

C8E

0.1

Generated by Doxygen 1.7.2

Sun Dec 12 2010 21:33:10

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	src/Logs.h File Reference	3
2.1.1	Define Documentation	3
2.1.1.1	DEFAULT_DEBUG_LEVEL	3
2.1.1.2	DEFAULT_OUTPUT_FILE	3
2.1.2	Enumeration Type Documentation	4
2.1.2.1	DEBUG_LEVELS	4
2.1.3	Function Documentation	4
2.1.3.1	addEntry	4
2.1.3.2	closeLogs	4
2.1.3.3	setupLogs	4
2.1.3.4	setupLogs	4
2.1.3.5	setupLogs	4
2.2	src/Memory.c File Reference	4
2.2.1	Detailed Description	4
2.2.2	Function Documentation	5
2.2.2.1	read	5
2.2.2.2	setupMemory	5
2.2.2.3	write	5
2.3	src/Memory.h File Reference	6
2.3.1	Define Documentation	6
2.3.1.1	DATA_SPACE_START	6
2.3.1.2	DATA_SPACE_STOP	7
2.3.1.3	MAX_REGISTERS	7
2.3.1.4	RESERVED_MEMORY_START	7
2.3.1.5	RESERVED_MEMORY_STOP	7
2.3.2	Function Documentation	7
2.3.2.1	read	7
2.3.2.2	setupMemory	7
2.3.2.3	write	8

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

src/ Logs.h	3
src/ Memory.c (Define all functions, variables and defines for memory management)	4
src/ Memory.h	6

Chapter 2

File Documentation

2.1 src/Logs.h File Reference

Defines

- #define `DEFAULT_DEBUG_LEVEL` 1
- #define `DEFAULT_OUTPUT_FILE` "DEBUG_LOGS"

Enumerations

- enum `DEBUG_LEVELS` { `ERROR` = 0, `WARNING` = 1, `DRAWING` = 2, `DISASSEMBLING` = 3 }

Functions

- void `setupLogs` ()
- void `setupLogs` (unsigned char debugLevel)
- void `setupLogs` (unsigned char debugLevel, char *outputFile)
- void `closeLogs` ()
- void `addEntry` (`DEBUG_LEVELS` level, const char *const message)

2.1.1 Define Documentation

2.1.1.1 #define `DEFAULT_DEBUG_LEVEL` 1

Definition at line 23 of file Logs.h.

2.1.1.2 #define `DEFAULT_OUTPUT_FILE` "DEBUG_LOGS"

Definition at line 26 of file Logs.h.

2.1.2 Enumeration Type Documentation

2.1.2.1 enum DEBUG_LEVELS

Enumerator:

ERROR
WARNING
DRAWING
DISASSEMBLING

Definition at line 19 of file Logs.h.

2.1.3 Function Documentation

2.1.3.1 void addEntry (DEBUG_LEVELS *level*, const char *const *message*)

2.1.3.2 void closeLogs ()

2.1.3.3 void setupLogs (unsigned char *debugLevel*)

2.1.3.4 void setupLogs (unsigned char *debugLevel*, char * *outputFile*)

2.1.3.5 void setupLogs ()

2.2 src/Memory.c File Reference

Define all functions, variables and defines for memory management.

Functions

- int [setupMemory](#) ()
Initialize memory to 0.
- int [write](#) (unsigned short addr, char *const data, unsigned int len)
write [len] bytes from [data] into memory at adress [addr]
- int [read](#) (short addr, unsigned short len, char *const buffer)
Read [len] bytes of data from address [addr] to buffer.

2.2.1 Detailed Description

Define all functions, variables and defines for memory management.

Version

0.1

Date

December 12, 2010

Author

Maxime Gaudin

Definition in file [Memory.c](#).**2.2.2 Function Documentation****2.2.2.1 int read (short *addr*, unsigned short *len*, char *const *buffer*)**Read [*len*] bytes of data from address [*addr*] to buffer.**Parameters**

in	<i>addr</i>	Address where read
in	<i>len</i>	Number of bytes read
out	<i>buffer</i>	Pointer to the data buffer

Returns

0 if success, 1 otherwise.

Definition at line 39 of file Memory.c.

2.2.2.2 int setupMemory ()

Initialize memory to 0.

Returns

0 if success, 1 otherwise.

Definition at line 19 of file Memory.c.

2.2.2.3 int write (unsigned short *addr*, char *const *data*, unsigned int *len*)write [*len*] bytes from [*data*] into memory at address [*addr*]**Parameters**

in	<i>addr</i>	Address where data will be written
in	<i>data</i>	Pointer to data buffer
in	<i>len</i>	Number of byte written

Returns

0 if success, 1 otherwise.

Definition at line 30 of file Memory.c.

2.3 src/Memory.h File Reference

Defines

- `#define RESERVED_MEMORY_START 0x0`
Specifies where memory starts (0x0, what a surprise isn't it ??).
- `#define RESERVED_MEMORY_STOP 0x200`
Specifies where the memory stops.
- `#define DATA_SPACE_START 0x200`
Specifies the beginning of the data space.
- `#define DATA_SPACE_STOP 0xFFF`
Specifies the end of the data space.
- `#define MAX_REGISTERS 0xF`
Specifies the maximum number of registers..

Functions

- `int setupMemory ()`
Initialize memory to 0.
- `int write (unsigned short addr, char *const data, unsigned int len)`
write [len] bytes from [data] into memory at adress [addr]
- `int read (short addr, unsigned short len, char *const buffer)`
Read [len] bytes of data from address [addr] to buffer.

2.3.1 Define Documentation

2.3.1.1 `#define DATA_SPACE_START 0x200`

Specifies the beginning of the data space.

Definition at line 36 of file Memory.h.

2.3.1.2 #define DATA_SPACE_STOP 0xFF

Specifies the end of the data space.

Definition at line 38 of file Memory.h.

2.3.1.3 #define MAX_REGISTERS 0xF

Specifies the maximum number of registers..

Definition at line 41 of file Memory.h.

2.3.1.4 #define RESERVED_MEMORY_START 0x0

Specifies where memory starts (0x0, what a surprise isn't it ??).

Definition at line 31 of file Memory.h.

2.3.1.5 #define RESERVED_MEMORY_STOP 0x200

Specifies where the memory stops.

Definition at line 33 of file Memory.h.

2.3.2 Function Documentation

2.3.2.1 int read (short *addr*, unsigned short *len*, char *const *buffer*)

Read [*len*] bytes of data from address [*addr*] to buffer.

Parameters

in	<i>addr</i>	Address where read
in	<i>len</i>	Number of bytes read
out	<i>buffer</i>	Pointer to the data buffer

Returns

0 if success, 1 otherwise.

Definition at line 39 of file Memory.c.

2.3.2.2 int setupMemory ()

Initialize memory to 0.

Returns

0 if success, 1 otherwise.

Definition at line 19 of file Memory.c.

2.3.2.3 int write (unsigned short *addr*, char *const *data*, unsigned int *len*)

write [len] bytes from [data] into memory at adress [addr]

Parameters

in	<i>addr</i>	Address where data will be written
in	<i>data</i>	Pointer to data buffer
in	<i>len</i>	Number of byte written

Returns

0 if success, 1 otherwise.

Definition at line 30 of file Memory.c.

Index

addEntry
 Logs.h, [4](#)

closeLogs
 Logs.h, [4](#)

DATA_SPACE_START
 Memory.h, [6](#)

DATA_SPACE_STOP
 Memory.h, [6](#)

DEBUG_LEVELS
 Logs.h, [4](#)

DEFAULT_DEBUG_LEVEL
 Logs.h, [3](#)

DEFAULT_OUTPUT_FILE
 Logs.h, [3](#)

DISASSEMBLING
 Logs.h, [4](#)

DRAWING
 Logs.h, [4](#)

ERROR
 Logs.h, [4](#)

Logs.h
 addEntry, [4](#)
 closeLogs, [4](#)
 DEBUG_LEVELS, [4](#)
 DEFAULT_DEBUG_LEVEL, [3](#)
 DEFAULT_OUTPUT_FILE, [3](#)
 DISASSEMBLING, [4](#)
 DRAWING, [4](#)
 ERROR, [4](#)
 setupLogs, [4](#)
 WARNING, [4](#)

MAX_REGISTERS
 Memory.h, [7](#)

Memory.c
 read, [5](#)
 setupMemory, [5](#)
 write, [5](#)

Memory.h
 DATA_SPACE_START, [6](#)
 DATA_SPACE_STOP, [6](#)
 MAX_REGISTERS, [7](#)
 read, [7](#)
 RESERVED_MEMORY_START, [7](#)
 RESERVED_MEMORY_STOP, [7](#)
 setupMemory, [7](#)
 write, [8](#)

read
 Memory.c, [5](#)
 Memory.h, [7](#)

RESERVED_MEMORY_START
 Memory.h, [7](#)

RESERVED_MEMORY_STOP
 Memory.h, [7](#)

setupLogs
 Logs.h, [4](#)

setupMemory
 Memory.c, [5](#)
 Memory.h, [7](#)

src/Logs.h, [3](#)

src/Memory.c, [4](#)

src/Memory.h, [6](#)

WARNING
 Logs.h, [4](#)

write
 Memory.c, [5](#)
 Memory.h, [8](#)