

Arduino Gyroscope Driver

Generated by Doxygen 1.8.9.1

Tue Aug 18 2015 22:51:58

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	1
2.1	Class List	2
3	File Index	2
3.1	File List	2
4	Class Documentation	2
4.1	External24cl256Eeprom Class Reference	2
4.1.1	Detailed Description	3
4.1.2	Constructor & Destructor Documentation	4
4.2	External24x16Eeprom Class Reference	4
4.2.1	Detailed Description	5
4.2.2	Constructor & Destructor Documentation	5
4.3	ExternalByteArrayEeprom Class Reference	6
4.3.1	Detailed Description	7
4.3.2	Constructor & Destructor Documentation	7
4.3.3	Member Function Documentation	7
4.3.4	Member Data Documentation	7
4.4	ExternalEeprom Class Reference	8
4.4.1	Detailed Description	9
4.4.2	Constructor & Destructor Documentation	9
4.4.3	Member Function Documentation	9
4.4.4	Member Data Documentation	11
4.5	ExternalMappedEeprom Class Reference	11
4.5.1	Detailed Description	12
4.5.2	Constructor & Destructor Documentation	13
4.5.3	Member Function Documentation	14
4.5.4	Member Data Documentation	14
5	File Documentation	15
5.1	External24cl256Eeprom.cpp File Reference	15
5.1.1	Macro Definition Documentation	15
5.2	External24cl256Eeprom.cpp	16
5.3	External24cl256Eeprom.h File Reference	16
5.4	External24cl256Eeprom.h	17
5.5	External24x16Eeprom.cpp File Reference	17
5.5.1	Macro Definition Documentation	18

5.6	External24x16Eeprom.cpp	18
5.7	External24x16Eeprom.h File Reference	19
5.8	External24x16Eeprom.h	19
5.9	ExternalByteArrayEeprom.cpp File Reference	20
5.9.1	Macro Definition Documentation	20
5.10	ExternalByteArrayEeprom.cpp	20
5.11	ExternalByteArrayEeprom.h File Reference	21
5.12	ExternalByteArrayEeprom.h	22
5.13	ExternalEeprom.cpp File Reference	22
5.13.1	Macro Definition Documentation	22
5.14	ExternalEeprom.cpp	23
5.15	ExternalEeprom.h File Reference	24
5.16	ExternalEeprom.h	25
5.17	ExternalFileEeprom.cpp File Reference	25
5.17.1	Macro Definition Documentation	25
5.18	ExternalFileEeprom.cpp	26
5.19	ExternalFileEeprom.h File Reference	26
5.20	ExternalFileEeprom.h	26
5.21	ExternalMappedEeprom.cpp File Reference	27
5.21.1	Macro Definition Documentation	27
5.22	ExternalMappedEeprom.cpp	27
5.23	ExternalMappedEeprom.h File Reference	28
5.24	ExternalMappedEeprom.h	29
	Index	31

1 Hierarchical Index

1.1 Class Hierarchy

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

EepromBasedWiredDevice

ExternalEeprom	8
External24cl256Eeprom	2
External24x16Eeprom	4
ExternalByteArrayEeprom	6
ExternalMappedEeprom	11

2 Class Index

2.1 Class List

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

External24cl256Eeprom	
Arduino - External 24cl256 eeprom	2
External24x16Eeprom	
Arduino - External 24x16 eeprom	4
ExternalByteArrayEeprom	
Arduino - External Virtual eeprom	6
ExternalEeprom	
Arduino - External eeprom	8
ExternalMappedEeprom	
Arduino - External eeprom	11

3 File Index

3.1 File List

Here is a list of all files with brief descriptions:

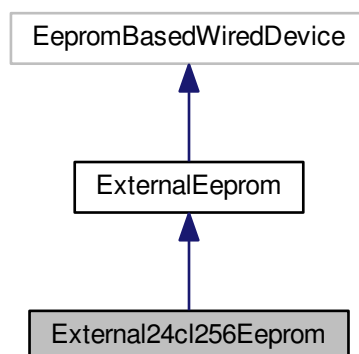
External24cl256Eeprom.cpp	15
External24cl256Eeprom.h	16
External24x16Eeprom.cpp	17
External24x16Eeprom.h	19
ExternalByteArrayEeprom.cpp	20
ExternalByteArrayEeprom.h	21
ExternalEeprom.cpp	22
ExternalEeprom.h	24
ExternalFileEeprom.cpp	25
ExternalFileEeprom.h	26
ExternalMappedEeprom.cpp	27
ExternalMappedEeprom.h	28

4 Class Documentation

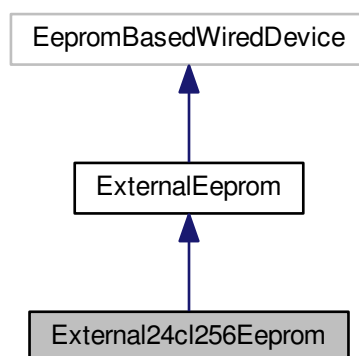
4.1 External24cl256Eeprom Class Reference

```
#include <External24cl256Eeprom.h>
```

Inheritance diagram for External24cl256Eeprom:



Collaboration diagram for External24cl256Eeprom:



Public Member Functions

- [External24cl256Eeprom](#) (unsigned char deviceAddress)

Additional Inherited Members

4.1.1 Detailed Description

Arduino - External 24cl256 eeprom.

[External24cl256Eeprom.h](#)

This an implementation of 24cl256 eeprom.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 16 of file [External24cl256Eeprom.h](#).

4.1.2 Constructor & Destructor Documentation**4.1.2.1 External24cl256Eeprom::External24cl256Eeprom (unsigned char *deviceAddress*)**

Public constructor.

Parameters

<i>deviceAddress</i>	The i2c addredd of the device.
----------------------	--------------------------------

Definition at line 18 of file [External24cl256Eeprom.cpp](#).

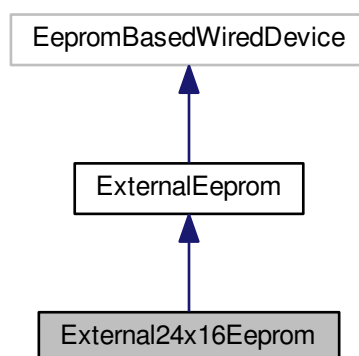
The documentation for this class was generated from the following files:

- [External24cl256Eeprom.h](#)
- [External24cl256Eeprom.cpp](#)

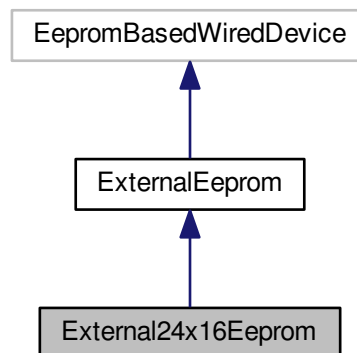
4.2 External24x16Eeprom Class Reference

```
#include <External24x16Eeprom.h>
```

Inheritance diagram for External24x16Eeprom:



Collaboration diagram for External24x16Eeprom:



Public Member Functions

- [External24x16Eeprom](#) (unsigned char deviceAddress)

Additional Inherited Members

4.2.1 Detailed Description

Arduino - External 24x16 eeprom.

[External24x16Eeprom.h](#)

This an implementation of 24X16 eeprom.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 17 of file [External24x16Eeprom.h](#).

4.2.2 Constructor & Destructor Documentation

4.2.2.1 External24x16Eeprom::External24x16Eeprom (unsigned char *deviceAddress*)

Public constructor.

Parameters

<i>device</i>	The i2c address of the device.
---------------	--------------------------------

Definition at line 16 of file [External24x16Eeprom.cpp](#).

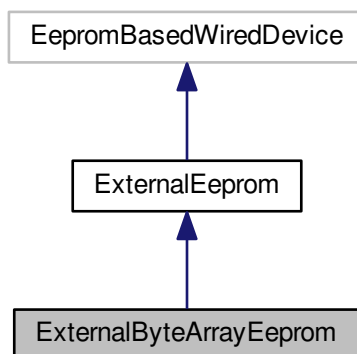
The documentation for this class was generated from the following files:

- [External24x16Eeprom.h](#)
- [External24x16Eeprom.cpp](#)

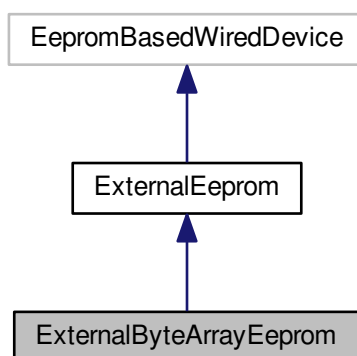
4.3 ExternalByteArrayEeprom Class Reference

```
#include <ExternalByteArrayEeprom.h>
```

Inheritance diagram for ExternalByteArrayEeprom:



Collaboration diagram for ExternalByteArrayEeprom:



Public Member Functions

- [ExternalByteArrayEeprom](#) (unsigned char *[byteArray](#), unsigned int [deviceSize](#))

Protected Member Functions

- virtual void [writeBlock](#) (unsigned int address, unsigned char *buf, int len)
- virtual void [readBlock](#) (unsigned int address, unsigned char *buf, int len)

Private Attributes

- unsigned char * [byteArray](#)

4.3.1 Detailed Description

Arduino - External Virtual eeprom.

[ExternalByteArrayEeprom.h](#)

This an implementation of VIRTUAL eeprom.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 16 of file [ExternalByteArrayEeprom.h](#).

4.3.2 Constructor & Destructor Documentation

4.3.2.1 ExternalByteArrayEeprom::ExternalByteArrayEeprom (unsigned char * *byteArray*, unsigned int *deviceSize*)

Public constructor.

Parameters

<i>device</i>	
---------------	--

Definition at line 16 of file [ExternalByteArrayEeprom.cpp](#).

4.3.3 Member Function Documentation

4.3.3.1 void ExternalByteArrayEeprom::readBlock (unsigned int *address*, unsigned char * *buf*, int *len*) [protected], [virtual]

Reads a block of bytes from the device.

Parameters

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

Definition at line 26 of file [ExternalByteArrayEeprom.cpp](#).

4.3.3.2 void ExternalByteArrayEeprom::writeBlock (unsigned int *address*, unsigned char * *buf*, int *len*) [protected], [virtual]

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

All bytes during a page write operation must reside on the same page.

Parameters

<i>address</i>	
<i>buffer</i>	
<i>len</i>	

Definition at line 20 of file [ExternalByteArrayEeprom.cpp](#).

4.3.4 Member Data Documentation

4.3.4.1 `unsigned char* ExternalByteArrayEeprom::byteArray` [private]

The used buffer.

Definition at line 21 of file [ExternalByteArrayEeprom.h](#).

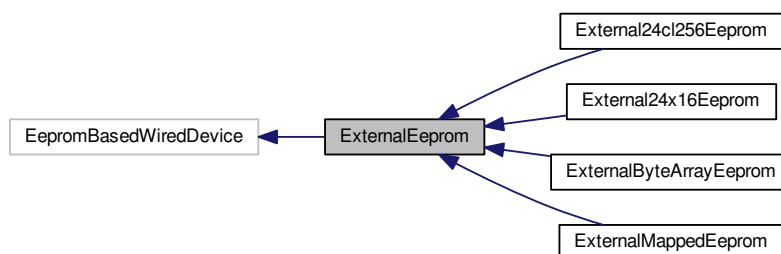
The documentation for this class was generated from the following files:

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

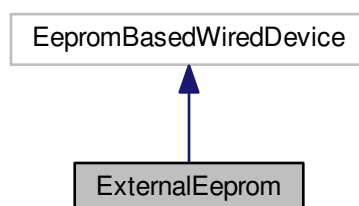
4.4 ExternalEeprom Class Reference

```
#include <ExternalEeprom.h>
```

Inheritance diagram for ExternalEeprom:



Collaboration diagram for ExternalEeprom:



Public Member Functions

- virtual void [write](#) (unsigned int address, unsigned char b)
- virtual void [writeBytes](#) (unsigned int address, unsigned char *buf, int len)
- virtual int [read](#) (unsigned int address)
- virtual int [readBytes](#) (unsigned int address, unsigned char *buf, int len)
- virtual int [setBytes](#) (unsigned int address, unsigned char b, int len)
- int [getPageSize](#) ()
- unsigned int [getDeviceSize](#) ()

Protected Member Functions

- [ExternalEeprom](#) (unsigned char deviceAddress, int [pageSize](#), unsigned int [deviceSize](#))
- unsigned int [endOfPage](#) (unsigned int address)

Private Attributes

- unsigned int [deviceSize](#)
- int [pageSize](#)

4.4.1 Detailed Description

Arduino - External eeprom.

[ExternalEeprom.h](#)

This is an abstract class of external eeprom.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 16 of file [ExternalEeprom.h](#).

4.4.2 Constructor & Destructor Documentation

4.4.2.1 `ExternalEeprom::ExternalEeprom (unsigned char deviceAddress, int pageSize, unsigned int deviceSize)` `[protected]`

Protected constructor.

Parameters

<i>device</i>	
---------------	--

Definition at line 18 of file [ExternalEeprom.cpp](#).

4.4.3 Member Function Documentation

4.4.3.1 `unsigned int ExternalEeprom::endOfPage (unsigned int address)` `[protected]`

Determines the length until first multiple of 'pageSize' of an address so writing always occurs up to 'pageSize' unsigned char boundaries.

Parameters

<i>address</i>	
----------------	--

Returns

Definition at line 110 of file [ExternalEeprom.cpp](#).

4.4.3.2 `unsigned int ExternalEeprom::getDeviceSize ()` `[inline]`

Gets the total size of the device.

Returns

Definition at line 86 of file [ExternalEeprom.h](#).

4.4.3.3 `int ExternalEeprom::getPageSize () [inline]`

Gets the page size of the device.

Returns

Definition at line 77 of file [ExternalEeprom.h](#).

4.4.3.4 `int ExternalEeprom::read (unsigned int address) [virtual]`

Reads a unsigned char from the device.

Parameters

<i>address</i>	
----------------	--

Returns

Definition at line 51 of file [ExternalEeprom.cpp](#).

4.4.3.5 `int ExternalEeprom::readBytes (unsigned int address, unsigned char * buf, int len) [virtual]`

Reads a buffer with len bytes from the device.

Parameters

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

Definition at line 59 of file [ExternalEeprom.cpp](#).

4.4.3.6 `int ExternalEeprom::setBytes (unsigned int address, unsigned char b, int len) [virtual]`

Writes len bytes at the address with data.

Parameters

<i>address</i>	
<i>data</i>	
<i>len</i>	

Definition at line 80 of file [ExternalEeprom.cpp](#).

4.4.3.7 `void ExternalEeprom::write (unsigned int address, unsigned char b) [virtual]`

Writes a unsigned char at the address into the device.

Parameters

<i>address</i>	The address where the data will be written.
<i>b</i>	The data to be written.

Definition at line 22 of file [ExternalEeprom.cpp](#).

4.4.3.8 `void ExternalEeprom::writeBytes (unsigned int address, unsigned char * buf, int len)` [virtual]

Writes a buffer of bytes at the address into the device.

Parameters

<i>address</i>	
<i>buffer</i>	
<i>len</i>	

Definition at line 26 of file [ExternalEeprom.cpp](#).

4.4.4 Member Data Documentation

4.4.4.1 `unsigned int ExternalEeprom::deviceSize` [private]

The size of the device.

Definition at line 21 of file [ExternalEeprom.h](#).

4.4.4.2 `int ExternalEeprom::pageSize` [private]

The size of the device page.

Definition at line 26 of file [ExternalEeprom.h](#).

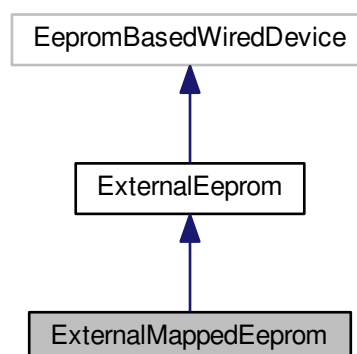
The documentation for this class was generated from the following files:

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

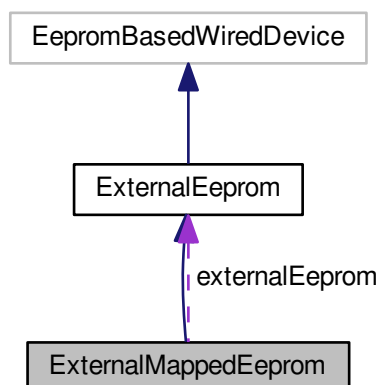
4.5 ExternalMappedEeprom Class Reference

```
#include <ExternalMappedEeprom.h>
```

Inheritance diagram for ExternalMappedEeprom:



Collaboration diagram for ExternalMappedEeprom:



Public Member Functions

- `ExternalMappedEeprom` (`ExternalEeprom *externalEeprom`, unsigned int `startAddress`, unsigned int `endAddress`)
- unsigned int `getDeviceSize` ()

Protected Member Functions

- virtual void `writeBlock` (unsigned int address, unsigned char *buf, int len)
- virtual void `readBlock` (unsigned int address, unsigned char *buf, int len)

Private Attributes

- `ExternalEeprom * externalEeprom`
- unsigned int `startAddress`
- unsigned int `endAddress`

4.5.1 Detailed Description

Arduino - External eeprom.

[ExternalMappedEeprom.h](#)

This is an abstract class of external eeprom.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 16 of file [ExternalMappedEeprom.h](#).

4.5.2 Constructor & Destructor Documentation

4.5.2.1 ExternalMappedEeprom::ExternalMappedEeprom (ExternalEeprom * *externalEeprom*, unsigned int *startAddress*, unsigned int *endAddress*)

Public constructor.

Parameters

<i>externalEeprom</i>	
<i>startAddress</i>	
<i>endAddress</i>	

Definition at line 16 of file [ExternalMappedEeprom.cpp](#).

4.5.3 Member Function Documentation**4.5.3.1 unsigned int ExternalMappedEeprom::getDeviceSize () [inline]**

Device size.

Returns

Definition at line 48 of file [ExternalMappedEeprom.h](#).

4.5.3.2 void ExternalMappedEeprom::readBlock (unsigned int *address*, unsigned char * *buf*, int *len*) [protected], [virtual]

Reads a block of bytes from the device.

Parameters

<i>address</i>	
<i>buffer</i>	
<i>len</i>	

Definition at line 31 of file [ExternalMappedEeprom.cpp](#).

4.5.3.3 void ExternalMappedEeprom::writeBlock (unsigned int *address*, unsigned char * *buf*, int *len*) [protected], [virtual]

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

All bytes during a page write operation must reside on the same page.

Parameters

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

Definition at line 22 of file [ExternalMappedEeprom.cpp](#).

4.5.4 Member Data Documentation**4.5.4.1 unsigned int ExternalMappedEeprom::endAddress [private]**

The end address of the map.

Definition at line 31 of file [ExternalMappedEeprom.h](#).

4.5.4.2 ExternalEeprom* ExternalMappedEeprom::externalEeprom [private]

The underlying external eeprom.

Definition at line 21 of file [ExternalMappedEeprom.h](#).

4.5.4.3 unsigned int ExternalMappedEeprom::startAddress [private]

The start address of the map.

Definition at line 26 of file [ExternalMappedEeprom.h](#).

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

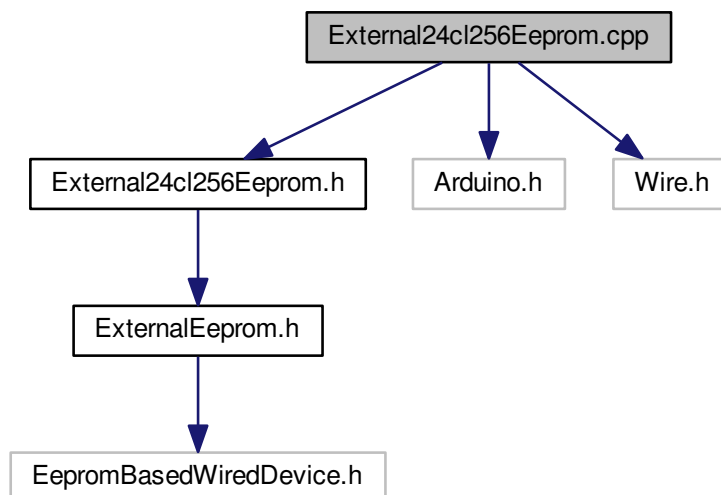
- [ExternalMappedEeprom.h](#)
- [ExternalMappedEeprom.cpp](#)

5 File Documentation

5.1 External24cl256Eeprom.cpp File Reference

```
#include "External24cl256Eeprom.h"
#include <Arduino.h>
#include <Wire.h>
```

Include dependency graph for External24cl256Eeprom.cpp:



Macros

- `#define __ARDUINO_EXTERNAL_24CL256_EEPROM_CPP__ 1`

5.1.1 Macro Definition Documentation

5.1.1.1 `#define __ARDUINO_EXTERNAL_24CL256_EEPROM_CPP__ 1`

Arduino - External 24cl256 eeprom.

[External24cl256Eeprom.cpp](#)

This an implementation of 24cl256 eeprom.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 12 of file [External24cl256Eeprom.cpp](#).

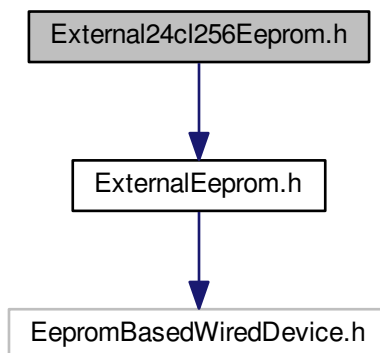
5.2 External24cl256Eeprom.cpp

```
00001
00011 #ifndef __ARDUINO_EXTERNAL_24CL256_EEPROM_CPP__
00012 #define __ARDUINO_EXTERNAL_24CL256_EEPROM_CPP__ 1
00013
00014 #include "External24cl256Eeprom.h"
00015 #include <Arduino.h>
00016 #include <Wire.h>
00017
00018 External24cl256Eeprom::External24cl256Eeprom(unsigned char
    deviceAddress) :
00019     ExternalEeprom(deviceAddress, 0x40, 0x7fff) {
00020     setAddressSize(0x02);
00021 }
00022
00023 #endif /* __ARDUINO_EXTERNAL_24CL256_EEPROM_CPP__ */
```

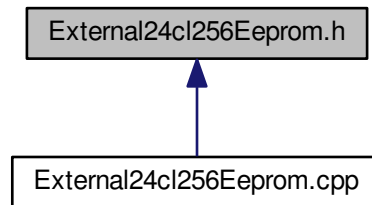
5.3 External24cl256Eeprom.h File Reference

```
#include <ExternalEeprom.h>
```

Include dependency graph for External24cl256Eeprom.h:



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



Classes

- class [External24cl256Eeprom](#)

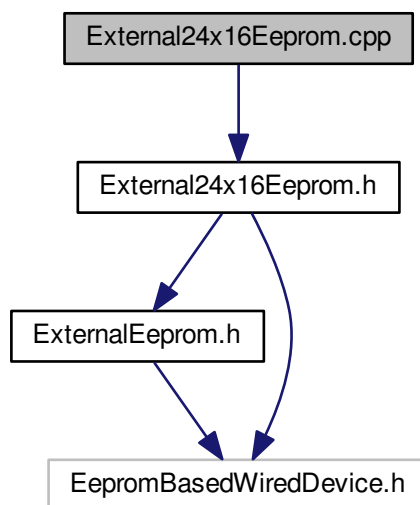
5.4 External24cl256Eeprom.h

```
00001
00011 #ifndef __ARDUINO_EXTERNAL_24CL256_EEPROM_H__
00012 #define __ARDUINO_EXTERNAL_24CL256_EEPROM_H__ 1
00013
00014 #include <ExternalEeprom.h>
00015
00016 class External24cl256Eeprom : public ExternalEeprom {
00017 public:
00018
00024     External24cl256Eeprom(unsigned char deviceAddress);
00025 };
00026
00027 #endif /* __ARDUINO_EXTERNAL_24CL256_EEPROM_H__ */
```

5.5 External24x16Eeprom.cpp File Reference

```
#include "External24x16Eeprom.h"
```

Include dependency graph for External24x16Eeprom.cpp:



Macros

- `#define __ARDUINO_EXTERNAL_24X16_EEPROM_CPP__ 1`

5.5.1 Macro Definition Documentation

5.5.1.1 `#define __ARDUINO_EXTERNAL_24X16_EEPROM_CPP__ 1`

Arduino - External 24x16 eeprom.

[External24x16Eeprom.cpp](#)

This an implementation of 24x16 eeprom.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 12 of file [External24x16Eeprom.cpp](#).

5.6 External24x16Eeprom.cpp

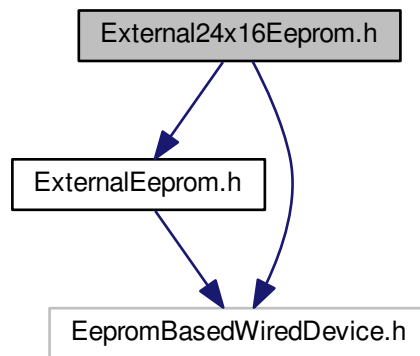
```

00001
00011 #ifndef __ARDUINO_EXTERNAL_24X16_EEPROM_CPP__
00012 #define __ARDUINO_EXTERNAL_24X16_EEPROM_CPP__ 1
00013
00014 #include "External24x16Eeprom.h"
00015
00016 External24x16Eeprom::External24x16Eeprom(unsigned char
    deviceAddress) :
00017     ExternalEeprom(0x20, 0x7ff, deviceAddress) {
00018     setAddressSize(0x01);
00019 }
00020
00021 #endif /* __ARDUINO_EXTERNAL_24X16_EEPROM_CPP__ */
  
```

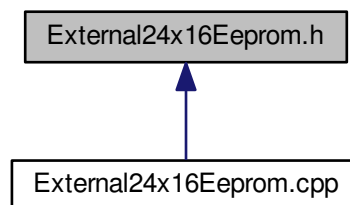
5.7 External24x16Eeprom.h File Reference

```
#include <ExternalEeprom.h>
#include <EepromBasedWiredDevice.h>
```

Include dependency graph for External24x16Eeprom.h:



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



Classes

- class [External24x16Eeprom](#)

5.8 External24x16Eeprom.h

```
00001
00011 #ifndef __ARDUINO_EXTERNAL_24X16_EEPROM_H__
00012 #define __ARDUINO_EXTERNAL_24X16_EEPROM_H__ 1
00013
00014 #include <ExternalEeprom.h>
00015 #include <EepromBasedWiredDevice.h>
00016
00017 class External24x16Eeprom : public ExternalEeprom {
00018 public:
00019
00025     External24x16Eeprom(unsigned char deviceAddress);
```

```

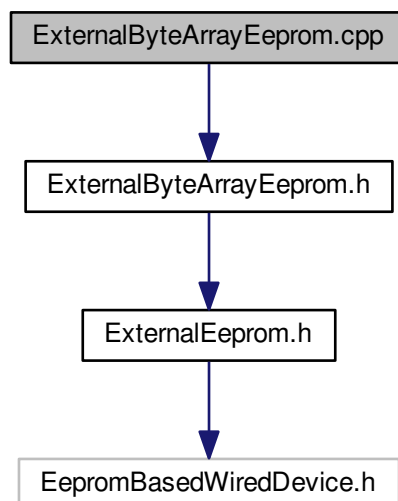
00026 };
00027
00028 #endif /* __ARDUINO_EXTERNAL_24X16_EEPROM_H__ */

```

5.9 ExternalByteArrayEeprom.cpp File Reference

```
#include "ExternalByteArrayEeprom.h"
```

Include dependency graph for ExternalByteArrayEeprom.cpp:



Macros

- `#define __ARDUINO_EXTERNAL_BYTE_ARRAY_EEPROM_CPP__ 1`

5.9.1 Macro Definition Documentation

5.9.1.1 `#define __ARDUINO_EXTERNAL_BYTE_ARRAY_EEPROM_CPP__ 1`

Arduino - External Virtual eeprom.

[ExternalByteArrayEeprom.cpp](#)

This an implementation of Virtual eeprom.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 12 of file [ExternalByteArrayEeprom.cpp](#).

5.10 ExternalByteArrayEeprom.cpp

```

00001
00011 #ifndef __ARDUINO_EXTERNAL_BYTE_ARRAY_EEPROM_CPP__
00012 #define __ARDUINO_EXTERNAL_BYTE_ARRAY_EEPROM_CPP__ 1

```

```

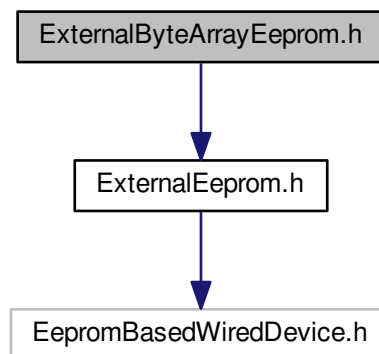
00013
00014 #include "ExternalByteArrayEeprom.h"
00015
00016 ExternalByteArrayEeprom::ExternalByteArrayEeprom(unsigned
    char* byteArray, unsigned int deviceSize)
00017     : ExternalEeprom(0, 16, deviceSize), byteArray(byteArray) {
00018 }
00019
00020 void ExternalByteArrayEeprom::writeBlock(unsigned int address, unsigned
    char* buf, int len) {
00021     for (int i = 0; i < len; i++) {
00022         byteArray[address + i] = buf[i];
00023     }
00024 }
00025
00026 void ExternalByteArrayEeprom::readBlock(unsigned int address, unsigned
    char* buf, int len) {
00027     for (int i = 0; i < len; i++) {
00028         buf[i] = byteArray[address + i];
00029     }
00030 }
00031
00032 #endif /* __ARDUINO_EXTERNAL_BYTE_ARRAY_EEPROM_CPP__ */

```

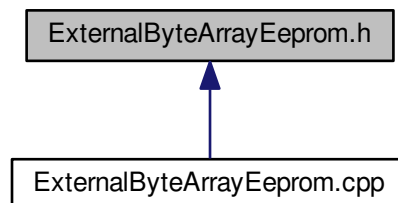
5.11 ExternalByteArrayEeprom.h File Reference

#include <ExternalEeprom.h>

Include dependency graph for ExternalByteArrayEeprom.h:



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



Classes

- class [ExternalByteArrayEeprom](#)

5.12 ExternalByteArrayEeprom.h

```

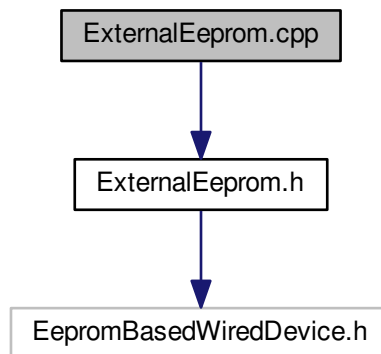
00001
00011 #ifndef __ARDUINO_EXTERNAL_BYTE_ARRAY_EEPROM_H__
00012 #define __ARDUINO_EXTERNAL_BYTE_ARRAY_EEPROM_H__ 1
00013
00014 #include <ExternalEeprom.h>
00015
00016 class ExternalByteArrayEeprom : public ExternalEeprom {
00017
00021     unsigned char* byteArray;
00022
00023 public:
00024
00030     ExternalByteArrayEeprom(unsigned char* byteArray, unsigned int
        deviceSize);
00031
00032 protected:
00033
00042     virtual void writeBlock(unsigned int address, unsigned char* buf, int len);
00043
00051     virtual void readBlock(unsigned int address, unsigned char* buf, int len);
00052 };
00053
00054 #endif /* __ARDUINO_EXTERNAL_BYTE_ARRAY_EEPROM_H__ */

```

5.13 ExternalEeprom.cpp File Reference

#include "ExternalEeprom.h"

Include dependency graph for ExternalEeprom.cpp:



Macros

- #define [__ARDUINO_EXTERNAL_EEPROM_CPP__](#) 1
- #define [min](#)(a, b) ((a > b) ? b : a)

5.13.1 Macro Definition Documentation

5.13.1.1 #define __ARDUINO_EXTERNAL_EEPROM_CPP__ 1

Arduino - External eeprom.

[ExternalEeprom.cpp](#)

This is an abstract class of external eeprom.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 12 of file [ExternalEeprom.cpp](#).

5.13.1.2 #define min(a, b) ((a > b) ? b : a)

Definition at line 14 of file [ExternalEeprom.cpp](#).

5.14 ExternalEeprom.cpp

```

00001
00011 #ifndef __ARDUINO_EXTERNAL_EEPROM_CPP__
00012 #define __ARDUINO_EXTERNAL_EEPROM_CPP__ 1
00013
00014 #define min(a, b) ((a > b) ? b : a)
00015
00016 #include "ExternalEeprom.h"
00017
00018 ExternalEeprom::ExternalEeprom(unsigned char deviceAddress, int pageSize,
    unsigned int deviceSize)
00019     : EepromBasedWiredDevice(0x50 | (deviceAddress & 0x07)), deviceSize(deviceSize), pageSize(pageSize)
    {
00020 }
00021
00022 void ExternalEeprom::write(unsigned int address, unsigned char b) {
00023     writeBlock(address, &b, 1);
00024 }
00025
00026 void ExternalEeprom::writeBytes(unsigned int address, unsigned char* buf, int len
    ) {
00027     unsigned int eop, room;
00028     int chunkSize;
00029     room = (deviceSize - address);
00030     if (room == 0) {
00031         return;
00032     }
00033     len = (room < (unsigned char) len) ? room : len;
00034     eop = endOfPage(address);
00035     chunkSize = min(eop, (unsigned char) len);
00036     if (chunkSize > 0) {
00037         writeBlock(address, buf, chunkSize);
00038         address += chunkSize;
00039         buf += chunkSize;
00040         len -= chunkSize;
00041     }
00042     while (len > 0) {
00043         chunkSize = min(len, pageSize);
00044         writeBlock(address, buf, chunkSize);
00045         address += chunkSize;
00046         buf += chunkSize;
00047         len -= chunkSize;
00048     }
00049 }
00050
00051 int ExternalEeprom::read(unsigned int address) {
00052     unsigned char b;
00053     if (readBytes(address, &b, 1) == -1) {
00054         return -1;
00055     }
00056     return (int) b;
00057 }
00058
00059 int ExternalEeprom::readBytes(unsigned int address, unsigned char* buf, int len) {
00060     int cnt, chunkSize = pageSize;
00061     unsigned int available;
00062     if (address >= deviceSize) {
00063         return -1;
00064     }
00065     available = (deviceSize - address);

```

```

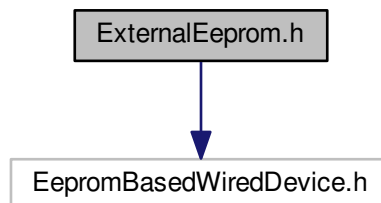
00066     if (available < (unsigned char) len) {
00067         len = (int) available;
00068     }
00069     cnt = len;
00070     while (len > 0) {
00071         chunkSize = min(len, pageSize);
00072         readBlock(address, buf, chunkSize);
00073         address += chunkSize;
00074         buf += chunkSize;
00075         len -= chunkSize;
00076     }
00077     return cnt;
00078 }
00079
00080 int ExternalEeprom::setBytes(unsigned int address, unsigned char b, int len) {
00081     unsigned char buf[pageSize];
00082     int eop, chunkSize;
00083     unsigned int room;
00084     if (address >= deviceSize) {
00085         return -1;
00086     }
00087     room = (deviceSize - address);
00088     if (room < (unsigned char) len) {
00089         len = (int) room;
00090     }
00091     for (int i = 0; i < pageSize; i++) {
00092         buf[i] = b;
00093     }
00094     eop = endOfPage(address);
00095     if (eop > 0) {
00096         chunkSize = min(eop, len);
00097         writeBlock(address, buf, chunkSize);
00098         address += chunkSize;
00099         len -= chunkSize;
00100     }
00101     while (len > 0) {
00102         chunkSize = min(len, pageSize);
00103         writeBlock(address, buf, chunkSize);
00104         address += chunkSize;
00105         len -= chunkSize;
00106     }
00107     return len;
00108 }
00109
00110 unsigned int ExternalEeprom::endOfPage(unsigned int address) {
00111     // Why / and then * by the same number?
00112     unsigned int eopAddr = ((address + pageSize - 1) / pageSize) *
        pageSize;
00113     return (eopAddr - address);
00114 }
00115
00116 #endif /* __ARDUINO_EXTERNAL_EEPROM_CPP__ */

```

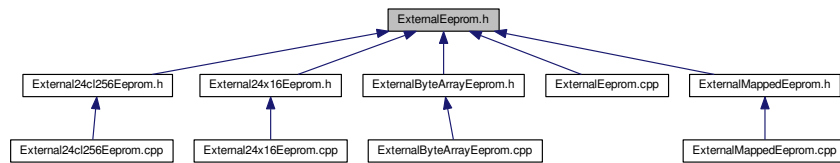
5.15 ExternalEeprom.h File Reference

#include <EepromBasedWiredDevice.h>

Include dependency graph for ExternalEeprom.h:



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



Classes

- class [ExternalEeprom](#)

5.16 ExternalEeprom.h

```

00001
00011 #ifndef __ARDUINO_EXTERNAL_EEPROM_H__
00012 #define __ARDUINO_EXTERNAL_EEPROM_H__ 1
00013
00014 #include <EepromBasedWiredDevice.h>
00015
00016 class ExternalEeprom : public EepromBasedWiredDevice {
00017     unsigned int deviceSize;
00021
00022     int pageSize;
00026 public:
00027
00028     virtual void write(unsigned int address, unsigned char b);
00035
00036     virtual void writeBytes(unsigned int address, unsigned char* buf, int len);
00044
00045     virtual int read(unsigned int address);
00052
00053     virtual int readBytes(unsigned int address, unsigned char* buf, int len);
00061
00062     virtual int setBytes(unsigned int address, unsigned char b, int len);
00070
00071     int getPageSize() {
00077         return pageSize;
00078     }
00079
00080     unsigned int getDeviceSize() {
00086         return deviceSize;
00087     }
00088
00089 protected:
00090
00091     ExternalEeprom(unsigned char deviceAddress, int pageSize, unsigned int deviceSize);
00097
00098     unsigned int endOfPage(unsigned int address);
00106 };
00107
00108
00109 #endif /* __ARDUINO_EXTERNAL_EEPROM_H__ */
  
```

5.17 ExternalFileEeprom.cpp File Reference

Macros

- #define [__ARDUINO_EXTERNAL_FILE_EEPROM_CPP__](#) 1

5.17.1 Macro Definition Documentation

5.17.1.1 #define __ARDUINO_EXTERNAL_FILE_EEPROM_CPP__ 1

Arduino - External Virtual eeprom.

ExternalFileEeprom.cpp

This an implementation of Virtual eeprom.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 12 of file [ExternalFileEeprom.cpp](#).

5.18 ExternalFileEeprom.cpp

```

00001
00011 #ifndef __ARDUINO_EXTERNAL_FILE_EEPROM_CPP__
00012 #define __ARDUINO_EXTERNAL_FILE_EEPROM_CPP__ 1
00013
00014 #if USE_FILE_LIBRARIES
00015
00016 #include "ExternalFileEeprom.h"
00017 #include <stdint.h>
00018 #include <stdlib.h>
00019 #include <stddef.h>
00020
00021 ExternalFileEeprom::ExternalFileEeprom(char *fileName, unsigned int deviceSize) :
00022     ExternalEeprom(0, 16, deviceSize), fileName(fileName) {
00023     fp = fopen(fileName, "rb+");
00024     if (fp == NULL) {
00025         printf("Error when opening file: %s.\n", fileName);
00026         exit(1);
00027     }
00028 }
00029 void ExternalFileEeprom::writeBlock(unsigned int address, unsigned char* buffer, int len) {
00030     fseek(fp, address, 0);
00031     fwrite(buffer, sizeof(unsigned char), len, fp);
00032 }
00033
00034 void ExternalFileEeprom::readBlock(unsigned int address, unsigned char* buffer, int len) {
00035     fseek(fp, address, 0);
00036     fread(buffer, sizeof(unsigned char), len, fp);
00037 }
00038
00039 #endif /* USE_FILE_LIBRARIES */
00040
00041 #endif /* __ARDUINO_EXTERNAL_FILE_EEPROM_CPP__ */

```

5.19 ExternalFileEeprom.h File Reference

5.20 ExternalFileEeprom.h

```

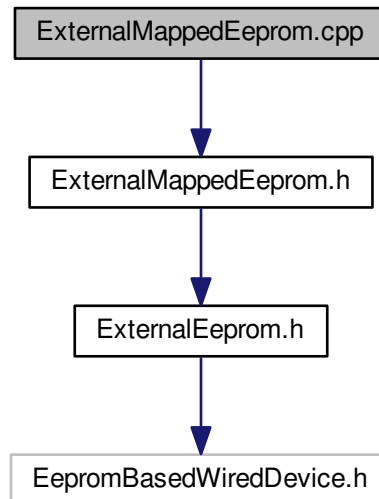
00001
00011 #ifndef __ARDUINO_EXTERNAL_FILE_EEPROM_H__
00012 #define __ARDUINO_EXTERNAL_FILE_EEPROM_H__ 1
00013
00014 #if USE_FILE_LIBRARIES
00015
00016 #include <ExternalEeprom.h>
00017
00018 class ExternalFileEeprom : public ExternalEeprom {
00019 private:
00020     char *fileName;
00021     FILE *fp;
00022 public:
00023
00029     ExternalFileEeprom(char *fileName, unsigned int size);
00030
00031 protected:
00032
00041     virtual void writeBlock(unsigned int address, unsigned char* buf, int len);
00042
00050     virtual void readBlock(unsigned int address, unsigned char* buf, int len);
00051 };
00052
00053 #endif /* USE_FILE_LIBRARIES */
00054
00055 #endif /* __ARDUINO_EXTERNAL_FILE_EEPROM_H__ */

```

5.21 ExternalMappedEeprom.cpp File Reference

```
#include "ExternalMappedEeprom.h"
```

Include dependency graph for ExternalMappedEeprom.cpp:



Macros

- `#define __ARDUINO_EXTERNAL_MAPPED_EEPROM_CPP__ 1`

5.21.1 Macro Definition Documentation

5.21.1.1 `#define __ARDUINO_EXTERNAL_MAPPED_EEPROM_CPP__ 1`

Arduino - External eeprom.

[ExternalMappedEeprom.cpp](#)

This is an abstract class of external eeprom.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 12 of file [ExternalMappedEeprom.cpp](#).

5.22 ExternalMappedEeprom.cpp

```

00001
00011 #ifndef __ARDUINO_EXTERNAL_MAPPED_EEPROM_CPP__
00012 #define __ARDUINO_EXTERNAL_MAPPED_EEPROM_CPP__ 1
00013
00014 #include "ExternalMappedEeprom.h"
00015
00016 ExternalMappedEeprom::ExternalMappedEeprom(
00017     ExternalEeprom* externalEeprom, unsigned int startAddress, unsigned int endAddress)
00017     : ExternalEeprom(externalEeprom->getDeviceAddress(), externalEeprom->getPageSize(),

```

```

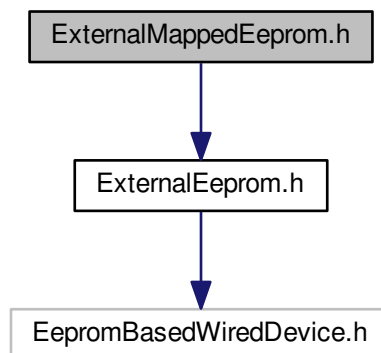
        externalEeprom->getDeviceSize()), externalEeprom(externalEeprom) {
00018     this->startAddress = startAddress;
00019     this->endAddress = endAddress;
00020 }
00021
00022 void ExternalMappedEeprom::writeBlock(unsigned int address, unsigned char*
    buf, int len) {
00023     unsigned int mappedAddress = (address + startAddress);
00024     if (mappedAddress < endAddress) {
00025         unsigned int available = (endAddress - mappedAddress);
00026         len = (len > available) ? available : len;
00027         externalEeprom->writeBlock(mappedAddress, buf, len);
00028     }
00029 }
00030
00031 void ExternalMappedEeprom::readBlock(unsigned int address, unsigned char*
    buf, int len) {
00032     unsigned int mappedAddress = (address + startAddress);
00033     if (mappedAddress < endAddress) {
00034         unsigned int available = (endAddress - mappedAddress);
00035         len = (len > available) ? available : len;
00036         externalEeprom->readBlock(mappedAddress, buf, len);
00037     }
00038 }
00039
00040 #endif /* __ARDUINO_EXTERNAL_MAPPED_EEPROM_CPP__ */

```

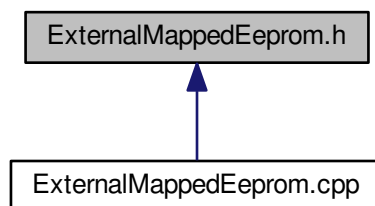
5.23 ExternalMappedEeprom.h File Reference

#include <ExternalEeprom.h>

Include dependency graph for ExternalMappedEeprom.h:



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



Classes

- class [ExternalMappedEeprom](#)

5.24 ExternalMappedEeprom.h

```

00001
00011 #ifndef __ARDUINO_EXTERNAL_MAPPED_EEPROM_H__
00012 #define __ARDUINO_EXTERNAL_MAPPED_EEPROM_H__ 1
00013
00014 #include <ExternalEeprom.h>
00015
00016 class ExternalMappedEeprom: public ExternalEeprom {
00017
00021     ExternalEeprom* externalEeprom;
00022
00026     unsigned int startAddress;
00027
00031     unsigned int endAddress;
00032 public:
00033
00041     ExternalMappedEeprom(ExternalEeprom* externalEeprom, unsigned int
startAddress, unsigned int endAddress);
00042
00048     unsigned int getDeviceSize() {
00049         return (endAddress - startAddress);
00050     }
00051
00052 protected:
00053
00062     virtual void writeBlock(unsigned int address, unsigned char* buf, int len);
00063
00071     virtual void readBlock(unsigned int address, unsigned char* buf, int len);
00072 };
00073
00074 #endif /* __ARDUINO_EXTERNAL_MAPPED_EEPROM_H__ */
  
```


Index

- `__ARDUINO_EXTERNAL_24CL256_EEPROM_CPP`↔
 - `External24cl256Eeprom.cpp`, 15
- `__ARDUINO_EXTERNAL_24X16_EEPROM_CPP`
 - `External24x16Eeprom.cpp`, 18
- `__ARDUINO_EXTERNAL_BYTE_ARRAY_EEPROM`↔
 - `__CPP`
 - `ExternalByteArrayEeprom.cpp`, 20
- `__ARDUINO_EXTERNAL_EEPROM_CPP`
 - `ExternalEeprom.cpp`, 22
- `__ARDUINO_EXTERNAL_FILE_EEPROM_CPP`
 - `ExternalFileEeprom.cpp`, 25
- `__ARDUINO_EXTERNAL_MAPPED_EEPROM_CP`↔
 - `P`
 - `ExternalMappedEeprom.cpp`, 27
- `byteArray`
 - `ExternalByteArrayEeprom`, 7
- `deviceSize`
 - `ExternalEeprom`, 11
- `endAddress`
 - `ExternalMappedEeprom`, 14
- `endOfPage`
 - `ExternalEeprom`, 9
- `External24cl256Eeprom`, 2
 - `External24cl256Eeprom`, 4
- `External24cl256Eeprom.cpp`, 15, 16
 - `__ARDUINO_EXTERNAL_24CL256_EEPROM`↔
 - `__CPP`, 15
- `External24cl256Eeprom.h`, 16, 17
- `External24x16Eeprom`, 4
 - `External24x16Eeprom`, 5
- `External24x16Eeprom.cpp`, 17, 18
 - `__ARDUINO_EXTERNAL_24X16_EEPROM_C`↔
 - `PP`, 18
- `External24x16Eeprom.h`, 19
- `ExternalByteArrayEeprom`, 6
 - `byteArray`, 7
 - `ExternalByteArrayEeprom`, 7
 - `readBlock`, 7
 - `writeBlock`, 7
- `ExternalByteArrayEeprom.cpp`, 20
 - `__ARDUINO_EXTERNAL_BYTE_ARRAY_EEP`↔
 - `ROM_CPP`, 20
- `ExternalByteArrayEeprom.h`, 21, 22
- `ExternalEeprom`, 8
 - `deviceSize`, 11
 - `endOfPage`, 9
 - `ExternalEeprom`, 9
 - `getDeviceSize`, 9
 - `getPageSize`, 10
 - `pageSize`, 11
 - `read`, 10
 - `readBytes`, 10
- `setBytes`, 10
 - `write`, 10
 - `writeBytes`, 11
- `externalEeprom`
 - `ExternalMappedEeprom`, 14
- `ExternalEeprom.cpp`, 22, 23
 - `__ARDUINO_EXTERNAL_EEPROM_CPP`, 22
 - `min`, 23
- `ExternalEeprom.h`, 24, 25
- `ExternalFileEeprom.cpp`, 25, 26
 - `__ARDUINO_EXTERNAL_FILE_EEPROM_CP`↔
 - `P`, 25
- `ExternalFileEeprom.h`, 26
- `ExternalMappedEeprom`, 11
 - `endAddress`, 14
 - `externalEeprom`, 14
 - `ExternalMappedEeprom`, 13
 - `getDeviceSize`, 14
 - `readBlock`, 14
 - `startAddress`, 14
 - `writeBlock`, 14
- `ExternalMappedEeprom.cpp`, 27
 - `__ARDUINO_EXTERNAL_MAPPED_EEPROM`↔
 - `__CPP`, 27
- `ExternalMappedEeprom.h`, 28, 29
- `getDeviceSize`
 - `ExternalEeprom`, 9
 - `ExternalMappedEeprom`, 14
- `getPageSize`
 - `ExternalEeprom`, 10
- `min`
 - `ExternalEeprom.cpp`, 23
- `pageSize`
 - `ExternalEeprom`, 11
- `read`
 - `ExternalEeprom`, 10
- `readBlock`
 - `ExternalByteArrayEeprom`, 7
 - `ExternalMappedEeprom`, 14
- `readBytes`
 - `ExternalEeprom`, 10
- `setBytes`
 - `ExternalEeprom`, 10
- `startAddress`
 - `ExternalMappedEeprom`, 14
- `write`
 - `ExternalEeprom`, 10
- `writeBlock`
 - `ExternalByteArrayEeprom`, 7
 - `ExternalMappedEeprom`, 14
- `writeBytes`
 - `ExternalEeprom`, 11