SRAM_main

Generated by Doxygen 1.8.2

Sat Dec 22 2012 21:58:51

Contents

Chapter 1

SRAM

Library for 23K640 SPI SRAM

Microchip 23K640 is a SPI 64Kb = 8KB SRAM

Developed with `embedXcode`

Author

Rei VILO
embedXcode.weebly.com

Date

déc. 22, 2012 14:26

Version

101

Copyright

See Also

ReadMe.txt for references

- 23A640/23K640 64K SPI Bus Low-Power Serial SRAM Data Sheet
  `http://ww1.microchip.com/downloads/en/DeviceDoc/22126E.pdf`
- Recommended Usage of Microchip 23X256/23X640 SPI Serial SRAM Devices
  `http://ww1.microchip.com/downloads/en/AppNotes/01245C.pdf`

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

Chapter 4

Class Documentation

4.1 SRAM Class Reference

Class SRAM.

```
#include <SRAM.h>
```

Public Member Functions

- SRAM (uint8_t pinChipSelect)

  Constructor.
- void begin ()

  Initialisation.
- void setMode (uint8_t mode)

  Set mode.
- uint8_t getMode ()

  Get mode.
- void write (uint16_t address, uint8_t ∗data, uint16_t length)

  Write length bytes from data to memory starting at address.
- void read (uint16_t address, uint8_t ∗data, uint16_t length)

  Read length bytes from memory starting at address to data.

4.1.1 Detailed Description

Class SRAM.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 SRAM::SRAM (uint8_t *pinChipSelect*)

Constructor.

Parameters

pinChipSelect	pin for chip select

4.1.3 Member Function Documentation

4.1.3.1 uint8_t SRAM::getMode ( )

Get mode.

Returns

byte, page or sequence mode

4.1.3.2 void SRAM::read (uint16_t *address,* uint8_t $*$ *data,* uint16_t *length*)

Read length bytes from memory starting at address to data.

Parameters

address	uint16 address
data	data
length	length in bytes

4.1.3.3 void SRAM::setMode (uint8_t *mode*)

Set mode.

Parameters

mode	byte, page or sequence mode Initialisation

4.1.3.4 void SRAM::write (uint16_t *address,* uint8_t $*$ *data,* uint16_t *length*)

Write length bytes from data to memory starting at address.

Parameters

address	uint16 address
data	data
length	length in bytes

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

- SRAM.h
- SRAM.cpp

Chapter 5

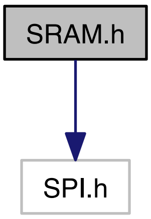
File Documentation

5.1 SRAM.h File Reference

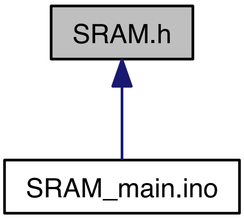
Library header.

```
#include "SPI.h"
```

Include dependency graph for SRAM.h:



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



Classes

- class SRAM

    Class SRAM.

Macros

Instruction Set

- #define SRAM_READ 0b00000011

    Read data from memory array beginning at selected address.
- #define SRAM_WRITE 0b00000010

    Write data to memory array beginning at selected address.
- #define SRAM_READ_STATUS 0b00000101

    Read STATUS register.
- #define SRAM_WRITE_STATUS 0b00000001

    Write STATUS register.

Status Register Instruction

5.1.1 Initialisation

- #define SRAM_BYTE_MODE 0b00000000

    Byte mode (default operation)
- #define SRAM_PAGE_MODE 0b10000000

    Page mode.
- #define SRAM_SEQUENCE_MODE 0b01000000

    Sequential mode.
- #define SRAM_RESERVED_MODE 0b11000000

    Reserved.
- #define SRAM_HOLD_MODE 0b00000001

    Set this bit to DISABLE hold mode.

5.1.2 Detailed Description

Library header. Library for 23K640 SPI SRAM

Project chipKIT_SRAM

Developed with `embedXcode`

Author

Rei VILO
embedXcode.weebly.com

Date

déc. 22, 2012 14:26

Version

101

Copyright

© Rei VILO, 2012
CC = BY NC SA

See Also

ReadMe.txt for references

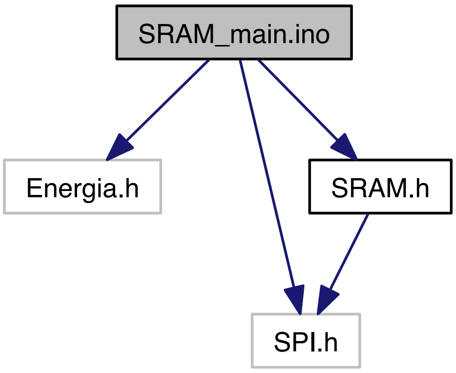
5.2 SRAM_main.ino File Reference

Main sketch

This example.

```
#include "Energia.h"
#include "SPI.h"
#include "SRAM.h"
```

Include dependency graph for SRAM_main.ino:



Functions

- void setup (void)
- void **loop** (void)

Variables

- const uint16_t **MAX** = 130
- uint8_t **modulo** = 26
- char **buffer** [MAX]
- uint8_t **i** = 'A'

5.2.1 Detailed Description

Main sketch

This example.

- prints 130 columns, saves them into the SRAM,
- then reads them back from SRAM and prints them again.

101

Developed with `embedXcode`

Author

Rei VILO
embedXcode.weebly.com

Date

déc. 22, 2012 14:26

Version

101

Copyright

© Rei VILO, 2012
CC = BY NC SA

See Also

ReadMe.txt for references

5.2.2 Function Documentation

5.2.2.1 void setup (void)

Note

SPI speed difference

- SPI_CLOCK_DIV2 for MSP430G2553 gives 8 MHz
- SPI_CLOCK_DIV2 for LM4F120H5QR gives 4 MHz!

Warning

SPI maximum speed

- SPI_CLOCK_DIV8 for MSP430G2553
- SPI_CLOCK_DIV2 for LM4F120H5QR