

GCC Code Coverage Report

Directory: ./

File: [storage/blockdevice/source/ReadOnlyBlockDevice.cpp](#)

Date: 2021-05-06 12:39:05

	Exec	Total	Coverage
Lines:	32	33	97.0 %
Branches:	1	2	50.0 %

Line	Branch	Exec	Source
1			/* mbed Microcontroller Library
2			* Copyright (c) 2017-2020 ARM Limited
3			* SPDX-License-Identifier: Apache-2.0
4			*
5			* Licensed under the Apache License, Version 2.0 (the "License");
6			* you may not use this file except in compliance with the License.
7			* You may obtain a copy of the License at
8			*
9			* http://www.apache.org/licenses/LICENSE-2.0
10			*
11			* Unless required by applicable law or agreed to in writing, software
12			* distributed under the License is distributed on an "AS IS" BASIS,
13			* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14			* See the License for the specific language governing permissions and
15			* limitations under the License.
16			*/
17			
18			/** \addtogroup storage */
19			/** @{*/
20			
21			#include "blockdevice/ReadOnlyBlockDevice.h"
22			#include "platform/mbed_error.h"
23			
24			namespace mbed {
25			
26		4	ReadOnlyBlockDevice::ReadOnlyBlockDevice(BlockDevice *bd)
27		4	: _bd(bd)
28			{
29			// Does nothing
30		4	}

```

32 4 ReadOnlyBlockDevice::~ReadOnlyBlockDevice()
33 {
34     // Does nothing
35 4 }
36
37 2 int ReadOnlyBlockDevice::init()
38 {
39 2     return _bd->init();
40 }
41
42 2 int ReadOnlyBlockDevice::deinit()
43 {
44 2     return _bd->deinit();
45 }
46
47 1 int ReadOnlyBlockDevice::sync()
48 {
49 1     return _bd->sync();
50 }
51
52 1 int ReadOnlyBlockDevice::read(void *buffer, bd_addr_t addr, bd_size_t size)
53 {
54 1     return _bd->read(buffer, addr, size);
55 }
56
57 1 int ReadOnlyBlockDevice::program(const void *buffer, bd_addr_t addr, bd_size_t size)
58 {
59 1     return MBED_ERROR_WRITE_PROTECTED;
60 }
61
62 1 int ReadOnlyBlockDevice::erase(bd_addr_t addr, bd_size_t size)
63 {
64 1     return MBED_ERROR_WRITE_PROTECTED;
65 }
66
67 1 bd_size_t ReadOnlyBlockDevice::get_read_size() const
68 {
69 1     return _bd->get_read_size();
70 }

```

```

71 1 bd_size_t ReadOnlyBlockDevice::get_program_size() const
72 {
73 1 return _bd->get_program_size();
74 }
75
76 1 bd_size_t ReadOnlyBlockDevice::get_erase_size() const
77 {
78 1 return _bd->get_erase_size();
79 }
80
81 1 bd_size_t ReadOnlyBlockDevice::get_erase_size(bd_addr_t addr) const
82 {
83 1 return _bd->get_erase_size(addr);
84 }
85
86 1 int ReadOnlyBlockDevice::get_erase_value() const
87 {
88 1 return _bd->get_erase_value();
89 }
90
91 1 bd_size_t ReadOnlyBlockDevice::size() const
92 {
93 1 return _bd->size();
94 }
95
96 1 const char *ReadOnlyBlockDevice::get_type() const
97 {
98 1 if (_bd != NULL) {
99 1 return _bd->get_type();
100 }
101
102 return NULL;
103 }
104
105 } // namespace mbed
106
107 /** @}*/
108

```

Generated by: [GCOVR \(Version 4.2\)](#)

