

Zeeprom Library

Generated by Doxygen 1.8.14

Contents

1	zEeprom	1
2	Data Structure Index	1
2.1	Data Structures	1
3	File Index	2
3.1	File List	2
4	Data Structure Documentation	2
4.1	ZEeprom Class Reference	2
4.1.1	Detailed Description	3
4.1.2	Constructor & Destructor Documentation	3
4.1.3	Member Function Documentation	3
5	File Documentation	7
5.1	README.md File Reference	7
5.2	ZEeprom.cpp File Reference	7
5.2.1	Detailed Description	8
5.2.2	Macro Definition Documentation	8
5.3	ZEeprom.h File Reference	10
5.3.1	Macro Definition Documentation	11
	Index	15

1 zEeprom

Arduino library for 24C01/02/04/08/16/32/... serial EEPROM

2 Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

[ZEeprom](#)

EEPROM 24C01 / 24C02 memory driver

[2](#)

3 File Index

3.1 File List

Here is a list of all files with brief descriptions:

[ZEeprom.cpp](#)

EEPROM 24C01 /02/04/08/16/32/64/128/254/512/1024 library for Arduino - Demonstration program

[7](#)[ZEeprom.h](#)[10](#)

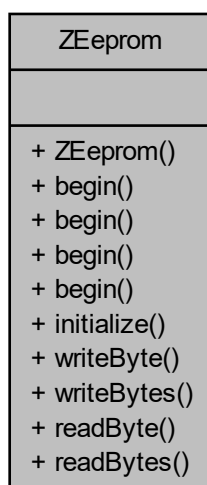
4 Data Structure Documentation

4.1 ZEeprom Class Reference

EEPROM 24C01 / 24C02 memory driver.

```
#include <ZEeprom.h>
```

Collaboration diagram for ZEeprom:



Public Member Functions

- [ZEeprom](#) ()
Constructor.
- void [begin](#) (TwoWire &MyWire, uint8_t addr, unsigned int memorytype)
Setups the I2C interface and hardware.
- void [begin](#) (TwoWire &MyWire, uint8_t addr)
- void [begin](#) (uint8_t addr)
Setups the I2C interface and hardware.
- void [begin](#) (void)
Setups the I2C interface and hardware.
- void [initialize](#) ()
Initialize library and TWI bus.
- void [writeByte](#) (unsigned int address, byte data)
Write a byte in EEPROM memory.
- void [writeBytes](#) (unsigned int address, unsigned int length, byte *p_data)
Write bytes in EEPROM memory.
- byte [readByte](#) (unsigned int address)
Read a byte in EEPROM memory.
- void [readBytes](#) (unsigned int address, unsigned int length, byte *p_buffer)
Read bytes in EEPROM memory.

4.1.1 Detailed Description

EEPROM 24C01 / 24C02 memory driver.

This driver is designed for 24C01 and 24C02 EEPROM memories.

Definition at line 72 of file ZEeprom.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 ZEeprom()

```
ZEeprom::ZEeprom ( )
```

Constructor.

Parameters

<i>deviceAddress</i>	EEPROM address on TWI bus.
----------------------	----------------------------

Definition at line 90 of file ZEeprom.cpp.

4.1.3 Member Function Documentation

4.1.3.1 `begin()` [1/4]

```
void ZEEPROM::begin (
    TwoWire & MyWire,
    uint8_t addr,
    unsigned int memorytype )
```

Setups the I2C interface and hardware.

Definition at line 373 of file ZEEPROM.cpp.

4.1.3.2 `begin()` [2/4]

```
void ZEEPROM::begin (
    TwoWire & MyWire,
    uint8_t addr )
```

Definition at line 383 of file ZEEPROM.cpp.

References AT24C512, and `begin()`.

Here is the call graph for this function:



4.1.3.3 `begin()` [3/4]

```
void ZEEPROM::begin (
    uint8_t addr )
```

Setups the I2C interface and hardware.

Definition at line 393 of file ZEEPROM.cpp.

References `begin()`.

Here is the call graph for this function:



4.1.3.4 begin() [4/4]

```
void ZEeprom::begin (
    void )
```

Setups the I2C interface and hardware.

Definition at line 403 of file ZEeprom.cpp.

Referenced by begin().

Here is the caller graph for this function:



4.1.3.5 initialize()

```
void ZEeprom::initialize ( )
```

Initialize library and TWI bus.

If several devices are connected to TWI bus, this method mustn't be called. TWI bus must be initialized out of this library using `Wire.begin()` method.

4.1.3.6 readByte()

```
byte ZEeprom::readByte (
    unsigned int address )
```

Read a byte in EEPROM memory.

Parameters

<i>address</i>	Address.
----------------	----------

Returns

Read byte.

Definition at line 234 of file ZEeprom.cpp.

References `DATAADDRESS`, and `DEVICEADDRESS`.

4.1.3.7 readBytes()

```
void ZEeprom::readBytes (
    unsigned int address,
    unsigned int length,
    byte * p_data )
```

Read bytes in EEPROM memory.

Parameters

<i>address</i>	Start address.
<i>length</i>	Number of bytes to read. [in] p_data Byte array to fill with read bytes.

Definition at line 269 of file ZEeprom.cpp.

References EEPROM__RD_BUFFER_SIZE.

4.1.3.8 writeByte()

```
void ZEeprom::writeByte (
    unsigned int address,
    byte data )
```

Write a byte in EEPROM memory.

Remarks

A delay of 10 ms is required after write cycle.

Parameters

<i>address</i>	Address.
<i>data</i>	Byte to write.

Definition at line 145 of file ZEeprom.cpp.

References DATAADDRESS, and DEVICEADDRESS.

4.1.3.9 writeBytes()

```
void ZEeprom::writeBytes (
    unsigned int address,
    unsigned int length,
    byte * p_data )
```

Write bytes in EEPROM memory.

Parameters

	<i>address</i>	Start address.
	<i>length</i>	Number of bytes to write.
in	<i>p_data</i>	Bytes to write.

Definition at line 175 of file ZEeprom.cpp.

References EEPROM__PAGE_SIZE.

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

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

5 File Documentation

5.1 README.md File Reference

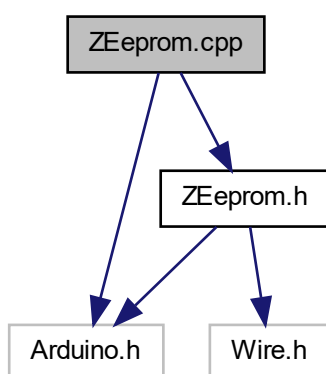
5.2 ZEeprom.cpp File Reference

EEPROM 24C01 /02/04/08/16/32/64/128/254/512/1024 library for Arduino - Demonstration program.

```
#include <Arduino.h>
```

```
#include <ZEeprom.h>
```

Include dependency graph for ZEeprom.cpp:



Macros

- `#define EEPROM__PAGE_SIZE PAGE_LENGTH()`
Size of a page in EEPROM memory. This size is given by EEPROM memory datasheet.
- `#define BUFFER_LENGTH SERIAL_BUFFER_SIZE`
- `#define EEPROM__RD_BUFFER_SIZE BUFFER_LENGTH`
Size of input TWI buffer. This size is equal to BUFFER_LENGTH defined in _i2c library (32 bytes).
- `#define EEPROM__WR_BUFFER_SIZE (BUFFER_LENGTH - 1)`
Size of output TWI buffer. This size is equal to BUFFER_LENGTH - 1 byte reserved for address.
- `#define DEVICEADDRESS ((m_deviceAddress & ~(memorytype >> 24)) | ((memorytype >> 24) & address))`
- `#define DATAADDRESS (address & memorytype & 0x00ffff)`

5.2.1 Detailed Description

EEPROM 24C01 /02/04/08/16/32/64/128/254/512/1024 library for Arduino - Demonstration program.

Author

Copyright (C) 2018 Pierre Valteau - www.zoubworld.com

Version

2.0

Date

201804224 Based on the work of Julien Le Sech - www.idreammicro.com (Copyright (C) 2012) source : https://github.com/jlesech/Eeprom24C01_02.git Extended to 04/08/16/32/64/128/254/512/1024 Eeprom

This file is part of the EEPROM 24C01 /02/04/08/16/32/64/128/254/512/1024 library for Arduino.

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

This library 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 Lesser General Public License for more details.

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

5.2.2 Macro Definition Documentation

5.2.2.1 BUFFER_LENGTH

```
#define BUFFER_LENGTH SERIAL_BUFFER_SIZE
```

Definition at line 56 of file ZEeprom.cpp.

5.2.2.2 DATAADDRESS

```
#define DATAADDRESS (address&memorytype&0x00ffffff)
```

compute the data address, sometime the device address content the data address MSB, some bit are on a 2nd byte.

Definition at line 75 of file ZEeprom.cpp.

Referenced by ZEeprom::readByte(), and ZEeprom::writeByte().

5.2.2.3 DEVICEADDRESS

```
#define DEVICEADDRESS ((m_deviceAddress&~(memorytype>>24)) | ((memorytype>>24)& address))
```

compute the device address, sometime the device address content the data address MSB

Definition at line 72 of file ZEeprom.cpp.

Referenced by ZEeprom::readByte(), and ZEeprom::writeByte().

5.2.2.4 EEPROM__PAGE_SIZE

```
#define EEPROM__PAGE_SIZE PAGE_LENGTH()
```

Size of a page in EEPROM memory. This size is given by EEPROM memory datasheet.

Definition at line 48 of file ZEeprom.cpp.

Referenced by ZEeprom::writeBytes().

5.2.2.5 EEPROM__RD_BUFFER_SIZE

```
#define EEPROM__RD_BUFFER_SIZE BUFFER_LENGTH
```

Size of input TWI buffer. This size is equal to BUFFER_LENGTH defined in _i2c library (32 bytes).

Definition at line 59 of file ZEeprom.cpp.

Referenced by ZEeprom::readBytes().

5.2.2.6 EEPROM__WR_BUFFER_SIZE

```
#define EEPROM__WR_BUFFER_SIZE (BUFFER_LENGTH - 1)
```

Size of output TWI buffer. This size is equal to BUFFER_LENGTH - 1 byte reserved for address.

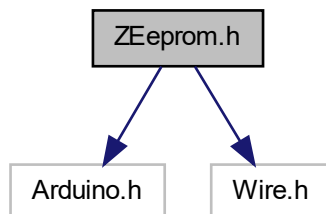
Definition at line 66 of file ZEeprom.cpp.

5.3 ZEeprom.h File Reference

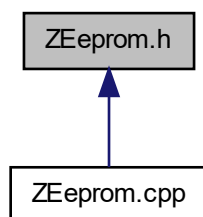
```
#include <Arduino.h>
```

```
#include <Wire.h>
```

Include dependency graph for ZEeprom.h:



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



Data Structures

- class [ZEeprom](#)
EEPROM 24C01 / 24C02 memory driver.

Macros

- #define [AT24C01A](#) ((uint32_t)0x0000007F)
EEPROM 24C01 / 02/04/08/16/32/64/128/254/512/1024 library for Arduino - Demonstration program.
- #define [AT24C02](#) ((uint32_t)0x000000FF)
- #define [AT24C04](#) ((uint32_t)0x010000FF)
- #define [AT24C08A](#) ((uint32_t)0x030000FF)
- #define [AT24C16A](#) ((uint32_t)0x070000FF)
- #define [AT24C32](#) ((uint32_t)0x00000fff)
- #define [AT24C64](#) ((uint32_t)0x00001fff)

- #define `AT24C128` ((uint32_t)0x00003fff)
- #define `AT24C256` ((uint32_t)0x00007fff)
- #define `AT24C512` ((uint32_t)0x0000ffff)
- #define `AT24C1024` ((uint32_t)0x0100ffff)
- #define `AT24Cxx_BASE_ADDR` 0x50

5.3.1 Macro Definition Documentation

5.3.1.1 AT24C01A

```
#define AT24C01A ((uint32_t)0x0000007F)
```

EEPROM 24C01 /02/04/08/16/32/64/128/254/512/1024 library for Arduino - Demonstration program.

Author

Copyright (C) 2018 Pierre Valteau - www.zoubworld.com

Version

2.0

Date

201804224 Based on the work of Julien Le Sech - www.idreammicro.com(Copyright (C) 2012) source :
https://github.com/jlesech/Eeprom24C01_02.git Extended to 04/08/16/32/64/128/254/512/1024
Eeprom

This file is part of the EEPROM 24C01 /02/04/08/16/32/64/128/254/512/1024 library for Arduino.

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

This library 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 Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/the> type of EEPROM content the coding value for address system : 0xddaaaaa,

- aa refer to data address sent after the device address,
- dd refers to MSB of data address apply on the device address.(it's a tip to save a 2nd(or 3rd) byte of data address)

Definition at line 49 of file ZEeprom.h.

5.3.1.2 AT24C02

```
#define AT24C02 ((uint32_t)0x000000FF)
```

Definition at line 50 of file ZEeprom.h.

5.3.1.3 AT24C04

```
#define AT24C04 ((uint32_t)0x010000FF)
```

Definition at line 51 of file ZEeprom.h.

5.3.1.4 AT24C08A

```
#define AT24C08A ((uint32_t)0x030000FF)
```

Definition at line 52 of file ZEeprom.h.

5.3.1.5 AT24C1024

```
#define AT24C1024 ((uint32_t)0x0100ffff)
```

Definition at line 61 of file ZEeprom.h.

5.3.1.6 AT24C128

```
#define AT24C128 ((uint32_t)0x00003fff)
```

Definition at line 58 of file ZEeprom.h.

5.3.1.7 AT24C16A

```
#define AT24C16A ((uint32_t)0x070000FF)
```

Definition at line 53 of file ZEeprom.h.

5.3.1.8 AT24C256

```
#define AT24C256 ((uint32_t)0x00007fff)
```

Definition at line 59 of file ZEeprom.h.

5.3.1.9 AT24C32

```
#define AT24C32 ((uint32_t)0x00000fff)
```

Definition at line 56 of file ZEeprom.h.

5.3.1.10 AT24C512

```
#define AT24C512 ((uint32_t)0x0000ffff)
```

Definition at line 60 of file ZEeprom.h.

Referenced by ZEeprom::begin().

5.3.1.11 AT24C64

```
#define AT24C64 ((uint32_t)0x00001fff)
```

Definition at line 57 of file ZEeprom.h.

5.3.1.12 AT24Cxx_BASE_ADDR

```
#define AT24Cxx_BASE_ADDR 0x50
```

Definition at line 63 of file ZEeprom.h.

Index

AT24C01A
 ZEeprom.h, [11](#)

AT24C02
 ZEeprom.h, [11](#)

AT24C04
 ZEeprom.h, [12](#)

AT24C08A
 ZEeprom.h, [12](#)

AT24C1024
 ZEeprom.h, [12](#)

AT24C128
 ZEeprom.h, [12](#)

AT24C16A
 ZEeprom.h, [12](#)

AT24C256
 ZEeprom.h, [12](#)

AT24C32
 ZEeprom.h, [12](#)

AT24C512
 ZEeprom.h, [13](#)

AT24C64
 ZEeprom.h, [13](#)

AT24Cxx_BASE_ADDR
 ZEeprom.h, [13](#)

BUFFER_LENGTH
 ZEeprom.cpp, [8](#)

begin
 ZEeprom, [3](#), [4](#)

DATAADDRESS
 ZEeprom.cpp, [8](#)

DEVICEADDRESS
 ZEeprom.cpp, [9](#)

EEPROM__PAGE_SIZE
 ZEeprom.cpp, [9](#)

EEPROM__RD_BUFFER_SIZE
 ZEeprom.cpp, [9](#)

EEPROM__WR_BUFFER_SIZE
 ZEeprom.cpp, [9](#)

initialize
 ZEeprom, [5](#)

README.md, [7](#)

readByte
 ZEeprom, [5](#)

readBytes
 ZEeprom, [5](#)

writeByte
 ZEeprom, [6](#)

writeBytes
 ZEeprom, [6](#)

ZEeprom, [2](#)

begin, [3](#), [4](#)

initialize, [5](#)

readByte, [5](#)

readBytes, [5](#)

writeByte, [6](#)

writeBytes, [6](#)

ZEeprom, [3](#)

ZEeprom.cpp, [7](#)

 BUFFER_LENGTH, [8](#)

 DATAADDRESS, [8](#)

 DEVICEADDRESS, [9](#)

 EEPROM__PAGE_SIZE, [9](#)

 EEPROM__RD_BUFFER_SIZE, [9](#)

 EEPROM__WR_BUFFER_SIZE, [9](#)

ZEeprom.h, [10](#)

 AT24C01A, [11](#)

 AT24C02, [11](#)

 AT24C04, [12](#)

 AT24C08A, [12](#)

 AT24C1024, [12](#)

 AT24C128, [12](#)

 AT24C16A, [12](#)

 AT24C256, [12](#)

 AT24C32, [12](#)

 AT24C512, [13](#)

 AT24C64, [13](#)

 AT24Cxx_BASE_ADDR, [13](#)