication Channel.

Parameters:

dev_nr RT-Motion USB device number

data Buffer pointer for incoming USB data package

cnt USB data size package in number of bytes
36 for Message Type 0, 17 for Message Type 2

Return values:

<0> for Success

<Negative Values> for Errors

The received data have a special format. The first 2 bytes contain the standard header:

- Byte 0 - U8 Last Received Message Index
- Byte 1 - U8 Last Received Message Type

For Message Type 0 (Standard Message):

- Byte 2-3 - U16 10-bit A/D Input for Channel 0
- Byte 4-5 - U16 10-bit A/D Input for Channel 1
- Byte 6-7 - U16 10-bit A/D Input for Channel 2
- Byte 8-9 - U16 10-bit A/D Input for Channel 3
- Byte 10-11 - U16 10-bit A/D Input for Channel 4
- Byte 12-13 - U16 16-bit A/D Input for Channel 0
- Byte 14-15 - U16 16-bit A/D Input for Channel 1
- Byte 16-19 - U32 Encoder Count for Channel X (0)
- Byte 20-23 - U32 Encoder Count for Channel Y (1)
- Byte 24-27 - U32 Digital I/O
- Byte 28-29 - U16 Encoder 1/T Count for Channel X (0)
- Byte 30-31 - U16 Encoder 1/T Count for Channel Y (1)
- Byte 32-35 - U32 Last Received PC Time Stamp

For Message Type 1 (Robot Eyes):

- Undocumented

For Message Type 2 (PID Controller Message):

- Byte 2-3 - Empty
- Byte 4-7 - U32 Lastest PID Controller Set-Point
- Byte 8-11 - U32 Lastest PID Controller Sencor Input
- Byte 12-15 - U32 Lastest PID Controller Output
- Byte 16 - U8 Lastest PID Controller Channel

3.1.3.26 unsigned char rtm_usb_Dread_ready (unsigned char *dev_nr*)

Is Data Read through Real-Time Communication Channel Completed (Callback Returned?).

Parameters:

dev_nr RT-Motion USB device number

Return values:

<0> for Success

<Non-Zero Values> for Errors

3.1.3.27 int rtm_usb_Dsend (unsigned char *dev_nr*, void * *data*, unsigned int *cnt*)

Send Data through Real-Time Communication Channel.

Parameters:

dev_nr RT-Motion USB device number

data Pointer to USB data package

cnt USB data size package in number of bytes
16 for Message Type 0, 17 for Message Type 2

Return values:

<0> for Success

<Negative Values> for Errors

The data should have a special format. The first 8 bytes are the standard header:

- Byte 0 - U8 Message Index (Optional)
- Byte 1 - U8 Message Type
- Byte 2-3 - Empty
- Byte 4-7 - U32 PC Time Stamp (Optional)

For Message Type 0 (Standard Message):

- Byte 8-9 - U16 D/A Output for Channel X (0)
- Byte 10-11 - U16 D/A Output for Channel Y (1)
- Byte 12-15 - U32 Digital I/O (bits for inputs will be disregarded in fw)

For Message Type 1 (Robot Eyes):

- Undocumented

For Message Type 2 (PID Controller Message):

- Byte 8-11 - PID Controller Set-Point
- Byte 12-15 - Feed-Forward Gain for PID Controller Set-Point
- Byte 16 - U8 Controller Channel

3.1.3.28 unsigned char rtm_usb_Dsend_ready (unsigned char *dev_nr*)

Is Data Send through Real-Time Communication Channel Completed (Callback Returned?).

Parameters:

dev_nr RT-Motion USB device number

Return values:

<0> for Success

<Non-Zero Values> for Errors

3.1.3.29 int rtm_usb_Enc_SetFreq (unsigned char *dev_nr*, unsigned int *value*)

Sets Encoder Counting Frequency(Max. 100kHz).

Parameters:

ch Channels 0-1

value Frequency Value in Hz

Return values:

<0> for Success

<Negative Values> for Errors

Note: When the two encoder channels are enabled with a encoder counting frequency of 100kHz, the software encoder counting algorithm is using about 75% percent of the processing power. For one channel, the algorithm is using about 46% of the procesing power. Therefore, the user is advised to use a maximum encoder counting frequency of less than 100kHz when both channels are enabled and less than 200kHz when only one channel is enabled.

3.1.3.30 unsigned char rtm_usb_EncAlg (unsigned char *dev_nr*, unsigned char *alg*)

Choose SW Encoder Counting Algorithm.

Parameters:

dev_nr Device Number

alg Encoder Algorithm Type

0 - Standard

1 - Estimation

2 - 1/T time between edges

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.31 unsigned char rtm_usb_EncDis (unsigned char *dev_nr*, unsigned char *ch*)

Disables Software Encoder Counting.

Parameters:

dev_nr RT-Motion USB Device Number

ch Encoder Channel 0 or 1

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.32 int rtm_usb_EncEN (unsigned char *dev_nr*, unsigned char *ch*, unsigned char *chA*, unsigned char *chB*, unsigned char *index*)

Enables Software Encoder Counting.

Parameters:

dev_nr Device Number

ch Channels 0-1

chA Pin Number for Encoder Channel A Signal

chB Pin Number for Encoder Channel B Signal

index Pin Number for Encoder Index Signal

Return values:

<0> for Success

<Negative Values> for Errors

Note: chA, chB and index must be different than each other.

3.1.3.33 int rtm_usb_Err_Clr (unsigned char *dev_nr*)

Clears Firmware Error List & Resets Error Count.

Parameters:

dev_nr RT-Motion USB Device Number

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.34 int rtm_usb_Err_Read (unsigned char *dev_nr*, unsigned char * *ErrState*, signed short int * *ErrBuffer*)

Return Firmware Error Count and Error IDs.

Parameters:

dev_nr RT-Motion USB Device Number

ErrState Pointer to firmware Error State

ErrBuffer Pointer to current Firmware Error Codes

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.35 int rtm_usb_get_dev_nr (int *busnum*, const char * *devpath*)

Get RT-Motion USB Device Number using Path Name(Linux only).

Parameters:

bus_num Linuc USB bus number

devpath Pointer to device path string

Return values:

<Positive Values> for RT-Motion USB Device Number

<Negative Values> for Errors

3.1.3.36 int rtm_usb_HCTL2032_Dis (unsigned char *dev_nr*, unsigned char *ch*)

Disable HCTL-2032 Encoder Read @ 33 MHz (Channels 0-1).

Parameters:

dev_nr RT-Motion USB device number

ch Encoder Channel 0 or 1

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.37 int rtm_usb_HCTL2032_En (unsigned char *dev_nr*, unsigned char *ch*)

Enable HCTL-2032 Encoder Read @ 33 MHz (Channels 0-1).

Parameters:

dev_nr RT-Motion USB device number

ch Encoder Channel 0 or 1

Return values:

<0> for Success
 <Negative Values> for Errors

3.1.3.38 int rtm_usb_HCTL2032_Reset (unsigned char *dev_nr*, unsigned char *ch*)

Reset HCTL-2032 Encoder Channel (Channels 0-1).

Parameters:

dev_nr RT-Motion USB device number
ch Encoder Channel 0 or 1

Return values:

<0> for Success
 <Negative Values> for Errors

3.1.3.39 int rtm_usb_HCTL2032_SetRes (unsigned char *dev_nr*, unsigned char *res*)

Set HCTL-2032 Resolution.

Parameters:

dev_nr RT-Motion USB device number
res Encoder Count Resolution
 0: 1x Resolution
 1: 2x Resolution
 2: 4x Resolution

Return values:

<0> for Success
 <Negative Values> for Errors

3.1.3.40 int rtm_usb_init ()

Initializes Real-Time FIFOs.

3.1.3.41 int rtm_usb_LED0 (unsigned char *dev_nr*, signed char *status*)

Turn On/Off LED0 (1: ON, 0: OFF).

Parameters:

dev_nr RT-Motion USB device number

status LED Status 0-off or 1-on

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.42 int rtm_usb_LED1 (unsigned char *dev_nr*, signed char *status*)

Turn On/Off LED1 (1: ON, 0: OFF).

Parameters:

dev_nr RT-Motion USB device number

status LED Status 0-off or 1-on

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.43 int rtm_usb_LED2 (unsigned char *dev_nr*, signed char *status*)

Turn On/Off LED2 (1: ON, 0: OFF).

Parameters:

dev_nr RT-Motion USB device number

status LED Status 0-off or 1-on

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.44 int rtm_usb_Pid_DeleteAll (unsigned char *dev_nr*)

Deletes all PID controller instances.

Parameters:

dev_nr RT-Motion USB Device Number

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.45 int rtm_usb_Pid_Dis (unsigned char *dev_nr*)

Disable PID Controller Task.

Parameters:

dev_nr RT-Motion USB device number

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.46 int rtm_usb_Pid_En (unsigned char *dev_nr*, unsigned int *freq*)

Enables Hard Real-Time PID Controller Task.

Parameters:

dev_nr RT-Motion USB device number

Freq Task Frequency in nano-seconds

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.47 int rtm_usb_Pid_EncConst (unsigned char *dev_nr*, float *ConstEncX*, float *ConstEncY*)

Assigns Encoder Constants for PID Controller Template.

Parameters:

dev_nr RT-Motion USB Device Number

ConstEncX Encoder constant for channel X

ConstEncY Encoder constant for channel Y

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.48 int rtm_usb_Pid_NoActive (unsigned char *dev_nr*, unsigned char * *ContNr*)

Returns number of active controller instances.

Parameters:

dev_nr RT-Motion USB device number

ContNr Buffer pointer for incoming return data

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.49 int rtm_usb_Pid_RegCont (unsigned char *dev_nr*, unsigned short int *SensIndex*, unsigned short int *ActIndex*, unsigned short int *RefIndex*, float *Kp*, float *Td*, float *Ti*, float *Ff*, float *Nd*)

Registers new PID Controller Instance

Parameters:

dev_nr RT-Motion USB Device Number
SensInd 0 - Encoder X, 1 - Encoder Y
ActInd 0 - Onboard Amplifier X, 1 - Onboard Amplifier Y
RefInd 0 - dummy
Kp Proportional Gain
Td Derivative Time Constant
Ti Integral Time Constant
Ff Feed-forward Gain
Nd Derivative Gain Limitation

Return values:

<0> for Success
 <Negative Values> for Errors

The PID controller template is implemented as follows:

- $u(kh) = Kp \cdot e(kh) + \text{PidInteg}(k \cdot h) + \text{PidDer}(k \cdot h) + Ff \cdot r(k \cdot h)$
 where u is controller output, which is saturated between -1 and 1, r is controller setpoint, x is sensor output and e is the error between set-point and sensor output.

The derivative and integral actions are calculated according to:

- $\text{PidDer}(k \cdot h) = Td / (Td + Nd \cdot h) \cdot \text{PidDer}(k \cdot h - h) - (Kp \cdot Td \cdot Nd) / (Td + Nd \cdot h) \cdot (x(k \cdot h) - x(k \cdot h - h))$
- $\text{PidInteg}(k \cdot h) = \text{PidInteg}(k \cdot h - h) + e(k \cdot h - h) \cdot (Kp \cdot h) / Ti$

3.1.3.50 int rtm_usb_Read_as5046 (unsigned char *dev_nr*, unsigned int * *ret_data*, unsigned char *ch*)

Read AS5046 (i2c Angle Sensor).

Parameters:

dev_nr RT-Motion USB device number
ret_data Buffer pointer for incoming return data
ch Channel

Return values:

<0> for Success
 <Negative Values> for Errors

3.1.3.51 int rtm_usb_RtmCount_DisCh (unsigned char *dev_nr*, unsigned char *ch*)

Disable RTM-Count Channel (Channels 0-1).

Parameters:

dev_nr RT-Motion USB device number

ch Encoder Channel 0 or 1

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.52 int rtm_usb_RtmCount_DisCycRead (unsigned char *dev_nr*)

Disable cyclic reading (real-time or non-real-time).

Parameters:

dev_nr RT-Motion USB device number

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.53 int rtm_usb_RtmCount_EnCh (unsigned char *dev_nr*, unsigned char *ch*)

Enable RTM-Count Channel (Channels 0-1).

Parameters:

dev_nr RT-Motion USB device number

ch Encoder Channel 0 or 1

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.54 int rtm_usb_RtmCount_EnCycRead (unsigned char *dev_nr*, unsigned char *RtMode*, unsigned int *T_usec*)

Enable cyclic reading (real-time or non-real-time).

Parameters:

dev_nr RT-Motion USB device number

RtMode Periodic Read Mode

0: non-real-time

1: real-time

T_usec Real-time sampling period (scheduler ticks for mode 0, usec for mode 1)

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.55 int rtm_usb_RtmCount_Read (unsigned char dev_nr, unsigned char ch, unsigned int * EncValue)

Sporadic encoder read (one time read).

Parameters:

dev_nr RT-Motion USB device number

ch Encoder Channel

EncValue Buffer pointer for incoming return data

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.56 int rtm_usb_RtmCount_ResetCh (unsigned char dev_nr, unsigned char ch)

Reset RTM-Count Channel (Channels 0-1).

Parameters:

dev_nr RT-Motion USB device number

ch Encoder Channel 0 or 1

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.57 int rtm_usb_RtmCount_SetRes (unsigned char dev_nr, unsigned char res)

Set RTM-Count Resolution.

Parameters:

dev_nr RT-Motion USB device number

res Encoder Count Resolution

0: 1x Resolution

1: 2x Resolution

2: 4x Resolution

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.58 int rtm_usb_RtmCount_SetSource (unsigned char *dev_nr*, unsigned char *source*, unsigned char *port*)

Select HW Encoder Counter Source.

Parameters:

dev_nr RT-Motion USB device number

res RTM-Count Source

0: External RTM-Count with Parallel Interface is Selected

1: External HCTL2032 with Parallel Interface is Selected

2: Onboard RTM-Count with Parallel Interface is Selected

port Digital I/O port (only port 0 for board version 2)

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.59 int rtm_usb_Sched_ClrList (unsigned char *dev_nr*)

Clears Non-RT Task Scheduler List.

Parameters:

dev_nr RT-Motion USB device number

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.60 int rtm_usb_Sched_Enable (unsigned char *dev_nr*, unsigned char *status*)

Starts or stops Non-RT Task Scheduler.

Parameters:

dev_nr RT-Motion USB device number

status 0 - Stop, 1- Start

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.61 int rtm_usb_Sched_SetTick (unsigned char *dev_nr*, unsigned int *tick*)

Sets the tick value for Non-RT Task Scheduler Timer.

Parameters:

dev_nr RT-Motion USB device number

tick Scheduler tick in number of milliseconds

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.62 int rtm_usb_Slip_Config (unsigned char *dev_nr*, unsigned char *ch*, unsigned char *src*, signed short int *AbsEncNum*, signed short int *AbsEncDen*, signed short int *EncNum*, signed short int *EncDen*, unsigned short int *limit*)

Configures Slip Coupling Watchdog.

The function should be called before rtm_usb_Slip_SchedNonRt.

Parameters:

dev_nr RT-Motion USB Device Number

ch motor channel

src source for absolute encoder (A/D channel)

AbsEncNum Absolute encoder constant numerator (not a floating point)

AbsEncDen Absolute encoder constant denominator (not a floating point)

EncNum Encoder constant numerator (not a floating point)

EncDen Encoder constant denominator (not a floating point)

limit Slip limit in degrees

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.63 int rtm_usb_Slip_SchedNonRt (unsigned char *dev_nr*, unsigned int *period*)

Schedules Slip Coupling Watchdog Task.

Parameters:

dev_nr RT-Motion USB Device Number

period Number of non-RT task scheduler timer ticks

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.64 int rtm_usb_terminate ()

Closes Real-Time FIFOs.

3.1.3.65 int rtm_usb_Thermal_Config (unsigned char *dev_nr*, unsigned char *ch*, unsigned char *Tsec*, unsigned short int *CurRmsLimit*)

Configures Thermal Watchdog for DC Motors Connected to Onboard Amplifiers.

The function should be called after rtm_usb_Thermal_SchedNonRt.

Parameters:

dev_nr RT-Motion USB Device Number

Tsec DC Motor thermal time constant in seconds

CurRmsLimit Current RMS limit for DC motor

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.66 int rtm_usb_Thermal_SchedNonRt (unsigned char *dev_nr*, unsigned int *period*)

Schedules Thermal Watchdog Task for DC Motors Connected to Onboard Amplifiers.

Parameters:

dev_nr RT-Motion USB Device Number

period number of non-RT task scheduler timer ticks

Return values:

<0> for Success

<Negative Values> for Errors

3.1.3.67 int rtm_usb_Version (unsigned char *dev_nr*, unsigned char * *version_str*)

Get RT-Motion USB Firmware Version.

Parameters:

dev_nr RT-Motion USB device number

version_str Pointer to version string

Return values:

<0> for Success

<*Negative* Values> for Errors

Index

ADC_NUM
rtmotion_usb_conf.h, 11

AMP_FASTDECAY
rtmotion_usb_conf.h, 11

AMP_MIX15DECAY
rtmotion_usb_conf.h, 11

AMP_MIX48DECAY
rtmotion_usb_conf.h, 11

AMP_SLOWDECAY
rtmotion_usb_conf.h, 11

AMPENABLEARM
rtmotion_usb_conf.h, 11

AMPENABLECPLD
rtmotion_usb_conf.h, 11

DADC_NUM
rtmotion_usb_conf.h, 11

DDAC_NUM
rtmotion_usb_conf.h, 11

ENC_CH_X
rtmotion_usb_conf.h, 11

ENC_CH_Y
rtmotion_usb_conf.h, 11

ENCODER_NUM
rtmotion_usb_conf.h, 11

ERR_INVALID_AMPCH
rtmotion_usb_conf.h, 12

ERR_INVALID_BOARD_VERSION
rtmotion_usb_conf.h, 12

ERR_INVALID_PID_ACT
rtmotion_usb_conf.h, 12

ERR_INVALID_PID_REF
rtmotion_usb_conf.h, 12

ERR_INVALID_PID_SENS
rtmotion_usb_conf.h, 12

ERR_INVALID_PWMSOURCE
rtmotion_usb_conf.h, 12

ERR_MAX_NONRT_TASKS
rtmotion_usb_conf.h, 12

ERR_MOTOR_SLIP
rtmotion_usb_conf.h, 12

ERR_MOTOR_THERMAL
rtmotion_usb_conf.h, 12

ERR_SCHED_ACTIVE
rtmotion_usb_conf.h, 12

ERR_SCHED_INVALIDTICK
rtmotion_usb_conf.h, 12

ERR_UNSUPPORTED
rtmotion_usb_conf.h, 13

ERROR_STATE_ERR
rtmotion_usb_conf.h, 13

ERROR_STATE_FATALERR
rtmotion_usb_conf.h, 13

ERROR_STATE_HALTER
rtmotion_usb_conf.h, 13

ERROR_STATE_NOERR
rtmotion_usb_conf.h, 13

HWPWMSOURCE
rtmotion_usb_conf.h, 13

MAX_CURRMS_SEC
rtmotion_usb_conf.h, 13

MAX_ERR_STORED
rtmotion_usb_conf.h, 13

MAX_NONRT_TASK
rtmotion_usb_conf.h, 13

MAX_ONBOARD_AMP_NUM
rtmotion_usb_conf.h, 13

NOPWMSOURCE
rtmotion_usb_conf.h, 13

ONBOARD_AMP_X
rtmotion_usb_conf.h, 14

ONBOARD_AMP_Y
rtmotion_usb_conf.h, 14

PID_ANALOG_REF
rtmotion_usb_conf.h, 14

PID_CNT_MAX
rtmotion_usb_conf.h, 14

PID_ENC_X_SENS
rtmotion_usb_conf.h, 14

PID_ENC_Y_SENS
rtmotion_usb_conf.h, 14

PID_ONBOARDAMPX_ACT
rtmotion_usb_conf.h, 14

PID_ONBOARDAMPY_ACT
rtmotion_usb_conf.h, 14

- rtm_usb_ADC_Dis
 - [rtmotion_usb_conf.h, 15](#)
- rtm_usb_ADC_EN
 - [rtmotion_usb_conf.h, 15](#)
- rtm_usb_Amp_Blank
 - [rtmotion_usb_conf.h, 15](#)
- rtm_usb_Amp_DecMode
 - [rtmotion_usb_conf.h, 15](#)
- rtm_usb_Amp_Dis
 - [rtmotion_usb_conf.h, 16](#)
- rtm_usb_Amp_DisablePwm
 - [rtmotion_usb_conf.h, 16](#)
- rtm_usb_Amp_EN
 - [rtmotion_usb_conf.h, 16](#)
- rtm_usb_Amp_EnablePwm
 - [rtmotion_usb_conf.h, 17](#)
- rtm_usb_Amp_ExtMode
 - [rtmotion_usb_conf.h, 17](#)
- rtm_usb_Amp_SetEnableSigPath
 - [rtmotion_usb_conf.h, 17](#)
- rtm_usb_Amp_SetPwmSrc
 - [rtmotion_usb_conf.h, 18](#)
- rtm_usb_Amp_SetPwmThreshold
 - [rtmotion_usb_conf.h, 18](#)
- rtm_usb_Amp_SetPwmThresholdGain
 - [rtmotion_usb_conf.h, 19](#)
- rtm_usb_Amp_SetSwPwmFreq
 - [rtmotion_usb_conf.h, 19](#)
- rtm_usb_Amp_SetSwPwmSigSrc
 - [rtmotion_usb_conf.h, 19](#)
- rtm_usb_Amp_Sleep
 - [rtmotion_usb_conf.h, 20](#)
- rtm_usb_BoardVersion
 - [rtmotion_usb_conf.h, 20](#)
- rtm_usb_CpldImageVersion
 - [rtmotion_usb_conf.h, 20](#)
- rtm_usb_dac7624_SetPort
 - [rtmotion_usb_conf.h, 21](#)
- rtm_usb_DAC_Dis
 - [rtmotion_usb_conf.h, 21](#)
- rtm_usb_DAC_EN
 - [rtmotion_usb_conf.h, 21](#)
- rtm_usb_DigIO_GetMask
 - [rtmotion_usb_conf.h, 21](#)
- rtm_usb_DigIO_SetMask
 - [rtmotion_usb_conf.h, 22](#)
- rtm_usb_Dread
 - [rtmotion_usb_conf.h, 22](#)
- rtm_usb_Dread_get
 - [rtmotion_usb_conf.h, 22](#)
- rtm_usb_Dread_ready
 - [rtmotion_usb_conf.h, 23](#)
- rtm_usb_Dsend
 - [rtmotion_usb_conf.h, 24](#)
- rtm_usb_Dsend_ready
 - [rtmotion_usb_conf.h, 24](#)
- rtm_usb_Enc_SetFreq
 - [rtmotion_usb_conf.h, 25](#)
- rtm_usb_EncAlg
 - [rtmotion_usb_conf.h, 25](#)
- rtm_usb_EncDis
 - [rtmotion_usb_conf.h, 25](#)
- rtm_usb_EncEN
 - [rtmotion_usb_conf.h, 26](#)
- rtm_usb_Err_Clr
 - [rtmotion_usb_conf.h, 26](#)
- rtm_usb_Err_Read
 - [rtmotion_usb_conf.h, 26](#)
- rtm_usb_get_dev_nr
 - [rtmotion_usb_conf.h, 27](#)
- rtm_usb_HCTL2032_Dis
 - [rtmotion_usb_conf.h, 27](#)
- rtm_usb_HCTL2032_En
 - [rtmotion_usb_conf.h, 27](#)
- rtm_usb_HCTL2032_Reset
 - [rtmotion_usb_conf.h, 28](#)
- rtm_usb_HCTL2032_SetRes
 - [rtmotion_usb_conf.h, 28](#)
- rtm_usb_init
 - [rtmotion_usb_conf.h, 28](#)
- rtm_usb_LED0
 - [rtmotion_usb_conf.h, 28](#)
- rtm_usb_LED1
 - [rtmotion_usb_conf.h, 29](#)
- rtm_usb_LED2
 - [rtmotion_usb_conf.h, 29](#)
- rtm_usb_Pid_DeleteAll
 - [rtmotion_usb_conf.h, 29](#)
- rtm_usb_Pid_Dis
 - [rtmotion_usb_conf.h, 29](#)
- rtm_usb_Pid_En
 - [rtmotion_usb_conf.h, 30](#)
- rtm_usb_Pid_EncConst
 - [rtmotion_usb_conf.h, 30](#)
- rtm_usb_Pid_NoActive
 - [rtmotion_usb_conf.h, 30](#)
- rtm_usb_Pid_RegCont
 - [rtmotion_usb_conf.h, 30](#)
- rtm_usb_Read_as5046
 - [rtmotion_usb_conf.h, 31](#)
- rtm_usb_RtmCount_DisCh
 - [rtmotion_usb_conf.h, 31](#)
- rtm_usb_RtmCount_DisCycRead
 - [rtmotion_usb_conf.h, 32](#)
- rtm_usb_RtmCount_EnCh
 - [rtmotion_usb_conf.h, 32](#)
- rtm_usb_RtmCount_EnCycRead
 - [rtmotion_usb_conf.h, 32](#)

- rtm_usb_RtmCount_Read
 - rtmotion_usb_conf.h, [33](#)
- rtm_usb_RtmCount_ResetCh
 - rtmotion_usb_conf.h, [33](#)
- rtm_usb_RtmCount_SetRes
 - rtmotion_usb_conf.h, [33](#)
- rtm_usb_RtmCount_SetSource
 - rtmotion_usb_conf.h, [34](#)
- rtm_usb_Sched_ClrList
 - rtmotion_usb_conf.h, [34](#)
- rtm_usb_Sched_Enable
 - rtmotion_usb_conf.h, [34](#)
- rtm_usb_Sched_SetTick
 - rtmotion_usb_conf.h, [35](#)
- rtm_usb_Slip_Config
 - rtmotion_usb_conf.h, [35](#)
- rtm_usb_Slip_SchedNonRt
 - rtmotion_usb_conf.h, [35](#)
- rtm_usb_terminate
 - rtmotion_usb_conf.h, [36](#)
- rtm_usb_Thermal_Config
 - rtmotion_usb_conf.h, [36](#)
- rtm_usb_Thermal_SchedNonRt
 - rtmotion_usb_conf.h, [36](#)
- rtm_usb_Version
 - rtmotion_usb_conf.h, [36](#)
- rtmotion_usb_conf.h, [8](#)
 - ADC_NUM, [11](#)
 - AMP_FASTDECAY, [11](#)
 - AMP_MIX15DECAY, [11](#)
 - AMP_MIX48DECAY, [11](#)
 - AMP_SLOWDECAY, [11](#)
 - AMPENABLEARM, [11](#)
 - AMPENABLECPLD, [11](#)
 - DADC_NUM, [11](#)
 - DDAC_NUM, [11](#)
 - ENC_CH_X, [11](#)
 - ENC_CH_Y, [11](#)
 - ENCODER_NUM, [11](#)
 - ERR_INVALID_AMPCH, [12](#)
 - ERR_INVALID_BOARD_VERSION, [12](#)
 - ERR_INVALID_PID_ACT, [12](#)
 - ERR_INVALID_PID_REF, [12](#)
 - ERR_INVALID_PID_SENS, [12](#)
 - ERR_INVALID_PWMSOURCE, [12](#)
 - ERR_MAX_NONRT_TASKS, [12](#)
 - ERR_MOTOR_SLIP, [12](#)
 - ERR_MOTOR_THERMAL, [12](#)
 - ERR_SCHED_ACTIVE, [12](#)
 - ERR_SCHED_INVALIDTICK, [12](#)
 - ERR_UNSUPPORTED, [13](#)
 - ERROR_STATE_ERR, [13](#)
 - ERROR_STATE_FATALERR, [13](#)
 - ERROR_STATE_HALTERR, [13](#)
 - ERROR_STATE_NOERR, [13](#)
 - HWPWMSOURCE, [13](#)
 - MAX_CURRMS_SEC, [13](#)
 - MAX_ERR_STORED, [13](#)
 - MAX_NONRT_TASK, [13](#)
 - MAX_ONBOARD_AMP_NUM, [13](#)
 - NOPWMSOURCE, [13](#)
 - ONBOARD_AMP_X, [14](#)
 - ONBOARD_AMP_Y, [14](#)
 - PID_ANALOG_REF, [14](#)
 - PID_CNT_MAX, [14](#)
 - PID_ENC_X_SENS, [14](#)
 - PID_ENC_Y_SENS, [14](#)
 - PID_ONBOARDAMPX_ACT, [14](#)
 - PID_ONBOARDAMPY_ACT, [14](#)
 - rtm_usb_ADC_Dis, [15](#)
 - rtm_usb_ADC_EN, [15](#)
 - rtm_usb_Amp_Blank, [15](#)
 - rtm_usb_Amp_DecMode, [15](#)
 - rtm_usb_Amp_Dis, [16](#)
 - rtm_usb_Amp_DisablePwm, [16](#)
 - rtm_usb_Amp_EN, [16](#)
 - rtm_usb_Amp_EnablePwm, [17](#)
 - rtm_usb_Amp_ExtMode, [17](#)
 - rtm_usb_Amp_SetEnableSigPath, [17](#)
 - rtm_usb_Amp_SetPwmSrc, [18](#)
 - rtm_usb_Amp_SetPwmThreshold, [18](#)
 - rtm_usb_Amp_SetPwmThresholdGain, [19](#)
 - rtm_usb_Amp_SetSwPwmFreq, [19](#)
 - rtm_usb_Amp_SetSwPwmSigSrc, [19](#)
 - rtm_usb_Amp_Sleep, [20](#)
 - rtm_usb_BoardVersion, [20](#)
 - rtm_usb_CpldImageVersion, [20](#)
 - rtm_usb_dac7624_SetPort, [21](#)
 - rtm_usb_DAC_Dis, [21](#)
 - rtm_usb_DAC_EN, [21](#)
 - rtm_usb_DigIO_GetMask, [21](#)
 - rtm_usb_DigIO_SetMask, [22](#)
 - rtm_usb_Dread, [22](#)
 - rtm_usb_Dread_get, [22](#)
 - rtm_usb_Dread_ready, [23](#)
 - rtm_usb_Dsend, [24](#)
 - rtm_usb_Dsend_ready, [24](#)
 - rtm_usb_Enc_SetFreq, [25](#)
 - rtm_usb_EncAlg, [25](#)
 - rtm_usb_EncDis, [25](#)
 - rtm_usb_EncEN, [26](#)
 - rtm_usb_Err_Clr, [26](#)
 - rtm_usb_Err_Read, [26](#)
 - rtm_usb_get_dev_nr, [27](#)
 - rtm_usb_HCTL2032_Dis, [27](#)
 - rtm_usb_HCTL2032_En, [27](#)
 - rtm_usb_HCTL2032_Reset, [28](#)
 - rtm_usb_HCTL2032_SetRes, [28](#)

- [rtm_usb_init](#), [28](#)
- [rtm_usb_LED0](#), [28](#)
- [rtm_usb_LED1](#), [29](#)
- [rtm_usb_LED2](#), [29](#)
- [rtm_usb_Pid_DeleteAll](#), [29](#)
- [rtm_usb_Pid_Dis](#), [29](#)
- [rtm_usb_Pid_En](#), [30](#)
- [rtm_usb_Pid_EncConst](#), [30](#)
- [rtm_usb_Pid_NoActive](#), [30](#)
- [rtm_usb_Pid_RegCont](#), [30](#)
- [rtm_usb_Read_as5046](#), [31](#)
- [rtm_usb_RtmCount_DisCh](#), [31](#)
- [rtm_usb_RtmCount_DisCycRead](#), [32](#)
- [rtm_usb_RtmCount_EnCh](#), [32](#)
- [rtm_usb_RtmCount_EnCycRead](#), [32](#)
- [rtm_usb_RtmCount_Read](#), [33](#)
- [rtm_usb_RtmCount_ResetCh](#), [33](#)
- [rtm_usb_RtmCount_SetRes](#), [33](#)
- [rtm_usb_RtmCount_SetSource](#), [34](#)
- [rtm_usb_Sched_ClrList](#), [34](#)
- [rtm_usb_Sched_Enable](#), [34](#)
- [rtm_usb_Sched_SetTick](#), [35](#)
- [rtm_usb_Slip_Config](#), [35](#)
- [rtm_usb_Slip_SchedNonRt](#), [35](#)
- [rtm_usb_terminate](#), [36](#)
- [rtm_usb_Thermal_Config](#), [36](#)
- [rtm_usb_Thermal_SchedNonRt](#), [36](#)
- [rtm_usb_Version](#), [36](#)
- [SCHED_TICK](#), [14](#)
- [SWPWMSOURCE](#), [14](#)

SCHED_TICK

- [rtmotion_usb_conf.h](#), [14](#)

SWPWMSOURCE

- [rtmotion_usb_conf.h](#), [14](#)