Embedded System Library Guide

(i2c_memory Library)

Last Modified Date:

Wednesday, July 10, 2019

Code author: Majid Derhambakhsh

GitHub: Majid-Derhambakhsh

E-mail

Contents

i2c_memory Library guide 3

Basic settings

Functions

Requirements 7

Important tips 7

Error & Warning's 8

Supported memories 9

Supported microcontroller's 10

Version history 11

Guide

To use this library, first add the c file to the program and include the following header.

```
#include "i2c_memory.h"
```

Memory Configuration

Follow the steps below to set up your library:

1. First open the file **i2c_memory_conf.h**:

   Edit the following values to adjust the IC code used and its address pins:

   ```
   #define _AT24C04

   #define _MEMORY_A0_PIN_STATE 0
   #define _MEMORY_A1_PIN_STATE 0
   #define _MEMORY_A2_PIN_STATE 0
   ```

   -- Used IC number (available in the supported memory section).

   -- Logic state of memory address pins (0 or 1).

   Edit the following values to config Write Protect pin:

   ```
   #define _WP_DISABLE

   #define _WRITE_PROTECT_PORT  GPIOx
   #define _WRITE_PROTECT_PIN   0
   ```

   -- For disable Write Protection (comment it for activation).

   -- Connected port & pin to Write Protect.

      Allowed value for port: GPIOx in ARM Or PORTx in AVR.

      Allowed value for pin: 0 ~ Number of pins supported per port.

I2C Configuration

Follow the steps below to set up I2C:

1. I2C in AVR:

First open the **i2c_unit_conf.h** file from the I2C_UNIT folder and edit the values below.

```
#define _F_SCL        100000UL
#define _PRESCALER   _PRE1
```

-- I2C Frequency.

-- I2C Unit prescaler (If not correct operation, edit with the values in Table 1-2):

ID	Value
_PRE1	1
_PRE4	4
_PRE16	16
_PRE64	64

Table 1-2

Tip: Add i2c_unit.c & gpio_unit.c in AVR compilers.

2. I2C in ARM:

STM32 Series:

First open the **stm32_i2c _conf.h** file from the **STM32_I2C** folder and edit the values below.

```
#define STM32F1
#define _CONNECTED_I2C hi2c1
#define _MEM_DEF_VAL_BUFF_LENGTH 50
```

-- Microcontroller Series.

-- Structure that contains the configuration information for the specified I2C.

-- Buffer size to clear memory (longer length more speed).

Tip: Add stm32_i2c.c in ARM compilers.

Functions

I2C_Memory_Init

Function name: void I2C_Memory_Init(void)

Function description: This function is used to initialize memory.

Parameters: -

Return values: -

Example:

- I2C_Memory_Init();

I2C_MemoryIsReady

Function name: *uint8_t* I2C_MemoryIsReady(*uint16_t* time_out)

Function description: This function is used to check memory availability.

Parameters

- **time_out:** Timeout value in millisecond.

Return values

- **AVR:** _STAT_OK / _STAT_ERROR
- **ARM:** HAL_OK / HAL_ERROR

Example:

- device_status = I2C_MemoryIsReady(100);

I2C_Memory_SingleWrite

Function name: *uint8_t* I2C_Memory_SingleWrite(*uint32_t* address, *uint8_t* udata, *uint16_t* time_out)

Function description: This function is used to write a byte of data to memory.

Parameters

- **address:** Memory address for write data.
- **udata:** Data for write in memory.
- **time_out:** Timeout value in millisecond.

Return values

- _MEM_SIZE_ERROR
- **AVR:** _STAT_OK / _STAT_ERROR
- **ARM:** HAL_OK / HAL_ERROR

Example:

- com_status = I2C_Memory_SingleWrite(0, 'A', 50);

I2C_Memory_BurstWrite

Function name `uint8_t I2C_Memory_BurstWrite(uint32_t address, uint8_t *udata, uint32_t size, uint16_t time_out)`

Function description This function is used to write a string of data to memory.

Parameters

- **address:** Memory address for write data.
- **udata:** Data's for write in memory.
- **size:** Length of data.
- **time_out:** Timeout value in millisecond.

Return values

- _MEM_SIZE_ERROR
- **AVR:** _STAT_OK / _STAT_ERROR
- **ARM:** HAL_OK / HAL_ERROR

Example:

- `com_status = I2C_Memory_BurstWrite(0, "Hello", 5, 50);`

I2C_Memory_SingleRead

Function name `uint8_t I2C_Memory_SingleRead(uint32_t address, uint8_t *udata, uint16_t time_out)`

Function description This function is used to read a byte of data from memory.

Parameters

- **address:** Memory address for reading data.
- **udata:** Pointer to the variable to store the data received from memory.
- **time_out:** Timeout value in millisecond.

Return values

- _MEM_SIZE_ERROR
- **AVR:** _STAT_OK / _STAT_ERROR
- **ARM:** HAL_OK / HAL_ERROR

Example:

- `com_status = I2C_Memory_SingleRead(0, &my_data, 50);`

I2C_Memory_BurstRead

Function name `uint8_t I2C_Memory_BurstRead(uint32_t address, uint8_t *udata, uint32_t size, uint16_t time_out)`

Function description This function is used to read a data string from memory.

Parameters

- **address:** Memory address for reading data.
- **udata:** Pointer to the variable to store the data received from memory.
- **size:** Length of data.
- **time_out:** Timeout value in millisecond.

Return values

- _MEM_SIZE_ERROR
- **AVR:** _STAT_OK / _STAT_ERROR
- **ARM:** HAL_OK / HAL_ERROR

Example:

- com_status = I2C_Memory_BurstRead(12, received_data_array, 10, 50);

I2C_Memory_Erase

Function name *uint8_t* I2C_Memory_Erase(*uint32_t* address, *uint32_t* quantity, *uint16_t* time_out)

Function description This function is used to read a data string from memory.

Parameters

- **address:** Memory address for reading data.
- **quantity:** Memory length to clear.
- **time_out:** Timeout value in millisecond.

Return values

- _MEM_SIZE_ERROR
- **AVR:** _STAT_OK / _STAT_ERROR
- **ARM:** HAL_OK / HAL_ERROR

Example:

- com_status = I2C_Memory_Erase(0, 65000, 50);

Requirement

- i2c_unit & gpio_unit drivers for AVR microcontrollers.
- HAL & stm32_i2c drivers for ARM microcontrollers STM32 series.

Important tips

- All commands and settings begin with _.
- All functions are written as Camel Case.
- The functions and codes used in all microcontrollers are the same.

Error & Warning's

Error's:

- Chip or I2C Library not supported: This error occurs when the microcontroller or its library not supported.
- Memory is not selected or not supported: This error occurs when the memory is not correctly specified or not supported by the library.

Warning's:

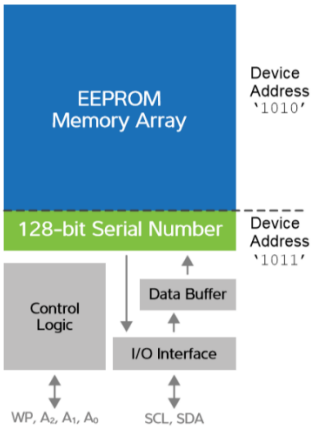
- Your Ax Pin state in not correct: This warning appears when the address pins not set correctly.

Supported memories

All 24Cxx family memories:

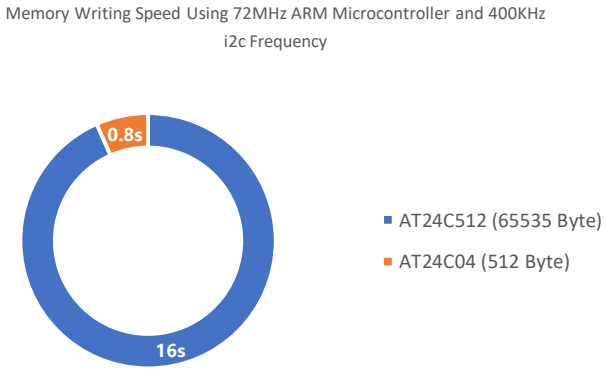
AT24C01 – AT24C02 – AT24C04 – AT24C08 – AT24C16 – AT24C32 – AT24C64 – AT24C128 – AT24C256 – AT24C512

AT24C1024



Memory blocks

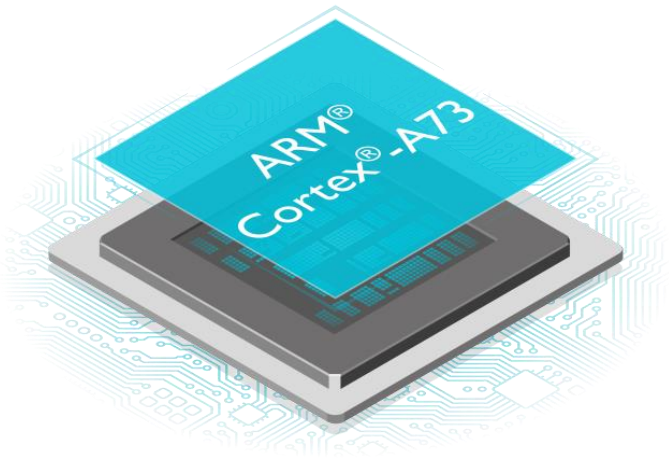
Speed of data writing



Supported microcontroller's

ATmega & ATtiny series of AVR Microcontroller's with i2c_unit & gpio_unit driver's

Codevision and GNUC compilers such as AtmelStudio



All STM32 series of ARM Microcontroller's with HAL & stm32_i2c driver's

All ARM Compiler's

Version history

Version 0.0.1

Fix: _MEM_SIZE_ERROR in Capacities greater than 256Kb

Version 0.0.0

Stable and tested version
