

## Arduino Device Driver

Generated by Doxygen 1.8.9.1

Tue Aug 18 2015 22:51:25

## Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>1</b>
2.1	Class List . . . . .	1
<b>3</b>	<b>File Index</b>	<b>1</b>
3.1	File List . . . . .	1
<b>4</b>	<b>Class Documentation</b>	<b>2</b>
4.1	Device Class Reference . . . . .	2
4.1.1	Detailed Description . . . . .	2
4.1.2	Constructor & Destructor Documentation . . . . .	2
4.2	EepromBasedWiredDevice 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	RegisterBasedWiredDevice 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	WiredDevice 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
<b>5</b>	<b>File Documentation</b>	<b>9</b>
5.1	Device.cpp File Reference . . . . .	9
5.2	Device.cpp . . . . .	9
5.3	Device.h File Reference . . . . .	10
5.4	Device.h . . . . .	10
5.5	EepromBasedWiredDevice.cpp File Reference . . . . .	10
5.6	EepromBasedWiredDevice.cpp . . . . .	11
5.7	EepromBasedWiredDevice.h File Reference . . . . .	12
5.7.1	Macro Definition Documentation . . . . .	12
5.8	EepromBasedWiredDevice.h . . . . .	13

5.9	<a href="#">RegisterBasedWiredDevice.cpp File Reference</a>	13
5.10	<a href="#">RegisterBasedWiredDevice.cpp</a>	14
5.11	<a href="#">RegisterBasedWiredDevice.h File Reference</a>	14
5.12	<a href="#">RegisterBasedWiredDevice.h</a>	15
5.13	<a href="#">WiredDevice.cpp File Reference</a>	16
5.14	<a href="#">WiredDevice.cpp</a>	16
5.15	<a href="#">WiredDevice.h File Reference</a>	16
5.16	<a href="#">WiredDevice.h</a>	17
<b>Index</b>		<b>19</b>

## 1 Hierarchical Index

### 1.1 Class Hierarchy

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

<b>Device</b>	<b>2</b>
<b>WiredDevice</b>	<b>8</b>
<b>EepromBasedWiredDevice</b>	<b>2</b>
<b>RegisterBasedWiredDevice</b>	<b>5</b>

## 2 Class Index

### 2.1 Class List

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

<b>Device</b>	
<b>Arduino - Device</b>	<b>2</b>
<b>EepromBasedWiredDevice</b>	<b>2</b>
<b>RegisterBasedWiredDevice</b>	
<b>Arduino - Register Based Wire Device</b>	<b>5</b>
<b>WiredDevice</b>	
<b>Arduino - Wired Device</b>	<b>8</b>

## 3 File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<b>Device.cpp</b>	<b>9</b>
<b>Device.h</b>	<b>10</b>

<a href="#">EepromBasedWiredDevice.cpp</a>	10
<a href="#">EepromBasedWiredDevice.h</a>	12
<a href="#">RegisterBasedWiredDevice.cpp</a>	13
<a href="#">RegisterBasedWiredDevice.h</a>	14
<a href="#">WiredDevice.cpp</a>	16
<a href="#">WiredDevice.h</a>	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](mailto: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

<i>address</i>	The wire address.
----------------	-------------------

Definition at line 3 of file [Device.cpp](#).

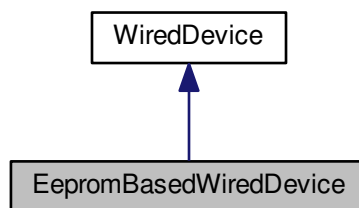
The documentation for this class was generated from the following files:

- [Device.h](#)
- [Device.cpp](#)

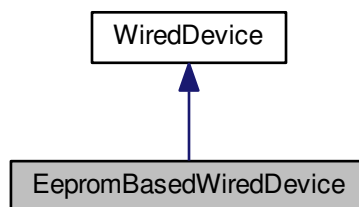
### 4.2 EepromBasedWiredDevice Class Reference

```
#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

<i>deviceAddress</i>	The wire address.
<i>addressSize</i>	How long is the internal device addresses.
<i>endianness</i>	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

<i>address</i>	
<i>buf</i>	
<i>len</i>	

Definition at line 29 of file [EepromBasedWiredDevice.cpp](#).

#### 4.2.4.3 void EepromBasedWiredDevice::setAddressSize ( char *addressSize* )

Sets the address size.

## Parameters

<i>addressSize</i>	
--------------------	--

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

<i>address</i>	
<i>buf</i>	
<i>len</i>	

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](#).

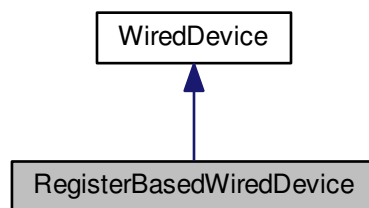
The documentation for this class was generated from the following files:

- [EepromBasedWiredDevice.h](#)
- [EepromBasedWiredDevice.cpp](#)

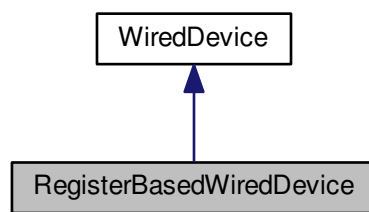
## 4.3 RegisterBasedWiredDevice Class Reference

```
#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](mailto: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

<i>address</i>	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

<i>reg</i>	The register number.
<i>mask</i>	The mask to be used.
<i>d</i>	The value to be used.

Definition at line 10 of file [RegisterBasedWiredDevice.cpp](#).

##### 4.3.3.2 int RegisterBasedWiredDevice::readRegister ( unsigned char *reg* )

Reads a value from a register.

Parameters

<i>reg</i>	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

<i>reg</i>	The register number.
<i>d</i>	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](#).

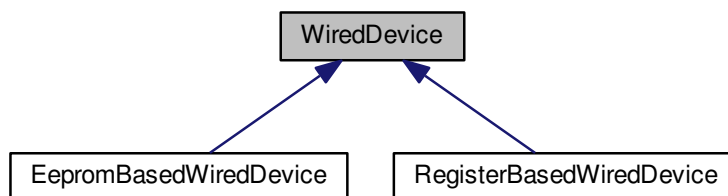
The documentation for this class was generated from the following files:

- [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](mailto: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

<i>address</i>	The wire address.
----------------	-------------------

Definition at line 4 of file [WiredDevice.cpp](#).

#### 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

<i>address</i>	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:

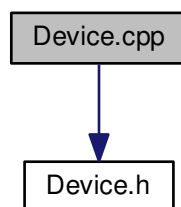
- [WiredDevice.h](#)
- [WiredDevice.cpp](#)

## 5 File Documentation

### 5.1 Device.cpp File Reference

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

Include dependency graph for Device.cpp:



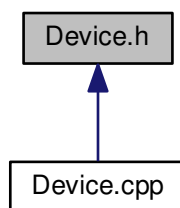
### 5.2 Device.cpp

```
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](#)

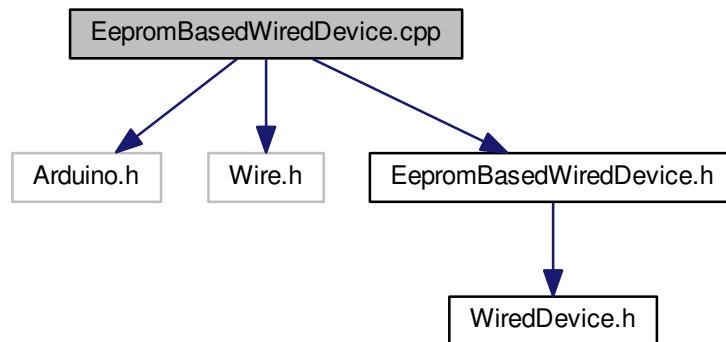
### 5.4 Device.h

```
00001
00009 #ifndef __ARDUINO_DRIVER_DEVICE_H__
00010 #define __ARDUINO_DRIVER_DEVICE_H__ 1
00011
00012 class Device {
00013
00014 public:
00015
00021     Device();
00022 };
00023
00024 #endif /* __ARDUINO_DRIVER_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
00005 EepromBasedWiredDevice::EepromBasedWiredDevice(unsigned char
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
char* buf, int len) {
00018     Wire.beginTransmission(getDeviceAddress());
00019     for (char i = addressSize; i > 0; i--) {
00020         Wire.write((unsigned char) (address >> ((i - 1) * 8)) & 0xff);
00021     }
00022     for (int i = 0; i < len; i++) {
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();
00037     delay(EEPROM_BASED_WIRED_DEVICE_AFTER_WRITE_DELAY);
00038     Wire.requestFrom((int) getDeviceAddress(), len);
00039     for (int i = 0; i < len; i++) {
00040         tries = MAX_RETRIES_ON_READING;
00041         while (!Wire.available() && --tries > 0) {
00042             delayMicroseconds(1);
00043         }
00044         if (tries == 0) {
00045             return;
00046         }
00047         buf[(endianness == BIG_ENDIAN) ? last - i : i] = Wire.read();
  
```

```

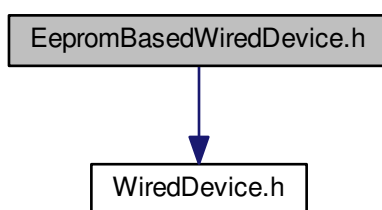
00048     }
00049 }
00050
00051 void EepromBasedWiredDevice::setAddressSize(char addressSize) {
00052     this->addressSize = addressSize;
00053 }
00054
00055 char EepromBasedWiredDevice::getAddressSize() {
00056     return addressSize;
00057 }

```

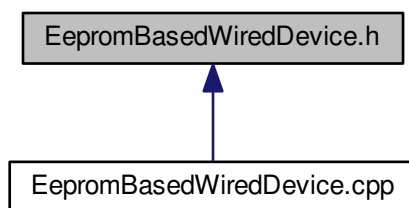
## 5.7 EepromBasedWiredDevice.h File Reference

```
#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](mailto: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 5
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 charendianness;
00022
00023 public:
00024
00025     enum {
00026         LITTLE_ENDIAN = 0x00,
00027         BIG_ENDIAN = 0x01
00028     } Endianness;
00029
00037     EepromBasedWiredDevice(unsigned char deviceAddress, char addressSize
, unsigned charendianness);
00038
00042     EepromBasedWiredDevice(unsigned char deviceAddress, char addressSize
);
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__ */

```

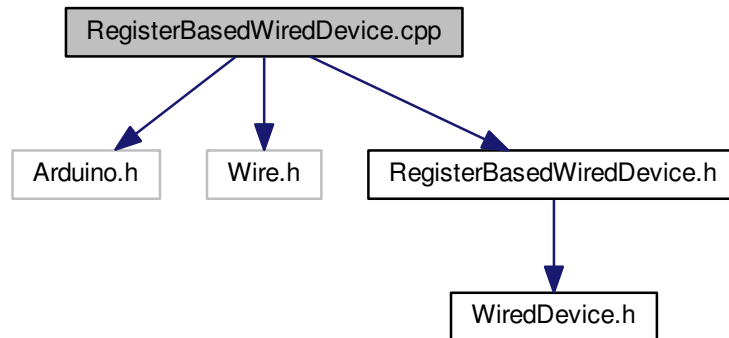
## 5.9 RegisterBasedWiredDevice.cpp File Reference

```

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

```

Include dependency graph for RegisterBasedWiredDevice.cpp:



## 5.10 RegisterBasedWiredDevice.cpp

```

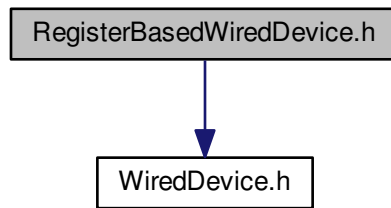
00001 #include <Arduino.h>
00002 #include <Wire.h>
00003 #include "RegisterBasedWiredDevice.h"
00004
00005 RegisterBasedWiredDevice::RegisterBasedWiredDevice(
00006     unsigned char address) :
00007     WiredDevice(address) {
00008 }
00009
00010 void RegisterBasedWiredDevice::configureRegisterBits(
00011     unsigned char reg,
00012     unsigned char mask, unsigned char d) {
00013     unsigned char n;
00014     n = readRegister(reg);
00015     n &= ~(mask);
00016     n |= d & mask;
00017     writeRegister(reg, n);
00018 }
00019 unsigned char RegisterBasedWiredDevice::writeRegister(unsigned char
00020     reg,
00021     unsigned char d) {
00022     Wire.beginTransmission(getDeviceAddress());
00023     Wire.write(reg);
00024     Wire.write(d);
00025     return Wire.endTransmission();
00026 }
00027 int RegisterBasedWiredDevice::readRegister(unsigned char reg) {
00028     char tries = MAX_RETRIES_ON_READING;
00029     Wire.beginTransmission(getDeviceAddress());
00030     Wire.write(reg);
00031     char status = Wire.endTransmission(false);
00032     if (status != 0) {
00033         return -(status);
00034     }
00035     Wire.requestFrom(getDeviceAddress(), (unsigned char) 1);
00036     while (!Wire.available() && --tries > 0) {
00037         delayMicroseconds(1);
00038     }
00039     if (tries == 0) {
00040         return -5;
00041     }
00042     return Wire.read();
00043 }
  
```

## 5.11 RegisterBasedWiredDevice.h File Reference

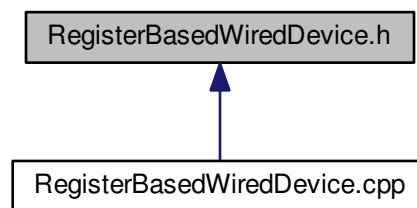
```
#include <WiredDevice.h>
```



Include dependency graph for RegisterBasedWiredDevice.h:



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



## Classes

- class [RegisterBasedWiredDevice](#)

## 5.12 RegisterBasedWiredDevice.h

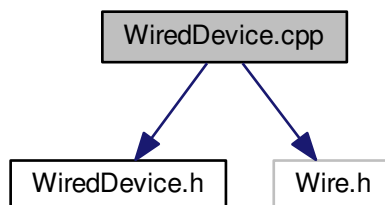
```

00001
00009 #ifndef __ARDUINO_DRIVER_REGISTER_BASED_WIRED_DEVICE_H__
00010 #define __ARDUINO_DRIVER_REGISTER_BASED_WIRED_DEVICE_H__ 1
00011
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
00034     void configureRegisterBits(unsigned char reg, unsigned char mask,
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:



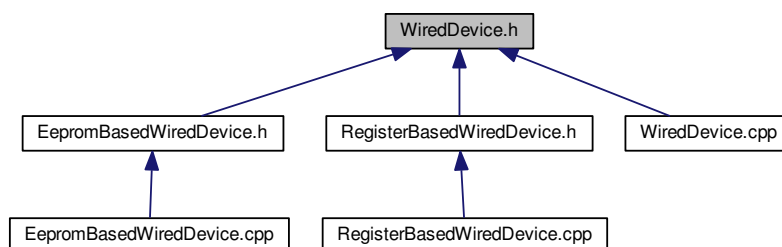
### 5.14 WiredDevice.cpp

```

00001 #include "WiredDevice.h"
00002 #include <Wire.h>
00003
00004 WiredDevice::WiredDevice(unsigned char deviceAddress)
00005     : deviceAddress(deviceAddress) {
00006     Wire.begin();
00007 }
00008
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

```
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
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
  - EepromBasedWiredDevice, [5](#)
- BIG\_ENDIAN
  - EepromBasedWiredDevice, [4](#)
- configureRegisterBits
  - RegisterBasedWiredDevice, [7](#)
- Device, [2](#)
  - Device, [2](#)
- Device.cpp, [9](#)
- Device.h, [10](#)
- deviceAddress
  - WiredDevice, [9](#)
- EEPROM\_BASED\_WIRED\_DEVICE\_AFTER\_WRITE\_DELAY
  - E\_DELAY
  - EepromBasedWiredDevice.h, [12](#)
- EepromBasedWiredDevice, [2](#)
  - addressSize, [5](#)
  - BIG\_ENDIAN, [4](#)
  - EepromBasedWiredDevice, [4](#)
  - Endianness, [5](#)
  - endianness, [5](#)
  - 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
  - EepromBasedWiredDevice, [5](#)
- endianness
  - EepromBasedWiredDevice, [5](#)
- getAddressSize
  - EepromBasedWiredDevice, [4](#)
- getDeviceAddress
  - WiredDevice, [9](#)
- LITTLE\_ENDIAN
  - EepromBasedWiredDevice, [4](#)
- MAX\_RETRIES\_ON\_READING
  - EepromBasedWiredDevice, [5](#)
  - RegisterBasedWiredDevice, [7](#)
- readBlock
  - EepromBasedWiredDevice, [4](#)
- readRegister
  - RegisterBasedWiredDevice, [7](#)
- RegisterBasedWiredDevice, [5](#)
  - configureRegisterBits, [7](#)
  - MAX\_RETRIES\_ON\_READING, [7](#)
  - readRegister, [7](#)
  - RegisterBasedWiredDevice, [7](#)
  - writeRegister, [7](#)
- RegisterBasedWiredDevice.cpp, [13](#), [14](#)
- RegisterBasedWiredDevice.h, [14](#), [15](#)
- setAddressSize
  - EepromBasedWiredDevice, [5](#)
- setDeviceAddress
  - WiredDevice, [9](#)
- WiredDevice, [8](#)
  - deviceAddress, [9](#)
  - getDeviceAddress, [9](#)
  - setDeviceAddress, [9](#)
  - WiredDevice, [8](#)
- WiredDevice.cpp, [16](#)
- WiredDevice.h, [16](#), [17](#)
- writeBlock
  - EepromBasedWiredDevice, [5](#)
- writeRegister
  - RegisterBasedWiredDevice, [7](#)