

SerialBasic

Generated by Doxygen 1.8.5

Sun Mar 2 2014 19:04:52

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	SerialBasic< Type > Class Template Reference	5
3.1.1	Detailed Description	5
3.1.2	Constructor & Destructor Documentation	6
3.1.2.1	SerialBasic	6
3.1.3	Member Function Documentation	6
3.1.3.1	getErrorCode	6
3.1.3.2	read	6
3.1.3.3	write	6
4	File Documentation	9
4.1	C:/Users/Andrew Powell/Documents/C/Learn Boost/SerialBasic/SerialBasic/SerialBasic.hpp File Reference	9
4.1.1	Detailed Description	9
4.1.2	LICENSE	10
	Index	11

Chapter 1

Class Index

1.1 Class List

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

SerialBasic< Type >	
Writes and reads data to a serial port with 8N1, for a particular Type	5

Chapter 2

File Index

2.1 File List

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

C:/Users/Andrew Powell/Documents/C/Learn Boost/SerialBasic/SerialBasic/[SerialBasic.hpp](#) 9

Chapter 3

Class Documentation

3.1 `SerialBasic< Type >` Class Template Reference

Writes and reads data to a serial port with 8N1, for a particular Type.

```
#include <SerialBasic.hpp>
```

Public Types

- typedef uint8_t **Byte**

Public Member Functions

- `SerialBasic` (uint16_t comPort, uint32_t baudRate)
Attempt to open serial port with a given comPort and baudRate.
- `~SerialBasic` ()
Destroy `SerialBasic` object.
- boost::system::error_code & `getErrorCode` ()
Get boost error code.
- template<class BeginIterator >
std::size_t `read` (BeginIterator beginIterator, std::size_t size)
Read serial data from the `SerialBasic` object's buffer (non-blocking)
- template<class BeginIterator >
void `write` (BeginIterator beginIterator, std::size_t size)
Write serial data to the serial port (blocking)

3.1.1 Detailed Description

```
template<class Type = uint8_t>class SerialBasic< Type >
```

Writes and reads data to a serial port with 8N1, for a particular Type.

Since `SerialBasic` was originally developed for communication with the Xbee transceivers, `SerialBasic` is generalized such that different data types can be serially transmitted. The default type of uint8_t is introduced with the assumption most developers will only transmit uint8_t arrays or other containers (e.g. vectors, lists, etc.).

Please note `SerialBasic` is developed with boost 1.55 and depends on boost threads (thus, ensure the boost compiled binaries are linked with the application utilizing `SerialBasic`). `SerialBasic` also depends on lambda functions from the C++0x standard. `SerialBasic` is also intended for Windows operating systems, however the constructor method can be modified for other operating systems supported by boost.

See Also

www.boost.org

3.1.2 Constructor & Destructor Documentation

3.1.2.1 `template<class Type> SerialBasic< Type>::SerialBasic (uint16_t comPort, uint32_t baudRate)`

Attempt to open serial port with a given comPort and baudRate.

Parameters

<i>comPort</i>	The COM port
<i>baudRate</i>	The baudrate

Exceptions

<i>boost::system::system_error</i>	Thrown if the attempt to open the serial port failed. Check boost error code to find out the reason of the failure.
------------------------------------	---

3.1.3 Member Function Documentation

3.1.3.1 `template<class Type> boost::system::error_code & SerialBasic< Type>::getErrorCode ()`

Get boost error code.

The error code can be used to verify whether a previously established connection is lost and why.

Returns

The boost error code.

3.1.3.2 `template<class Type> template<class BeginIterator> std::size_t SerialBasic< Type>::read (BeginIterator beginIterator, std::size_t size)`

Read serial data from the [SerialBasic](#) object's buffer (non-blocking)

Parameters

<i>beginIterator</i>	The starting location of where the data is saved. Can be a pointer to an array or an iterator of a container.
<i>size</i>	The maximum amount of data to copy from the SerialBasic object's buffer. For instance, if beginIterator is a pointer to an array of characters, size is the maximum amount of characters copied from the SerialBasic object's buffer. It is up to the developer to ensure the buffer to which beginIterator refers has the capacity to contain the maximum amount of requested data.

Returns

The actual amount of data saved.

3.1.3.3 `template<class Type> template<class BeginIterator> void SerialBasic< Type>::write (BeginIterator beginIterator, std::size_t size)`

Write serial data to the serial port (blocking)

Parameters

<i>beginIterator</i>	The starting location of where the data is taken. Can be a pointer to an array or an iterator of a container.
<i>size</i>	The maximum amount of data to write to the serial port. For instance, if beginIterator is a pointer to an array of characters, size is the maximum amount of characters written to the serial port associated with the SerialBasic object. It is up to the developer to ensure the buffer to which beginIterator refers is at least as large as the specified size.

Exceptions

<i>boost::system::system_error</i>	Thrown if there is a failure to write data to the serial port. Check boost error code to find out the reason of the failure.
------------------------------------	--

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

- C:/Users/Andrew Powell/Documents/C/Learn Boost/SerialBasic/SerialBasic/[SerialBasic.hpp](#)

Chapter 4

File Documentation

4.1 C:/Users/Andrew Powell/Documents/C/Learn Boost/SerialBasic/SerialBasic/Serial-Basic.hpp File Reference

```
#include <string>
#include <boost/thread.hpp>
#include <boost/bind.hpp>
#include <boost/asio.hpp>
#include <boost/system/error_code.hpp>
#include <stdint>
#include <list>
#include <sstream>
#include <iterator>
#include <memory>
```

Classes

- class [SerialBasic< Type >](#)
Writes and reads data to a serial port with 8N1, for a particular Type.
- class [SerialBasic< Type >](#)
Writes and reads data to a serial port with 8N1, for a particular Type.

Typedefs

- typedef [SerialBasic](#) **Serial**

4.1.1 Detailed Description

Author

Andrew Powell andrew.powell@temple.edu

Version

1.0

4.1.2 LICENSE

Copyright (C) 2014 Andrew Powell

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Distributed under the Boost Software License, Version 1.0. (See accompanying file LICENSE_1_0.txt or copy at http://www.boost.org/LICENSE_1_0.txt)

Index

C:/Users/Andrew Powell/Documents/C/Learn Boost/
SerialBasic/SerialBasic/SerialBasic.hpp, [9](#)

getErrorCode
SerialBasic, [6](#)

read
SerialBasic, [6](#)

SerialBasic
getErrorCode, [6](#)
read, [6](#)
SerialBasic, [6](#)
SerialBasic, [6](#)
write, [6](#)

SerialBasic< Type >, [5](#)

write
SerialBasic, [6](#)