Arduino Device Driver

Generated by Doxygen 1.8.9.1

Wed Aug 19 2015 01:06:31

ii CONTENTS

Contents

1	Hier	archical	I Index		1
	1.1	Class F	Hierarchy		. 1
2	Clas	s Index	t c		1
	2.1	Class L	List		. 1
3	File	Index			1
	3.1	File Lis	st		. 1
4	Clas	s Docur	mentation		2
	4.1	Device	e Class Reference		. 2
		4.1.1	Detailed Description		. 2
		4.1.2	Constructor & Destructor Documentation		. 2
	4.2	Eepron	mBasedWiredDevice Class Reference		. 2
		4.2.1	Detailed Description		. 4
		4.2.2	Member Enumeration Documentation		. 4
		4.2.3	Constructor & Destructor Documentation		. 4
		4.2.4	Member Function Documentation		. 4
		4.2.5	Member Data Documentation		. 5
	4.3	Registe	erBasedWiredDevice Class Reference		. 5
		4.3.1	Detailed Description		. 6
		4.3.2	Constructor & Destructor Documentation		. 7
		4.3.3	Member Function Documentation		. 7
		4.3.4	Member Data Documentation		. 7
	4.4	WiredD	Device Class Reference		. 8
		4.4.1	Detailed Description		. 8
		4.4.2	Constructor & Destructor Documentation		. 8
		4.4.3	Member Function Documentation		. 9
		4.4.4	Member Data Documentation		. 9
5	File	Docume	entation		ç
_	5.1		e.cpp File Reference		_
	5.2		e.cpp		
	5.3		e.h File Reference		
	5.4		e.h		
	5.5		mBasedWiredDevice.cpp File Reference		
	5.6		mBasedWiredDevice.cpp		
	5.7		mBasedWiredDevice.h File Reference		
	2-1	5.7.1	Macro Definition Documentation		
	5.8		mBasedWiredDevice.h		
				•	

1 Hierarchical Index

į	5.9 RegisterBasedWiredDevice.cpp File Reference	. 13
į	5.10 RegisterBasedWiredDevice.cpp	. 14
į	5.11 RegisterBasedWiredDevice.h File Reference	. 14
į	5.12 RegisterBasedWiredDevice.h	. 15
	5.13 WiredDevice.cpp File Reference	
į	5.14 WiredDevice.cpp	. 16
	5.15 WiredDevice.h File Reference	
	5.16 WiredDevice.h	. 17
Inde	ex	19
1	Hierarchical Index	
1.1	Class Hierarchy	
This	s inheritance list is sorted roughly, but not completely, alphabetically:	
I	Device	2
١	WiredDevice	8
	EepromBasedWiredDevice	2
	RegisterBasedWiredDevice	5
		Ĭ
2	Class Index	
2.1	Class List	
Here	re are the classes, structs, unions and interfaces with brief descriptions:	
ı	Device Arduino - Device	2
		2
	EepromBasedWiredDevice	
	RegisterBasedWiredDevice Arduino - Register Based Wire Device	5
\	WiredDevice Arduino - Wired Device	8
3	File Index	
3.1	File List	
Here	re is a list of all files with brief descriptions:	
I	Device.cpp	9
ı	Device.h	10

EepromBasedWiredDevice.cpp	10
EepromBasedWiredDevice.h	12
RegisterBasedWiredDevice.cpp	13
RegisterBasedWiredDevice.h	14
WiredDevice.cpp	16
WiredDevice.h	16

4 Class Documentation

4.1 Device Class Reference

```
#include <Device.h>
```

Public Member Functions

• Device ()

4.1.1 Detailed Description

Arduino - Device.

Device.cpp

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 12 of file Device.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Device::Device ()

Public constructor.

Parameters

address The wire address.

Definition at line 3 of file Device.cpp.

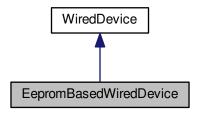
The documentation for this class was generated from the following files:

- Device.h
- Device.cpp

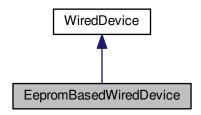
4.2 EepromBasedWiredDevice Class Reference

#include <EepromBasedWiredDevice.h>

Inheritance diagram for EepromBasedWiredDevice:



Collaboration diagram for EepromBasedWiredDevice:



Public Types

• enum { LITTLE_ENDIAN = 0x00, BIG_ENDIAN = 0x01 }

Public Member Functions

- EepromBasedWiredDevice (unsigned char deviceAddress, char addressSize, unsigned char endianness)
- EepromBasedWiredDevice (unsigned char deviceAddress, char addressSize)
- EepromBasedWiredDevice (unsigned char deviceAddress)
- void writeBlock (unsigned int address, unsigned char *buf, int len)
- void readBlock (unsigned int address, unsigned char *buf, int len)
- void setAddressSize (char addressSize)
- char getAddressSize ()

Public Attributes

• enum EepromBasedWiredDevice:: { ... } Endianness

Private Attributes

- char addressSize
- unsigned char endianness

Static Private Attributes

static const char MAX_RETRIES_ON_READING = 0x7f

4.2.1 Detailed Description

Definition at line 16 of file EepromBasedWiredDevice.h.

- 4.2.2 Member Enumeration Documentation
- 4.2.2.1 anonymous enum

Enumerator

LITTLE_ENDIAN BIG_ENDIAN

Definition at line 25 of file EepromBasedWiredDevice.h.

- 4.2.3 Constructor & Destructor Documentation
- 4.2.3.1 EepromBasedWiredDevice::EepromBasedWiredDevice (unsigned char *deviceAddress*, char *addressSize*, unsigned char *endianness*)

Public constructor.

Parameters

Γ	deviceAddress	The wire address.
	addressSize	How long is the internal device addresses.
	endianness	The endianness.

Definition at line 5 of file EepromBasedWiredDevice.cpp.

4.2.3.2 EepromBasedWiredDevice::EepromBasedWiredDevice (unsigned char deviceAddress, char addressSize)

Public constructor.

Definition at line 9 of file EepromBasedWiredDevice.cpp.

4.2.3.3 EepromBasedWiredDevice::EepromBasedWiredDevice (unsigned char deviceAddress)

Public constructor.

Definition at line 13 of file EepromBasedWiredDevice.cpp.

- 4.2.4 Member Function Documentation
- 4.2.4.1 char EepromBasedWiredDevice::getAddressSize ()

Gets the address size.

addressSize

Definition at line 55 of file EepromBasedWiredDevice.cpp.

4.2.4.2 void EepromBasedWiredDevice::readBlock (unsigned int address, unsigned char * buf, int len)

Reads a block of bytes from the device.

Sends the address MSB fist.

Parameters

address	
buf	
len	

Definition at line 29 of file EepromBasedWiredDevice.cpp.

4.2.4.3 void EepromBasedWiredDevice::setAddressSize (char addressSize)

Sets the address size.

Parameters

addressSize	

Definition at line 51 of file EepromBasedWiredDevice.cpp.

4.2.4.4 void EepromBasedWiredDevice::writeBlock (unsigned int address, unsigned char * buf, int len)

Writes a block of bytes separately by pages to the device.

Parameters

address	
buf	
len	

Definition at line 17 of file EepromBasedWiredDevice.cpp.

4.2.5 Member Data Documentation

4.2.5.1 char EepromBasedWiredDevice::addressSize [private]

Definition at line 20 of file EepromBasedWiredDevice.h.

4.2.5.2 unsigned char EepromBasedWiredDevice::endianness [private]

Definition at line 21 of file EepromBasedWiredDevice.h.

4.2.5.3 enum { ... } EepromBasedWiredDevice::Endianness

4.2.5.4 const char EepromBasedWiredDevice::MAX_RETRIES_ON_READING = 0x7f [static], [private]

Definition at line 18 of file EepromBasedWiredDevice.h.

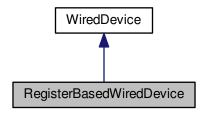
The documentation for this class was generated from the following files:

- EepromBasedWiredDevice.h
- EepromBasedWiredDevice.cpp

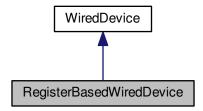
4.3 RegisterBasedWiredDevice Class Reference

#include <RegisterBasedWiredDevice.h>

Inheritance diagram for RegisterBasedWiredDevice:



Collaboration diagram for RegisterBasedWiredDevice:



Public Member Functions

- RegisterBasedWiredDevice (unsigned char address)
- void configureRegisterBits (unsigned char reg, unsigned char mask, unsigned char d)
- unsigned char writeRegister (unsigned char reg, unsigned char d)
- int readRegister (unsigned char reg)

Static Private Attributes

• static const unsigned char MAX_RETRIES_ON_READING = 10

4.3.1 Detailed Description

Arduino - Register Based Wire Device.

RegisterBasedWireDevice.cpp

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 14 of file RegisterBasedWiredDevice.h.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 RegisterBasedWiredDevice::RegisterBasedWiredDevice (unsigned char address)

Public constructor.

Parameters

address	The wire address.

Definition at line 5 of file RegisterBasedWiredDevice.cpp.

4.3.3 Member Function Documentation

4.3.3.1 void RegisterBasedWiredDevice::configureRegisterBits (unsigned char reg, unsigned char mask, unsigned char d)

Configures a register.

Parameters

reg	The register number.
mask	The mask to be used.
d	The value to be used.

Definition at line 10 of file RegisterBasedWiredDevice.cpp.

4.3.3.2 int RegisterBasedWiredDevice::readRegister (unsigned char reg)

Reades a value from a register.

Parameters

reg	The register number.
-----	----------------------

Returns

The register value.

Definition at line 27 of file RegisterBasedWiredDevice.cpp.

4.3.3.3 unsigned char RegisterBasedWiredDevice::writeRegister (unsigned char reg, unsigned char d)

Writes a value to a register.

Parameters

reg	The register number.
d	The value to be used.

Definition at line 19 of file RegisterBasedWiredDevice.cpp.

4.3.4 Member Data Documentation

4.3.4.1 const unsigned char RegisterBasedWiredDevice::MAX_RETRIES_ON_READING = 10 [static], [private]

Definition at line 16 of file RegisterBasedWiredDevice.h.

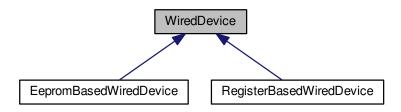
The documentation for this class was generated from the following files:

- · RegisterBasedWiredDevice.h
- RegisterBasedWiredDevice.cpp

4.4 WiredDevice Class Reference

#include <WiredDevice.h>

Inheritance diagram for WiredDevice:



Public Member Functions

- WiredDevice (unsigned char deviceAddress)
- unsigned char getDeviceAddress ()
- void setDeviceAddress (unsigned char deviceAddress)

Private Attributes

• unsigned char deviceAddress

4.4.1 Detailed Description

Arduino - Wired Device.

WiredDevice.cpp

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 12 of file WiredDevice.h.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 WiredDevice::WiredDevice (unsigned char deviceAddress)

Public constructor.

Parameters

address	The wire address.

Definition at line 4 of file WiredDevice.cpp.

5 File Documentation 9

4.4.3 Member Function Documentation

4.4.3.1 unsigned char WiredDevice::getDeviceAddress ()

Gets the device address.

Returns

address

Definition at line 9 of file WiredDevice.cpp.

4.4.3.2 void WiredDevice::setDeviceAddress (unsigned char deviceAddress)

Sets the device address.

Parameters

address The device address.

Definition at line 13 of file WiredDevice.cpp.

4.4.4 Member Data Documentation

4.4.4.1 unsigned char WiredDevice::deviceAddress [private]

Definition at line 14 of file WiredDevice.h.

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

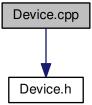
- WiredDevice.h
- WiredDevice.cpp

5 File Documentation

5.1 Device.cpp File Reference

```
#include "Device.h"
```

Include dependency graph for Device.cpp:



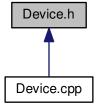
5.2 Device.cpp

00001 #include "Device.h"

```
00002
00003 Device::Device() {
00004 }
```

5.3 Device.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

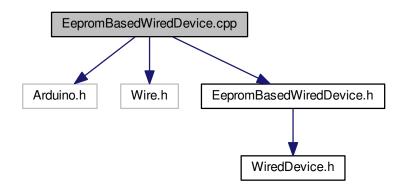
class Device

5.4 Device.h

5.5 EepromBasedWiredDevice.cpp File Reference

```
#include <Arduino.h>
#include <Wire.h>
#include "EepromBasedWiredDevice.h"
```

Include dependency graph for EepromBasedWiredDevice.cpp:



5.6 EepromBasedWiredDevice.cpp

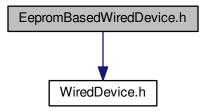
```
00001 #include <Arduino.h>
00002 #include <Wire.h>
00003 #include "EepromBasedWiredDevice.h"
00004
{\tt 00005\ EepromBasedWiredDevice:: EepromBasedWiredDevice (unsigned\ charges)}
     deviceAddress, char addressSize, unsigned char endianness)
00006
              : WiredDevice (deviceAddress), addressSize (addressSize), endianness (endianness) {
00007 }
00008
00009 EepromBasedWiredDevice::EepromBasedWiredDevice(unsigned char
      deviceAddress, char addressSize)
00010
              : EepromBasedWiredDevice(deviceAddress, addressSize, LITTLE ENDIAN) {
00011 }
00012
00013 EepromBasedWiredDevice::EepromBasedWiredDevice(unsigned char
      deviceAddress)
00014
              : EepromBasedWiredDevice(deviceAddress, 0x02) {
00015 }
00016
00017 void EepromBasedWiredDevice::writeBlock(unsigned int address, unsigned
00018
          Wire.beginTransmission(getDeviceAddress());
00019
          for (char i = addressSize; i > 0; i--) {
              Wire.write((unsigned char) (address >> ((i - 1) \star 8)) & 0xff);
00020
00021
00022
          for (int i = 0; i < len; i++) {</pre>
00023
              Wire.write(buf[i]);
00024
00025
          Wire.endTransmission();
00026
          delay(EEPROM_BASED_WIRED_DEVICE_AFTER_WRITE_DELAY);
00027 }
00028
00029 void EepromBasedWiredDevice::readBlock(unsigned int address, unsigned char
      * buf, int len) {
00030
          char tries;
00031
          unsigned char last = len - 1;
00032
          Wire.beginTransmission(getDeviceAddress());
00033
          for (char i = addressSize; i > 0; i--) {
00034
              Wire.write((unsigned char) (address >> ((i - 1) * 8)) & 0xff);
00035
00036
          Wire.endTransmission();
          delay(EEPROM_BASED_WIRED_DEVICE_AFTER_WRITE_DELAY);
00037
          Wire.requestFrom((int) getDeviceAddress(), len);
00038
          for (int i = 0; i < len; i++) {
   tries = MAX_RETRIES_ON_READING;</pre>
00039
00040
00041
              while (!Wire.available() && --tries > 0) {
                  delayMicroseconds(1);
00042
00043
              if (tries == 0) {
00044
00045
00046
00047
              buf[(endianness == BIG_ENDIAN) ? last - i : i] = Wire.read();
```

```
00048     }
00049 }
00050
00051 void EepromBasedWiredDevice::setAddressSize(char addressSize) {
00052     this->addressSize = addressSize;
00053 }
00054
00055 char EepromBasedWiredDevice::getAddressSize() {
00056     return addressSize;
00057 }
```

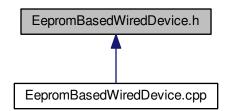
5.7 EepromBasedWiredDevice.h File Reference

```
#include <WiredDevice.h>
```

Include dependency graph for EepromBasedWiredDevice.h:



This graph shows which files directly or indirectly include this file:



Classes

• class EepromBasedWiredDevice

Macros

#define EEPROM_BASED_WIRED_DEVICE_AFTER_WRITE_DELAY 5

5.7.1 Macro Definition Documentation

5.7.1.1 #define EEPROM_BASED_WIRED_DEVICE_AFTER_WRITE_DELAY 5

Arduino - Register Based Wire Device.

RegisterBasedWireDevice.cpp

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 12 of file EepromBasedWiredDevice.h.

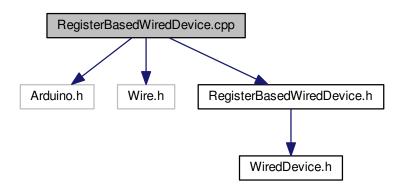
5.8 EepromBasedWiredDevice.h

```
00001
00009 #ifndef __ARDUINO_DRIVER_EEPROM_BASED_WIRED_DEVICE_H_
00010 #define __ARDUINO_DRIVER_EEPROM_BASED_WIRED_DEVICE_H_ 1
00011
00012 #define EEPROM_BASED_WIRED_DEVICE_AFTER_WRITE_DELAY
00013
00014 #include <WiredDevice.h>
00015
00016 class EepromBasedWiredDevice: public WiredDevice {
00017
00018
          const static char MAX_RETRIES_ON_READING = 0x7f;
00019
00020
          char addressSize;
00021
          unsigned char endianness:
00022
00023 public:
00024
00025
          enum {
             LITTLE_ENDIAN = 0 \times 00,
00026
00027
              BIG\_ENDIAN = 0x01
00028
          } Endianness;
00029
00037
          EepromBasedWiredDevice(unsigned char deviceAddress, char addressSize
      , unsigned char endianness);
00038
          EepromBasedWiredDevice(unsigned char deviceAddress, char addressSize
00042
     );
00043
00047
          EepromBasedWiredDevice(unsigned char deviceAddress);
00048
00056
          void writeBlock(unsigned int address, unsigned char* buf, int len);
00057
00067
          void readBlock(unsigned int address, unsigned char* buf, int len);
00068
00074
          void setAddressSize(char addressSize);
00075
00081
          char getAddressSize();
00082 };
00083
00084 #endif /* __ARDUINO_DRIVER_EEPROM_BASED_WIRED_DEVICE_H_ */
```

5.9 RegisterBasedWiredDevice.cpp File Reference

```
#include <Arduino.h>
#include <Wire.h>
#include "RegisterBasedWiredDevice.h"
```

Include dependency graph for RegisterBasedWiredDevice.cpp:



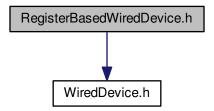
5.10 RegisterBasedWiredDevice.cpp

```
00001 #include <Arduino.h>
00002 #include <Wire.h>
00003 #include "RegisterBasedWiredDevice.h"
00004
00005 RegisterBasedWiredDevice::RegisterBasedWiredDevice(
00006
              unsigned char address) :
00007
              WiredDevice(address) {
00008 }
00009
00010 void RegisterBasedWiredDevice::configureRegisterBits(
     unsigned char reg,
            unsigned char mask, unsigned char d) {
00012
          unsigned char n;
00013
          n = readRegister(reg);
00014
          n &= ~(mask);
00015
          n |= d & mask;
00016
          writeRegister(reg, n);
00017 }
00018
00019 unsigned char RegisterBasedWiredDevice::writeRegister(unsigned char
00020
               unsigned char d) {
00021
          Wire.beginTransmission(getDeviceAddress());
00022
          Wire.write(reg);
00023
          Wire.write(d);
00024
          return Wire.endTransmission();
00025 }
00026
00027 int RegisterBasedWiredDevice::readRegister(unsigned char reg) {
          char tries = MAX_RETRIES_ON_READING;
00028
00029
          Wire.beginTransmission(getDeviceAddress());
00030
          Wire.write(reg);
00031
          char status = Wire.endTransmission(false);
          if (status != 0) {
00032
00033
              return -(status);
00034
          Wire.requestFrom(getDeviceAddress(), (unsigned char) 1);
while (!Wire.available() && --tries > 0) {
00035
00036
00037
              delayMicroseconds(1);
00038
00039
          if (tries == 0)
              return -5;
00040
00041
00042
          return Wire.read();
00043 }
```

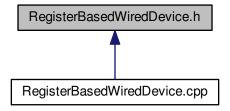
5.11 RegisterBasedWiredDevice.h File Reference

#include <WiredDevice.h>

Include dependency graph for RegisterBasedWiredDevice.h:



This graph shows which files directly or indirectly include this file:



Classes

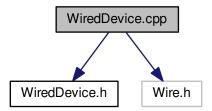
· class RegisterBasedWiredDevice

5.12 RegisterBasedWiredDevice.h

```
00001
00009 #ifndef __ARDUINO_DRIVER_REGISTER_BASED_WIRED_DEVICE_H_
00010 #define __ARDUINO_DRIVER_REGISTER_BASED_WIRED_DEVICE_H_ 1
00012 #include <WiredDevice.h>
00013
00014 class RegisterBasedWiredDevice: public WiredDevice {
00015
00016
          const static unsigned char MAX_RETRIES_ON_READING = 10;
00017
00018 public:
00019
00025
          RegisterBasedWiredDevice(unsigned char address);
00026
         void configureRegisterBits(unsigned char reg, unsigned char mask,
00034
00035
                  unsigned char d);
00036
00043
          unsigned char writeRegister(unsigned char reg, unsigned char d);
00044
00051
          int readRegister(unsigned char reg);
00052 };
00053
00054 #endif /* __ARDUINO_DRIVER_REGISTER_BASED_WIRED_DEVICE_H__ */
```

5.13 WiredDevice.cpp File Reference

```
#include "WiredDevice.h"
#include <Wire.h>
Include dependency graph for WiredDevice.cpp:
```

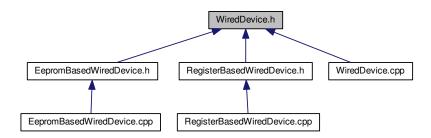


5.14 WiredDevice.cpp

```
00001 #include "WiredDevice.h" 00002 #include <Wire.h>
00003
00004 WiredDevice::WiredDevice(unsigned char deviceAddress)
00005
                : deviceAddress(deviceAddress) {
00006
           Wire.begin();
00007 }
80000
00009 unsigned char WiredDevice::getDeviceAddress() {
00010
            return deviceAddress;
00011 }
00012
00013 void WiredDevice::setDeviceAddress(unsigned char deviceAddress) {
00014 this->deviceAddress = deviceAddress;
00015 }
```

5.15 WiredDevice.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

• class WiredDevice

5.16 WiredDevice.h

5.16 WiredDevice.h

```
00001
00009 #ifndef _ARDUINO_DRIVER_WIRED_DEVICE_H_
00010 #define _ARDUINO_DRIVER_WIRED_DEVICE_H_ 1
00011
00012 class WiredDevice {
00013
00014
           unsigned char deviceAddress;
00015
00013
00016 public:
00017
00023
             WiredDevice(unsigned char deviceAddress);
00024
00030
             unsigned char getDeviceAddress();
00031
00037
             void setDeviceAddress(unsigned char deviceAddress);
00038 };
00039
00040 #endif /* __ARDUINO_DRIVER_WIRED_DEVICE_H_ */
```

Index

addressSize	configureRegisterBits, 7
EepromBasedWiredDevice, 5	MAX_RETRIES_ON_READING, 7
DIO ENDIANI	readRegister, 7
BIG_ENDIAN	RegisterBasedWiredDevice, 7
EepromBasedWiredDevice, 4	writeRegister, 7
configureRegisterBits	RegisterBasedWiredDevice.cpp, 13, 14
RegisterBasedWiredDevice, 7	RegisterBasedWiredDevice.h, 14, 15
riogistor Bassarrii Gaborico, 7	setAddressSize
Device, 2	EepromBasedWiredDevice, 5
Device, 2	setDeviceAddress
Device.cpp, 9	WiredDevice, 9
Device.h, 10	,-
deviceAddress	WiredDevice, 8
WiredDevice, 9	deviceAddress, 9
FEDROM BASED WIDED DEVICE AFTER WRIT	getDeviceAddress, 9
EEPROM_BASED_WIRED_DEVICE_AFTER_WRIT← E DELAY	setDeviceAddress, 9
E_DELAT EepromBasedWiredDevice.h, 12	WiredDevice, 8
EepromBasedWiredDevice, 2	WiredDevice.cpp, 16
addressSize, 5	WiredDevice.h, 16, 17
BIG_ENDIAN, 4	writeBlock
EepromBasedWiredDevice, 4	EepromBasedWiredDevice, 5
Endianness, 5	writeRegister RegisterBasedWiredDevice, 7
endianness, 5	negisterbasedwiredbevice, 7
getAddressSize, 4	
LITTLE_ENDIAN, 4	
MAX_RETRIES_ON_READING, 5	
readBlock, 4	
setAddressSize, 5	
writeBlock, 5	
EepromBasedWiredDevice.cpp, 10, 11	
EepromBasedWiredDevice.h, 12, 13	
EEPROM_BASED_WIRED_DEVICE_AFTER_←	
WRITE_DELAY, 12	
Endianness ForcemPaged Wired Davies 5	
EepromBasedWiredDevice, 5 endianness	
EepromBasedWiredDevice, 5	
Ecprombased vireabevide, o	
getAddressSize	
EepromBasedWiredDevice, 4	
getDeviceAddress	
WiredDevice, 9	
LITTLE_ENDIAN	
EepromBasedWiredDevice, 4	
MAX_RETRIES_ON_READING	
EepromBasedWiredDevice, 5	
RegisterBasedWiredDevice, 7	
- 0.5.5. = 5.5. = 5.00,	
readBlock	
EepromBasedWiredDevice, 4	
readRegister	
RegisterBasedWiredDevice, 7	
RegisterBasedWiredDevice, 5	