EEEPROM LIBRARY

Generated by Doxygen 1.9.6

Chapter 1

EEPROM Library Integration Guide

This guide provides instructions on how to integrate the EEPROM library into your STM32 project using STM32Cube IDE or any other compatible IDE.

1.1 Step 1: Download the library files

Download the following EEPROM library files:

1. `EEPROM.h` - The header file containing the function prototypes and necessary definitions.
2. `EEPROM.c` - The source file containing the function implementations.

1.2 Step 2: Add the library files to your project

Follow these steps to add the EEPROM library files to your project:

1.2.1 STM32Cube IDE

1. In STM32Cube IDE, open your STM32 project.
2. Navigate to the project tree in the "Project Explorer" tab.
3. Place the `EEPROM.h` file into the "Inc" folder (or the folder where header files are stored in your project).
4. Place the `EEPROM.c` file into the "Src" folder (or the folder where source files are stored in your project).

1.2.2 Other IDEs

1. Open your STM32 project in the IDE you are using.
2. Place the `EEPROM.h` file in the folder where header files are stored in your project (usually an "include" or "inc" folder).
3. Place the `EEPROM.c` file in the folder where source files are stored in your project (usually a "source" or "src" folder).

1.3 Step 3: Include the library header in main.c

In the `main.c` file of your project, add the following include statement at the beginning of the file, along with other include statements:

```
#include "EEPROM.h"
```

1.4 Step 4: Usage Examples

ere are some examples of how to use the EEPROM library in your STM32 project:

1.4.1 Write data to EEPROM

```
 uint8_t data_to_write[] = "Hello, EEPROM!";
EEPROM_Write(0, 0, data_to_write, sizeof(data_to_write));
```

1.4.2 Read data from EEPROM

```
uint8_t read_buffer[16];
EEPROM_Read(0, 0, read_buffer, sizeof(read_buffer));
```

1.4.3 Erase a page in EEPROM

```
EEPROM_PageErase(2);
```

1.4.4 COmprenhensive example on how to use it in main.c

```
#include "main.h"
#include "EEPROM.h"

I2C_HandleTypeDef hi2c1;

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_I2C1_Init(void);

int main(void)
{
    HAL_Init();
    SystemClock_Config();

    MX_GPIO_Init();
    MX_I2C1_Init();

    // Write data to EEPROM
    uint8_t data_to_write[] = "Hello, EEPROM!";
    EEPROM_Write(0, 0, data_to_write, sizeof(data_to_write));

    // Read data from EEPROM
    uint8_t read_buffer[16];
    EEPROM_Read(0, 0, read_buffer, sizeof(read_buffer));

    // Write a float number to EEPROM
    float number_to_write = 3.14159265;
    EEPROM_Write_NUM(1, 0, number_to_write);

    // Read a float number from EEPROM
    float read_number;
    read_number = EEPROM_Read_NUM(1, 0);

    // Erase a page in EEPROM
    EEPROM_PageErase(2);

    while (1)
    {
        // Main loop
    }
}

void SystemClock_Config(void)
{
    // System clock configuration code...
}

static void MX_GPIO_Init(void)
{
    // GPIO initialization code...
}

static void MX_I2C1_Init(void)
{
    // I2C1 initialization code...
}
```

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

File Documentation

3.1 EEPROM.c File Reference

Using the HAL I2C Functions.

```
#include "EEPROM.h"
#include "math.h"
#include "string.h"
```

Macros

- #define **EEPROM_I2C** &hi2c1
- #define **EEPROM_ADDR** 0xA0
- #define **PAGE_SIZE** 64
- #define **PAGE_NUM** 512

Functions

- void EEPROM_Write (uint16_t page, uint16_t offset, uint8_t ∗data, uint16_t size)

  Write data to the EEPROM.
- void EEPROM_Write_NUM (uint16_t page, uint16_t offset, float data)

  Write a float/integer value to the EEPROM.
- float EEPROM_Read_NUM (uint16_t page, uint16_t offset)

  Read a single float/integer value from the EEPROM.
- void EEPROM_Read (uint16_t page, uint16_t offset, uint8_t ∗data, uint16_t size)

  Read data from the EEPROM.
- void EEPROM_PageErase (uint16_t page)

  Erase a page in the EEPROM Memory.

Variables

- I2C_HandleTypeDef **hi2c1**

3.1.1 Detailed Description

Using the HAL I2C Functions.

Author

ControllersTech

Date

Feb 16, 2021

Attention

3.1.2 Function Documentation

3.1.2.1 EEPROM_PageErase()

```
void EEPROM_PageErase (
            uint16_t page )
```

Erase a page in the EEPROM Memory.

Parameters

page	Number of page to erase In order to erase multiple pages, just use this function in the for loop

Return values

None	

3.1.2.2 EEPROM_Read()

```
void EEPROM_Read (
            uint16_t page,
            uint16_t offset,
            uint8_t * data,
            uint16_t size )
```

Read data from the EEPROM.

Parameters

page	Number of the start page. Range from 0 to PAGE_NUM-1.
offset	Start byte offset in the page. Range from 0 to PAGE_SIZE-1.
data	Pointer to the data to write in bytes.
size	Size of the data.

Return values

None.	

3.1.2.3 EEPROM_Read_NUM()

```
float EEPROM_Read_NUM (
             uint16_t page,
             uint16_t offset )
```

Read a single float/integer value from the EEPROM.

Read a float or integer value from the EEPROM.

Parameters

page	Number of the start page. Range from 0 to PAGE_NUM-1.
offset	Start byte offset in the page. Range from 0 to PAGE_SIZE-1.

Return values

Float/integer	value.

3.1.2.4 EEPROM_Write()

```
void EEPROM_Write (
             uint16_t page,
             uint16_t offset,
             uint8_t * data,
             uint16_t size )
```

Write data to the EEPROM.

Parameters

page	Start page number (0 to PAGE_NUM-1).
offset	Start byte offset in the page (0 to PAGE_SIZE-1).
data	Pointer to the data to write in bytes.
size	Size of the data.

Return values

None	

3.1.2.5 EEPROM_Write_NUM()

```
void EEPROM_Write_NUM (
             uint16_t page,
             uint16_t offset,
             float data )
```

Write a float/integer value to the EEPROM.

Write a float or integer value to the EEPROM.

Parameters

page	Number of the start page. Range from 0 to PAGE_NUM-1.
offset	Start byte offset in the page. Range from 0 to PAGE_SIZE-1.
data	Float/integer value that you want to write.

Return values

None.	

3.2 EEPROM.h File Reference

Using the HAL I2C Functions.

```
#include "stdint.h"
#include "stm32f4xx_hal.h"
```

Functions

- void EEPROM_Write (uint16_t page, uint16_t offset, uint8_t ∗data, uint16_t size)

  Write data to the EEPROM.

- void EEPROM_Read (uint16_t page, uint16_t offset, uint8_t ∗data, uint16_t size)

  Read data from the EEPROM.
- void EEPROM_PageErase (uint16_t page)

  Erase a page in the EEPROM Memory.
- void EEPROM_Write_NUM (uint16_t page, uint16_t offset, float fdata)

  Write a float or integer value to the EEPROM.
- float EEPROM_Read_NUM (uint16_t page, uint16_t offset)

  Read a float or integer value from the EEPROM.

3.2.1 Detailed Description

Using the HAL I2C Functions.

Author

ControllersTech

Date

Feb 16, 2021

Attention

3.2.2 Function Documentation

3.2.2.1 EEPROM_PageErase()

```
void EEPROM_PageErase (
              uint16_t page )
```

Erase a page in the EEPROM Memory.

Parameters

page	Page number to erase.

Return values

None	

Parameters

page	Number of page to erase In order to erase multiple pages, just use this function in the for loop

Return values

None	

3.2.2.2 EEPROM_Read()

```
void EEPROM_Read (
            uint16_t page,
            uint16_t offset,
            uint8_t * data,
            uint16_t size )
```

Read data from the EEPROM.

Parameters

page	Start page number (0 to PAGE_NUM-1).
offset	Start byte offset in the page (0 to PAGE_SIZE-1).
data	Pointer to the data to read in bytes.
size	Size of the data.

Return values

None	

Parameters

page	Number of the start page. Range from 0 to PAGE_NUM-1.
offset	Start byte offset in the page. Range from 0 to PAGE_SIZE-1.
data	Pointer to the data to write in bytes.
size	Size of the data.

Return values

None.	

3.2.2.3 EEPROM_Read_NUM()

```
float EEPROM_Read_NUM (
            uint16_t page,
            uint16_t offset )
```

Read a float or integer value from the EEPROM.

Parameters

page	Start page number (0 to PAGE_NUM-1).
offset	Start byte offset in the page (0 to PAGE_SIZE-1).

Return values

Float	or integer value read from the EEPROM.

Read a float or integer value from the EEPROM.

Parameters

page	Number of the start page. Range from 0 to PAGE_NUM-1.
offset	Start byte offset in the page. Range from 0 to PAGE_SIZE-1.

Return values

Float/integer	value.

3.2.2.4 EEPROM_Write()

```
void EEPROM_Write (
            uint16_t page,
            uint16_t offset,
            uint8_t * data,
            uint16_t size )
```

Write data to the EEPROM.

Parameters

page	Start page number (0 to PAGE_NUM-1).
offset	Start byte offset in the page (0 to PAGE_SIZE-1).
data	Pointer to the data to write in bytes.
size	Size of the data.

Return values

None	

3.2.2.5 EEPROM_Write_NUM()

```
void EEPROM_Write_NUM (
             uint16_t page,
             uint16_t offset,
             float data )
```

Write a float or integer value to the EEPROM.

Parameters

page	Start page number (0 to PAGE_NUM-1).
offset	Start byte offset in the page (0 to PAGE_SIZE-1).
fdata	Float or integer value to write.

Return values

None	

Write a float or integer value to the EEPROM.

Parameters

page	Number of the start page. Range from 0 to PAGE_NUM-1.
offset	Start byte offset in the page. Range from 0 to PAGE_SIZE-1.
data	Float/integer value that you want to write.

Return values

None.	

3.3 EEPROM.h

Go to the documentation of this file.

```
00001
00019 #ifndef INC_EEPROM_H_
00020 #define INC_EEPROM_H_
00021
00022 #include "stdint.h"
00023 #include "stm32f4xx_hal.h"
00024
00033 void EEPROM_Write(uint16_t page, uint16_t offset, uint8_t *data, uint16_t size);
00034
00043 void EEPROM_Read(uint16_t page, uint16_t offset, uint8_t *data, uint16_t size);
```

```

void EEPROM_PageErase(uint16_t page);

void EEPROM_Write_NUM(uint16_t page, uint16_t offset, float fdata);

float EEPROM_Read_NUM(uint16_t page, uint16_t offset);

#endif /* INC_EEPROM_H_ */
```

Index