

RoboControllerSDK

Generated by Doxygen 1.8.4

Tue May 21 2013 15:13:13

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	_BoardStatus Struct Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Data Documentation	5
3.1.2.1	accelRampEnable	5
3.1.2.2	pidEnable	5
3.1.2.3	saveToEeprom	5
3.1.2.4	wdEnable	5
3.2	_RobotConfiguration Struct Reference	6
3.2.1	Detailed Description	6
3.2.2	Member Data Documentation	6
3.2.2.1	EncoderCprLeft	6
3.2.2.2	EncoderCprRight	6
3.2.2.3	EncoderPosition	6
3.2.2.4	Height	7
3.2.2.5	Lenght	7
3.2.2.6	MaxAmpereMotorLeft	7
3.2.2.7	MaxAmpereMotorRight	7
3.2.2.8	MaxRpmMotorLeft	7
3.2.2.9	MaxRpmMotorRight	7
3.2.2.10	MaxTorqueMotorLeft	7
3.2.2.11	MaxTorqueMotorRight	7
3.2.2.12	MotorEnableLevel	7
3.2.2.13	RatioMotorLeft	7
3.2.2.14	RatioMotorRight	7
3.2.2.15	RatioShaftLeft	7

3.2.2.16	RatioShaftRight	8
3.2.2.17	Weight	8
3.2.2.18	WheelBase	8
3.2.2.19	WheelRadiusLeft	8
3.2.2.20	WheelRadiusRight	8
3.2.2.21	Width	8
3.3	roboctrl::RcException Class Reference	8
3.3.1	Detailed Description	9
3.3.2	Constructor & Destructor Documentation	9
3.3.2.1	RcException	9
3.3.2.2	RcException	9
3.3.3	Member Function Documentation	9
3.3.3.1	GetMessage	9
3.3.3.2	GetType	9
3.3.3.3	operator const char *	9
3.4	roboctrl::RoboControllerSDK Class Reference	10
3.4.1	Member Function Documentation	11
3.4.1.1	getMotorPidGains	11
3.4.1.2	getMotorPWM	11
3.4.1.3	getMotorSpeed	11
3.4.1.4	getRobotConfigurationFromIni	12
3.4.1.5	run	12
3.4.1.6	saveRobotConfigurationToIni	12
3.4.1.7	setBoardStatus	12
3.4.1.8	setCommMode	12
3.4.1.9	setMotorPidGains	13
3.4.1.10	setMotorPWM	13
3.4.1.11	setMotorSpeed	13

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

<code>_BoardStatus</code>	5
<code>_RobotConfiguration</code>	6
<code>QThread</code>	
<code>roboctrl::RoboControllerSDK</code>	10
<code>roboctrl::RcException</code>	8

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

_BoardStatus	Used to maintain the state of the configuration of the board	5
_RobotConfiguration	Used to maintain the state of the configuration of the robot	6
roboctrl::RcException	A generic error exception that can be thrown by the application	8
roboctrl::RoboControllerSDK	10

Chapter 3

Class Documentation

3.1 `_BoardStatus` Struct Reference

Used to maintain the state of the configuration of the board.

```
#include <RoboControllerSDK_global.h>
```

Public Attributes

- bool `pidEnable`
- bool `wdEnable`
- bool `saveToEeprom`
- bool `accelRampEnable`

3.1.1 Detailed Description

Used to maintain the state of the configuration of the board.

3.1.2 Member Data Documentation

3.1.2.1 `bool _BoardStatus::accelRampEnable`

Indicates if acceleration is limited by speed ramps

3.1.2.2 `bool _BoardStatus::pidEnable`

Indicates if the motor PID controls are enabled

3.1.2.3 `bool _BoardStatus::saveToEeprom`

Indicates if the Board parameters are saved to Eeprom when changed

3.1.2.4 `bool _BoardStatus::wdEnable`

Indicates if the Command WatchDog is enabled. If it is true and the board does not receive command for N msec (`getWatchDogTime` and `setWatchDogTime`) the motors stop

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

- `RoboControllerSDK_global.h`

3.2 `_RobotConfiguration` Struct Reference

Used to maintain the state of the configuration of the robot.

```
#include <RoboControllerSDK_global.h>
```

Public Attributes

- quint16 [Weight](#)
- quint16 [Width](#)
- quint16 [Height](#)
- quint16 [Lenght](#)
- quint16 [WheelBase](#)
- quint16 [WheelRadiusLeft](#)
- quint16 [WheelRadiusRight](#)
- quint16 [EncoderCprLeft](#)
- quint16 [EncoderCprRight](#)
- quint16 [MaxRpmMotorLeft](#)
- quint16 [MaxRpmMotorRight](#)
- quint16 [MaxAmpereMotorLeft](#)
- quint16 [MaxAmpereMotorRight](#)
- quint16 [MaxTorqueMotorLeft](#)
- quint16 [MaxTorqueMotorRight](#)
- quint16 [RatioShaftLeft](#)
- quint16 [RatioShaftRight](#)
- quint16 [RatioMotorLeft](#)
- quint16 [RatioMotorRight](#)
- PinLevel [MotorEnableLevel](#)
- EncoderPos [EncoderPosition](#)

3.2.1 Detailed Description

Used to maintain the state of the configuration of the robot.

3.2.2 Member Data Documentation

3.2.2.1 `quint16 _RobotConfiguration::EncoderCprLeft`

Count per Round of the left encoder

3.2.2.2 `quint16 _RobotConfiguration::EncoderCprRight`

Count per Round of the right encoder

3.2.2.3 `EncoderPos _RobotConfiguration::EncoderPosition`

Encoder on the shaft of the motor or of the wheel

3.2.2.4 quint16 _RobotConfiguration::Height

Robot Height (mm)

3.2.2.5 quint16 _RobotConfiguration::Lenght

Robot Lenght (mm)

3.2.2.6 quint16 _RobotConfiguration::MaxAmpereMotorLeft

Max current assorbed by left motor (mA)

3.2.2.7 quint16 _RobotConfiguration::MaxAmpereMotorRight

Max current assorbed by right motor (mA)

3.2.2.8 quint16 _RobotConfiguration::MaxRpmMotorLeft

Max RPM of the left motor

3.2.2.9 quint16 _RobotConfiguration::MaxRpmMotorRight

Max RPM of the right motor

3.2.2.10 quint16 _RobotConfiguration::MaxTorqueMotorLeft

Max torque of the left motor (Ncm)

3.2.2.11 quint16 _RobotConfiguration::MaxTorqueMotorRight

Max torque of the right motor (Ncm)

3.2.2.12 PinLevel _RobotConfiguration::MotorEnableLevel

Enable Level of the Robot Driver (Low/High)

3.2.2.13 quint16 _RobotConfiguration::RatioMotorLeft

Reduction Ratio on the left Motor Shaft (Put 1 if you set Max RPM considering it just reduced)

3.2.2.14 quint16 _RobotConfiguration::RatioMotorRight

Reduction Ratio on the right Motor Shaft (Put 1 if you set Max RPM considering it just reduced)

3.2.2.15 quint16 _RobotConfiguration::RatioShaftLeft

Reduction Ratio from the shaft of the left motor to the shaft of the left wheel

3.2.2.16 quint16_RobotConfiguration::RatioShaftRight

Reduction Ratio from the shaft of the right motor to the shaft of the right wheel

3.2.2.17 quint16_RobotConfiguration::Weight

Robot Weight (g)

3.2.2.18 quint16_RobotConfiguration::WheelBase

Distance between the center of the Wheels (mm)

3.2.2.19 quint16_RobotConfiguration::WheelRadiusLeft

Radius of the left wheel (0.01mm)

3.2.2.20 quint16_RobotConfiguration::WheelRadiusRight

Radius of the right wheel (0.01mm)

3.2.2.21 quint16_RobotConfiguration::Width

Robot Width (mm)

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

- RoboControllerSDK_global.h

3.3 roboctrl::RcException Class Reference

A generic error exception that can be thrown by the application.

```
#include <exception.h>
```

Public Member Functions

- [RcException](#) (RcExceptionType type, const char *message,...)
- [RcException](#) (RcExceptionType type, const char *message, [RcException](#) *e,...)
- const char * [GetMessage](#) () const
The exception error message.
- [operator const char *](#) () const
The exception error message.
- RcExceptionType [GetType](#) ()
The exception error type.

Protected Attributes

- char **message** [1024]
- RcExceptionType **mExcType**

3.3.1 Detailed Description

A generic error exception that can be thrown by the application.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 `roboctrl::RcException::RcException (RcExceptionType type, const char * message, ...) [inline]`

Construct an exception

Parameters

<i>message</i>	It is a printf-like format string, arguments follows ;)
----------------	---

3.3.2.2 `roboctrl::RcException::RcException (RcExceptionType type, const char * message, RcException * e, ...) [inline]`

Construct an exception

Parameters

<i>message</i>	It is a printf-like format string, arguments follows ;)
----------------	---

3.3.3 Member Function Documentation

3.3.3.1 `const char* roboctrl::RcException::GetMessage () const [inline]`

The exception error message.

Returns

The error message associated to the exception

3.3.3.2 `RcExceptionType roboctrl::RcException::GetType () [inline]`

The exception error type.

Returns

The error type associated to the exception

3.3.3.3 `roboctrl::RcException::operator const char * () const [inline]`

The exception error message.

Returns

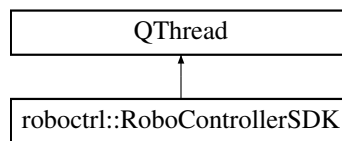
The error message associated to the exception

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

- exception.h

3.4 roboctrl::RoboControllerSDK Class Reference

Inheritance diagram for roboctrl::RoboControllerSDK:



Signals

- void [tcpConnected](#) ()
Signal emitted when TCP Socket is connected.
- void [tcpDisconnected](#) ()
Signal emitted when TCP Socket is disconnected.
- void [udpConnected](#) ()
Signal emitted when UDP Socket is connected.
- void [udpDisconnected](#) ()
Signal emitted when UDP Socket is disconnected.
- void [newMotorPwmValue](#) (quint16 motorIdx, quint16 value)
Signal emitted when a new PWM value is received.
- void [newMotorSpeedValue](#) (quint16 motorIdx, double value)
Signal emitted when a new SPEED value is received (speed is in m/sec)
- void [newMotorPwmValue](#) (quint16 motorIdx, double value)
Signal emitted when a new PWM value is received.
- void [newMotorPIDGains](#) (quint16 motorIdx, quint16 Kp, quint16 Ki, quint16 Kd)
Signal emitted when new PID gains are received.
- void [newRobotConfiguration](#) ([RobotConfiguration](#) &robConf)
Signal emitted when a new Robot Configuration is received from robot.
- void [newBoardStatus](#) ([BoardStatus](#) &status)
Signal emitted when a new Board Status is ready.

Public Member Functions

- **RoboControllerSDK** (int udpListenPort=4550, int udpSendPort=4560, QString tcpAddr=QString("localhost"), int tcpPort=4500)
- void [setCommMode](#) (CommMode mode)
Sets the current mode.
- void [getMotorSpeed](#) (quint16 motorIdx)
Send a request for motor speed. The reply is received with /ref newMotorSpeedValue signal.
- void [setMotorSpeed](#) (quint16 motorIdx, double speed)
Sets the speed of the motor in m/sec.
- void [getMotorPWM](#) (quint16 motorIdx)
Send a request for motor pwm. The reply is received with /ref newMotorPwmValue signal.
- void [setMotorPWM](#) (quint16 motorIdx, int pwm)
Sets the PWM of the motor in m/sec.
- void [setMotorPidGains](#) (quint16 motorIdx, quint16 Kp, quint16 Ki, quint16 Kd)
Sets motor PID Controllers parameters.
- void [getMotorPidGains](#) (quint16 motorIdx)

- *Send a request for motor PID gains. The reply is received with [newMotorPIDGains](#) signal.*
- void [getBoardStatus](#) ()
Gets current board status (BoardStatus) The reply is received with [newBoardStatus](#).
- void [setBoardStatus](#) (BoardStatus &status)
Sets a new BoardStatus.
- bool [getRobotConfigurationFromIni](#) (QString iniFile=ROBOT_CONFIG_INI_FILE)
Load the Robot Configuration from ini file The Robot configuration is not stored on Robot EEPROM unless saving is active.
- void [getRobotConfigurationFromEeprom](#) ()
Load the Robot Configuration from Robot EEPROM The Robot configuration is retrieved from Robot.
- void [saveRobotConfigurationToIni](#) (QString iniFile=ROBOT_CONFIG_INI_FILE)
Save the Robot Configuration to Robot EEPROM The Robot configuration is saved on the Robot.
- void [saveRobotConfigurationToEeprom](#) ()
*Save the Robot Configuration to Robot EEPROM The Robot configuration is saved on the Robot *.*

Protected Slots

- void [onTcpReadyRead](#) ()
- void [onTcpError](#) (QAbstractSocket::SocketError err)
- void [onTcpHostFound](#) ()
- void [onPingTimerTimeout](#) ()
Ping Timer handler.

Protected Member Functions

- virtual void [run](#) ()
Disables the Communication Watchdog WatchDog if active stops motors if communication is lost.

3.4.1 Member Function Documentation

3.4.1.1 void roboctrl::RoboControllerSDK::getMotorPidGains (quint16 motorIdx)

Send a request for motor PID gains. The reply is received with [newMotorPIDGains](#) signal.

Parameters

<i>motorIdx</i>	Index of the motor (0 or 1)
-----------------	-----------------------------

3.4.1.2 void roboctrl::RoboControllerSDK::getMotorPWM (quint16 motorIdx)

Send a request for motor pwm. The reply is received with [newMotorPwmValue](#) signal.

Parameters

<i>motorIdx</i>	Index of the motor (0 or 1)
-----------------	-----------------------------

3.4.1.3 void roboctrl::RoboControllerSDK::getMotorSpeed (quint16 motorIdx)

Send a request for motor speed. The reply is received with [newMotorSpeedValue](#) signal.

Parameters

<i>motorIdx</i>	Index of the motor (0 or 1)
-----------------	-----------------------------

3.4.1.4 `bool roboctrl::RoboControllerSDK::getRobotConfigurationFromIni (QString iniFile = ROBOT_CONFIG_INI_FILE)`

Load the Robot Configuration from ini file The Robot configuration is not stored on Robot EEPROM unless saving is active.

Parameters

<i>iniFile</i>	The path of the ini file
----------------	--------------------------

Returns

true if everything is ok

3.4.1.5 `void roboctrl::RoboControllerSDK::run () [protected],[virtual]`

Disables the Communication Watchdog WatchDog if active stops motors if communication is lost.

Enables the Communication Watchdog WatchDog if active stops motors if communication is lostThread function

3.4.1.6 `void roboctrl::RoboControllerSDK::saveRobotConfigurationToIni (QString iniFile = ROBOT_CONFIG_INI_FILE)`

Save the Robot Configuration to Robot EEPROM The Robot configuration is saved on the Robot.

Parameters

<i>iniFile</i>	The path of the ini file
----------------	--------------------------

3.4.1.7 `void roboctrl::RoboControllerSDK::setBoardStatus (BoardStatus & status)`

Sets a new BoardStatus.

Parameters

<i>status</i>	The new status to be set
---------------	--------------------------

3.4.1.8 `void roboctrl::RoboControllerSDK::setCommMode (CommMode mode)`

Sets the current mode.

Disables the Communication Watchdog.

Parameters

<i>mode</i>	The mode to be set. See CommMode
-------------	----------------------------------

Exceptions

<i>RcException</i>	Enables the Communication Watchdog
------------------------------------	------------------------------------

3.4.1.9 void roboctrl::RoboControllerSDK::setMotorPidGains (quint16 *motorIdx*, quint16 *Kp*, quint16 *Ki*, quint16 *Kd*)

Sets motor PID Controllers parameters.

Parameters

<i>motorIdx</i>	Index of the motor (0 or 1)
<i>Kp</i>	Proportional Action gain
<i>Ki</i>	Integral Action gain
<i>Kd</i>	Derivative Action gain

3.4.1.10 void roboctrl::RoboControllerSDK::setMotorPWM (quint16 *motorIdx*, int *pwm*)

Sets the PWM of the motor in m/sec.

Parameters

<i>motorIdx</i>	Index of the motor (0 or 1)
<i>pwm</i>	PWM range: [-2048/2047]

Note

This function works only when mMotorCtrlMode is equal to mcPid, else it does nothing

3.4.1.11 void roboctrl::RoboControllerSDK::setMotorSpeed (quint16 *motorIdx*, double *speed*)

Sets the speed of the motor in m/sec.

Parameters

<i>motorIdx</i>	Index of the motor (0 or 1)
<i>speed</i>	The speed in m/sec - Speed range: [-32.768/+32.767] m/sec

Note

This function works only when mMotorCtrlMode is equal to mcPid, else it does nothing

The documentation for this class was generated from the following files:

- robocontrollersdk.h
- robocontrollersdk.cpp

Index

- [_BoardStatus](#), [5](#)
 - [accelRampEnable](#), [5](#)
 - [pidEnable](#), [5](#)
 - [saveToEeprom](#), [5](#)
 - [wdEnable](#), [5](#)
 - [_RobotConfiguration](#), [6](#)
 - [EncoderCprLeft](#), [6](#)
 - [EncoderCprRight](#), [6](#)
 - [EncoderPosition](#), [6](#)
 - [Height](#), [6](#)
 - [Lenght](#), [7](#)
 - [MaxAmpereMotorLeft](#), [7](#)
 - [MaxAmpereMotorRight](#), [7](#)
 - [MaxRpmMotorLeft](#), [7](#)
 - [MaxRpmMotorRight](#), [7](#)
 - [MaxTorqueMotorLeft](#), [7](#)
 - [MaxTorqueMotorRight](#), [7](#)
 - [MotorEnableLevel](#), [7](#)
 - [RatioMotorLeft](#), [7](#)
 - [RatioMotorRight](#), [7](#)
 - [RatioShaftLeft](#), [7](#)
 - [RatioShaftRight](#), [7](#)
 - [Weight](#), [8](#)
 - [WheelBase](#), [8](#)
 - [WheelRadiusLeft](#), [8](#)
 - [WheelRadiusRight](#), [8](#)
 - [Width](#), [8](#)
- [accelRampEnable](#)
 - [_BoardStatus](#), [5](#)
- [EncoderCprLeft](#)
 - [_RobotConfiguration](#), [6](#)
- [EncoderCprRight](#)
 - [_RobotConfiguration](#), [6](#)
- [EncoderPosition](#)
 - [_RobotConfiguration](#), [6](#)
- [GetMessage](#)
 - [roboctrl::RcException](#), [9](#)
- [getMotorPWM](#)
 - [roboctrl::RoboControllerSDK](#), [11](#)
- [getMotorPidGains](#)
 - [roboctrl::RoboControllerSDK](#), [11](#)
- [getMotorSpeed](#)
 - [roboctrl::RoboControllerSDK](#), [11](#)
- [getRobotConfigurationFromIni](#)
 - [roboctrl::RoboControllerSDK](#), [12](#)
- [GetType](#)
 - [roboctrl::RcException](#), [9](#)
- [Height](#)
 - [_RobotConfiguration](#), [6](#)
- [Lenght](#)
 - [_RobotConfiguration](#), [7](#)
- [MaxAmpereMotorLeft](#)
 - [_RobotConfiguration](#), [7](#)
- [MaxAmpereMotorRight](#)
 - [_RobotConfiguration](#), [7](#)
- [MaxRpmMotorLeft](#)
 - [_RobotConfiguration](#), [7](#)
- [MaxRpmMotorRight](#)
 - [_RobotConfiguration](#), [7](#)
- [MaxTorqueMotorLeft](#)
 - [_RobotConfiguration](#), [7](#)
- [MaxTorqueMotorRight](#)
 - [_RobotConfiguration](#), [7](#)
- [MotorEnableLevel](#)
 - [_RobotConfiguration](#), [7](#)
- [operator const char *](#)
 - [roboctrl::RcException](#), [9](#)
- [pidEnable](#)
 - [_BoardStatus](#), [5](#)
- [RatioMotorLeft](#)
 - [_RobotConfiguration](#), [7](#)
- [RatioMotorRight](#)
 - [_RobotConfiguration](#), [7](#)
- [RatioShaftLeft](#)
 - [_RobotConfiguration](#), [7](#)
- [RatioShaftRight](#)
 - [_RobotConfiguration](#), [7](#)
- [RcException](#)
 - [roboctrl::RcException](#), [9](#)
- [roboctrl::RcException](#), [8](#)
 - [GetMessage](#), [9](#)
 - [GetType](#), [9](#)
 - [operator const char *](#), [9](#)
 - [RcException](#), [9](#)
- [roboctrl::RoboControllerSDK](#), [10](#)
 - [getMotorPWM](#), [11](#)
 - [getMotorPidGains](#), [11](#)
 - [getMotorSpeed](#), [11](#)
 - [getRobotConfigurationFromIni](#), [12](#)
 - [run](#), [12](#)
 - [saveRobotConfigurationToIni](#), [12](#)
 - [setBoardStatus](#), [12](#)
 - [setCommMode](#), [12](#)

- setMotorPWM, [13](#)
 - setMotorPidGains, [13](#)
 - setMotorSpeed, [13](#)
- run
 - roboctrl::RoboControllerSDK, [12](#)
- saveRobotConfigurationToIni
 - roboctrl::RoboControllerSDK, [12](#)
- saveToEeprom
 - _BoardStatus, [5](#)
- setBoardStatus
 - roboctrl::RoboControllerSDK, [12](#)
- setCommMode
 - roboctrl::RoboControllerSDK, [12](#)
- setMotorPWM
 - roboctrl::RoboControllerSDK, [13](#)
- setMotorPidGains
 - roboctrl::RoboControllerSDK, [13](#)
- setMotorSpeed
 - roboctrl::RoboControllerSDK, [13](#)
- wdEnable
 - _BoardStatus, [5](#)
- Weight
 - _RobotConfiguration, [8](#)
- WheelBase
 - _RobotConfiguration, [8](#)
- WheelRadiusLeft
 - _RobotConfiguration, [8](#)
- WheelRadiusRight
 - _RobotConfiguration, [8](#)
- Width
 - _RobotConfiguration, [8](#)