SPI Low Level Drivers User Guide

Release Version ~~11.4.1~~

Publication Number SPI_Low_Level_Drivers_UG **Revision** 02 **Issue Date** December 8, 2010

Table of Contents

1. Overview

This document describes the functions and features contained in the SPI Low Level Driver (SLLD). The SLLD contains both higher-level “operation” functions and low-level “command” functions. The operation functions implement programming operations, erase operations and protect operations. Additionally, the “command functions” allow the programmer to easily add new functionality by invoking the lower-level command functions in the desired sequence. Please refer to Section 6., *Porting Considerations* on page 10 for more details regarding the SLLD system integration.

2. Function Categories

The SLLD functions fall into three categories:

- Operation Functions:

These functions implement embedded operations by invoking the “command” functions, and then polling for embedded operation completion.

The operation function naming convention is: `slld_xxxxOp`

e.g. `slld_PPOp`

This function implements Page Programming operation

- Command Functions:

These functions send flash command sequences to the device.

The command function naming convention is: `slld_xxxxCmd`

e.g. `slld_ReadCmd`

This function writes a Read Command to Flash Device and read data.

- Utility Functions:

Several utility functions are available to a calling application as well. These functions do not act directly on the flash device but encapsulate common tasks.

The utility function names do not have a `Cmd` or `Op` suffix.

e.g. `slld_StatusGet`

This function determines the flash device status and returns the information back to the caller (e.g. device is busy or device is not busy)

2.1 Parameter Description

The following is a list of parameters used in the SLLD.

`sys_addr`	device address given by system: must be 32 bits
`source`	a single byte to write to flash
`target`	variable in which to store read data
`data_buf`	variable containing data to program
`len_in_bytes`	number of bytes on which to operate
`dev_status_ptr`	variable to store device status
`dev_softwareprotect_status_ptr`	variable to store device software protect status
`status_val`	variable to store status register value
`config_val`	variable to store configuration register value
`asp_val`	variable to store ASP register value
`bnk_val`	variable to store bank addressing register value
`abt_val`	variable to store Auto Boot register value
`mode`	variable to the mode bits value

2.2 Data Types

For portability, typedefs are used for basic data types.

```
typedef unsigned char  BYTE;          /*  8 Bits wide */
typedef unsigned short WORD;          /* 16 Bits wide */
typedef unsigned long  DWORD;         /* 32 Bits wide */
typedef BYTE           FLASHDATA;     /*  8 Bits wide */
```

These definitions may require modification on the target system.

The following additional typedefs are used in the SLLD:

`ADDRESS`	Used for system level addressing. Must be implemented as a 32-bit unsigned integer.
`PARAM`	Used for function options.
`BYTECOUNT`	Indicates number of bytes on which to operate. Typically implemented as a 32-bit unsigned integer.

`DEVSTATUS`	an `enum` listing possible device statuses
	`dev_status_unknown,`
	`dev_not_busy,`
	`dev_program_error,`
	`dev_erase_error,`
	`dev_suspend,`
	`dev_busy.`

`DEV_SOFTWARE_PROTECT_STATUS`	an `enum` listing possible device software protect statuses
	`FLASH_SOFTWARE_UNPROTECTED`
	`FLASH_SOFTWARE_PROTECTED`

`SLLD_STATUS`	an `enum` listing function return values
	`SLLD_OK,`
	`SLLD_E_DEVICE_SOFTARE_PROTECTED,`
	`SLLD_E_HAL_ERROR,`
	`SLLD_ERROR.`

2.3 Functions Returned Value Description

The following returned values are listed in an `enum SLLD_STATUS`.

`SLLD_OK`	Function finishes successfully. **Note**: For some operation functions (e.g., `slld_PPOp, slld_SEOp`, etc) if the target area is protected function returns `SLLD_OK` but device does not execute the operation. In this case, you will have to check the dev_status_ptr returned value.
`SLLD_E_DEVICE_SOFTWARE_PROTECTED`	Command is not accepted because the target device is in software protect mode.
`SLLD_E_HAL_ERROR`	Error occurs during HAL function.
`SLLD_ERROR`	Error occurs during operation function.

3. Hardware Abstraction Layer (HAL)

The Hardware Abstraction Layer is used to adapt the SLLD to the target system.

The SLLD HAL consists of two functions:

`FLASH_READ(…)`	Basic read function – (one CS# cycle).
`FLASH_WRITE(…)`	Basic write function – (one CS# cycle).

These functions return `SLLD_OK` on success and `SLLD_E_HAL_ERROR` on failure.

They take the following parameters:

`BYTE`	`command,`	/* command to write to the SPI flash */
`ADDRESS`	`sys_addr,`	/* system address to be used */
`BYTE`	`*data_buffer,`	/* Pointer to the data buffer containing data to be written (respectively to be read) */
`int`	`Number_Of_Bytes`	/* number of bytes to be written */

The basic source code for the HAL functions is provided, however, this code is not complete and changes will be required to the HAL functions to adapt them to the target system and optimize them for performance. The provided source code contains comments that will guide you through your customization process.

4. Functions List

For the details, please refer to Section 7., *Appendix – API Details* on page 11.

4.1 Operation Functions

`slld_WriteOp` `on page 11`

Performs a Single / Quad Page Programming Operation

`slld_ReadOp` `on page 11`

Performs a Single / Fast / Dual / Quad Read Operation

`slld_PPOp` `on page 12`

Performs a Page Programming Operation

`slld_PP_4BOp` `on page 12`

Performs a Page Programming Operation using 4-bytes addressing scheme

`slld_QPPOp` `on page 13`

Performs a Quad input Page Programming Operation

`slld_QPP_4BOp` `on page 13`

Performs a Quad input Page Programming Operation using 4-bytes addressing scheme

`slld_BufferedProgramOp` `on page 14`

Performs a Programming Operation. Unlike `slld_PPOp`, this function enables program operation over page boundary

`slld_BufferedProgram_4BOp` `on page 14`

Performs a Programming Operation using 4-bytes addressing scheme. Unlike `slld_PP_4BOp`, this function enables program operation over page boundary

`slld_OTPPOp` `on page 14`

Performs a OTP Programming Operation

slld_SEOp on page 15

Performs a Sector Erase Operation

slld_SE_4BOp on page 15

Performs a Sector Erase Operation using 4-bytes addressing scheme

slld_P4EOp on page 15

Performs a Parameter sector Erase Operation. This function erases one of the 4 KB sectors

slld_P8EOp on page 16

Performs a Parameter sector Erase Operation. This function erases two of the 4 KB sectors

slld_P8E_4BOp on page 16

Performs a Parameter sector Erase Operation using 4-bytes addressing scheme. This function erases two of the 4 KB sectors

slld_BEOp on page 16

Performs a Bulk Erase Operation

slld_WRSROp on page 17

Performs a Write Status Register Operation

slld_WRROp on page 17

Writes a Write Registers Command Sequence to Flash Device

slld_WASPOp on page 17

Writes a Write ASP Command Sequence to Flash Device

slld_WBNKOp on page 18

Writes a Write bank addressing Command Sequence to Flash Device

slld_WABTOp on page 18

Writes a Write Auto Boot Command Sequence to Flash Device

slld_WPWDOp on page 18

Writes a Write password Command Sequence to Flash Device

slld_BlockProtectOp on page 19

Performs a Block Protect Operation

slld_PPB_PGOp on page 19

Performs a PPB programming Operation

slld_DYB_PGOp on page 19

Performs a DYB programming Operation

4.2 Command Functions

All Command Functions except `slld_Read_IDCmd`, `slld_SPCmd` and `slld_RESCmd` check the software protect status of target device (this information is stored in RAM) before issuing command sequences.

slld_ReadCmd on page 20

Writes a Read Command to Flash Device and reads data

slld_Read_4BCmd on page 20

Writes a Read Command to Flash Device and reads data using 4-bytes addressing scheme

slld_Fast_ReadCmd on page 20

Writes a Flash Read Command Sequence to Flash Device and read data

slld_Fast_Read_4BCmd on page 21

Writes a Flash Read Command Sequence to Flash Device and reads data using 4-bytes addressing scheme

slld_DualIOReadCmd on page 21

Writes a Dual I/O Read Command to Flash Device and reads data

slld_DualIORead_4BCmd on page 21

Writes a Dual I/O Read Command to Flash Device and reads data using 4-bytes addressing scheme

slld_DualIOHPReadCmd on page 22

Writes a Dual I/O High Performance Read Command to Flash Device and reads data

slld_DualIOHPRead_4BCmd on page 22

Writes a Dual I/O High Performance Read Command to Flash Device and read data using 4-bytes addressing scheme

slld_QuadIOReadCmd on page 22

Writes a Quad I/O Read Command to Flash Device and reads data

slld_QuadIORead_4BCmd on page 23

Writes a Quad I/O Read Command to Flash Device and reads data using 4-bytes addressing scheme

slld_QuadIOHPReadCmd on page 23

Writes a Quad I/O High Performance Read Command to Flash Device and reads data

slld_QuadIOHPRead_4BCmd on page 23

Writes a Quad I/O High Performance Read Command to Flash Device and reads data using 4-bytes addressing scheme

slld_Read_IDCmd on page 24

Writes a Read ID Command Sequence to Flash Device and reads Device_ID

slld_RDIDCmd on page 24

Writes a RDID Command Sequence to Flash Device and reads Device_ID

slld_Read_IdentificationCmd on page 24

Writes a Read Electronic ID Command Sequence to Flash Device and reads Device_ID

slld_RDSRCmd on page 25

Writes a Read from Status Register Command Sequence to Flash Device and reads status register

slld_SRSTCmd on page 25

Writes the software reset command to the flash device

slld_RASPCmd on page 25

Writes a Read from ASP Register Command Sequence to Flash Device and reads ASP register *`slld_RBNKCmd` on page 26*

Writes a Read from Bank Addressing Register Command Sequence to Flash Device and reads the bank addressing register

`slld_RABTCmd` on page 26

Writes a Read from Auto Boot Register Command Sequence to Flash Device and reads Auto Boot register

`slld_RECCCmd` on page 26

Writes a Read from ECC Register Command Sequence to Flash Device and reads ECC register

`slld_RPWDCmd` on page 26

Writes a Read from password Command Sequence to Flash Device and reads the password

`slld_RCRCmd` on page 27

Writes a Read Configuration Register Command Sequence to Flash Device and reads configuration register

`slld_WRENCmd` on page 27

Writes a Write Enable Command Sequence to Flash Device

`slld_WRDICmd` on page 27

Writes a Write Disable Command Sequence to Flash Device

`slld_WRSRCmd` on page 27

Writes a Write Status Register Command Sequence to Flash Device

`slld_WRRCmd` on page 28

Writes a Write Registers Command Sequence to Flash Device

`slld_WASPCmd` on page 28

Writes a Write ASP register Command Sequence to Flash Device

`slld_WBNKCmd` on page 28

Writes a Write bank addressing Command Sequence to Flash Device

`slld_WABTCmd` on page 28

Writes a Write Auto Boot Register Command Sequence to Flash Device

`slld_WPWDCmd` on page 29

Writes a write password Command Sequence to Flash Device

`slld_PPCmd` on page 29

Writes a Page Program Command Sequence to Flash Device

`slld_PP_4BCmd` on page 29

Writes a Page Program Command Sequence to Flash Device using 4-bytes addressing scheme

`slld_QPPCmd` on page 30

Writes a Quad input Page Program Command Sequence to Flash Device

`slld_QPP_4BCmd` on page 30

Writes a Quad input Page Program Command Sequence to Flash Device using 4-bytes addressing scheme

`slld_SECmd` on page 30

Writes a Sector Erase Command Sequence to Flash Device

`slld_SE_4BCmd` on page 30

Writes a Sector Erase Command Sequence to Flash Device using 4-bytes addressing scheme *slld_ERS_SSPCmd* on page 31

Writes a Sector Erase Suspend command to Flash Device

slld_ERS_RESCmd on page 31

Writes a Sector Erase Resume command to Flash Device

slld_RCVRCmd on page 31

Writes a Initiate Recovery mode command to Flash Device

slld_RCSPCmd on page 31

Writes a Recovery Suspend command to Flash Device

slld_RCRSCmd on page 32

Writes a Recovery Resume command to Flash Device

slld_P4ECmd on page 32

Writes a 4KB Parameter Sector Erase Command Sequence to Flash Device

slld_P8ECmd on page 32

Writes an 8KB Parameter Sector Erase Command Sequence to Flash Device

slld_P8E_4BCmd on page 32

Writes an 8 KB Parameter Sector Erase Command Sequence to Flash Device using 4-bytes addressing scheme

slld_BECmd on page 33

Writes a Bulk Erase Command Sequence to Flash Device

slld_OTPPCmd on page 33

Writes an OTP Program Command Sequence to Flash Device

slld_OTPRCmd on page 33

Writes an OTP Read Command Sequence to Flash Device and reads OTP

slld_SPCmd on page 33

Writes a Software Protect Command Sequence to Flash Device

slld_ClearStatusRegisterCmd on page 34

Writes a Clear Status Register Command Sequence to Flash Device

slld_PPB_PGCmd on page 34

Writes a PPB program Command Sequence to Flash Device

slld_DYB_PGCmd on page 34

Writes a DYB program Command Sequence to Flash Device

4.3 Utility Functions

slld_Poll on page 35

Polls flash device for embedded operation completion

slld_StatusGet on page 35

Determines Flash Status

slld_SoftwareProtectStatusGet on page 35

Gets the software protect status from the variable in RAM

5. Files

The SLLD source code is provided as five files (three header files and two C source code files).

`slld.h`	header file containing SLLD function prototypes
`slld_target_specific.h`	header file containing code customization macros
`slld.c`	C file containing SLLD function definitions
`slld_hal.h`	header file containing HAL function prototypes
`slld_hal_example.c`	C example file containing HAL function definitions

We provide the `trace.c / trace.h` modules that allow you to enable the software traces, which helps a lot during debug phases.

6. Porting Considerations

Spansion's SPI Low Level Driver (SLLD) is written in ANSI C for easy integration with customer applications. To port the SLLD to a given system, the programmer must modify HAL `typedefs #defines` and system specific HAL functions.

The `slld_target_specific.h, slld_hal.h` and `slld.h` files contain all defines that must be changed when porting the SLLD to your system. You can also find in the `slld_hal.h` header file the API parameters and the HAL function prototypes.

The `slld_target_specific.h` header file allows you to customize the SLLD code by enabling or disabling some features which you might not be using or in case you're worried about the code footprint. You should enable the correspondent macro to the device you're using in your system (FL-A, FL-D, FL-P…).

The S/W Trace feature should be enabled in here by un-commenting the line where TRACE macro is defined. Nevertheless, and in order to be able to use the S/W trace, you should make sure to link `trace.c` in your project. This module is made available for you upon your request through the Spansion® technical support web form.

Typedefs

The following `typedefs`, located in `slld.h`, may need to be updated for your target system:

```
typedef unsigned char  BYTE;   /* 8 bits wide */
typedef unsigned short WORD;   /* 16 bits wide */
typedef unsigned long  DWORD;  /* 32 bits wide */
```

If these default assignments happen to be convenient for your target system, no modification is required.

On the other hand, FLASH_READ() and FLASH_WRITE() functions in `slld_hal_example.c` have to be adapted in order to reflect your controller specifications and requirements which is going to drive the SPI chip cycles. You can find some helping comments in the locations where your specific target code would have to be inserted.

7. Appendix – API Details

7.1 Operation Functions

Function Name	slld_WriteOp
Purpose	Performs a Single / Quad Page Programming Operation
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
len_in_bytes	number of bytes to program
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function programs location to the specified data. If the data size to program is larger than PAGE_SIZE, this function will execute page programming operation. NOTE: If the page boundary is encountered during page programming, additional bytes are wrapped around to the start of the same page. If WriteOp is attempted on the protected area, the function returns SLLD_OK but program operation is not executed.

Function Name	slld_ReadOp
Purpose	Performs a Single / Fast, Dual / Quad Read Operation
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Read commands (Single / Fast / Dual / Quad) to SPI Flash and reads data from the array. Data size is specified by len_in_bytes.

Function Name	slld_PPOp
Purpose	Performs a Page Programming Operation
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
len_in_bytes	number of bytes to program
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function programs location to the specified data. If the data size to program is larger than PAGE_SIZE this function returns SLLD_ERROR and doe not execute operation. **Note**: If the page boundary is encountered during page programming, additional bytes are wrapped around to the start of the same page. If PPOp is attempted on the protected area, the function returns SLLD_OK but program operation is not executed.

Function Name	slld_PP_4BOp
Purpose	Performs a Page Programming Operation using 4-bytes addressing scheme
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
len_in_bytes	number of bytes to program
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function programs location to the specified data. If the data size to program is larger than PAGE_SIZE this function returns SLLD_ERROR and doe not execute operation. **Note**: If the page boundary is encountered during page programming, additional bytes are wrapped around to the start of the same page. If PP_4BOp is attempted on the protected area, the function returns SLLD_OK but program operation is not executed.

Function Name	slld_QPPOp
Purpose	Performs a Quad input Page Programming Operation
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
len_in_bytes	number of bytes to program
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function programs location to the specified data. If the data size to program is larger than PAGE_SIZE this function returns SLLD_ERROR and doe not execute operation. **Note**: If the page boundary is encountered during page programming, additional bytes are wrapped around to the start of the same page. If QPPOp is attempted on the protected area, the function returns SLLD_OK but program operation is not executed.

Function Name	slld_QPP_4BOp
Purpose	Performs a Quad input Page Programming Operation using 4-bytes addressing scheme
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
len_in_bytes	number of bytes to program
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function programs location to the specified data using 4-bytes addressing scheme. If the data size to program is larger than PAGE_SIZE this function returns SLLD_ERROR and doe not execute operation. **Note**: If the page boundary is encountered during page programming, additional bytes are wrapped around to the start of the same page. If QPP_4BOp is attempted on the protected area, the function returns SLLD_OK but program operation is not executed.

Function Name	slld_BufferedProgramOp
Purpose	Performs a Programming Operation
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
len_in_bytes	number of bytes to program
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function programs location to the specified data. There is no limitation on data size to program. If the page boundary is encountered during BufferedProgramOp, additional bytes are written to the start of the next page. If BufferedProgramOp is attempted on the protected area, the function returns SLLD_OK but program operation on the protected area are not executed.

Function Name	slld_BufferedProgram_4BOp
Purpose	Performs a Programming Operation using 4-bytes addressing scheme
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
len_in_bytes	number of bytes to program
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function programs location to the specified data using 4-bytes addressing scheme. There is no limitation on data size to program. If the page boundary is encountered during BufferedProgram_4BOp, additional bytes are written to the start of the next page. If BufferedProgram_4BOp is attempted on the protected area, the function returns SLLD_OK but program operation on the protected area are not executed.

Function Name	slld_OTPPOp
Purpose	Performs a OTP Programming Operation
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function programs specified data in the OTP region, which is in a different address space from the main array data.

Function Name	slld_SEOp
Purpose	Performs a Sector Erase Operation
Parameters	
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function erases the data in the specified Sector. Function issues all required commands and polls for completion. If SEOp is attempted on the protected area, the function returns SLLD_OK but erase operation is not executed.

Function Name	slld_SE_4BOp
Purpose	Performs a Sector Erase Operation using 4-bytes addressing scheme.
Parameters	
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function erases the data in the specified Sector. Function issues all required commands and polls for completion. If SE_4BOp is attempted on the protected area, the function returns SLLD_OK but erase operation is not executed.

Function Name	slld_P4EOp
Purpose	Performs a 4 KB Parameter Sector Erase Operation
Parameters	
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function erases the data in one of the 4 KB Parameter Sector. Function issues all required commands and polls for completion. If P4EOp is attempted on the protected area, the function returns SLLD_OK but erase operation is not executed.

Function Name	slld_P8EOp
Purpose	Performs a 8 KB Parameter Sector Erase Operation
Parameters	
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function erases the data in two of the 4 KB Parameter Sector. Function issues all required commands and polls for completion. If P8EOp is attempted on the protected area, the function returns SLLD_OK but erase operation is not executed.

Function Name	slld_P8E_4BOp
Purpose	Performs a 8KB Parameter Sector Erase Operation using 4-bytes addressing scheme
Parameters	
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function erases the data in two of the 4 KB Parameter Sector using 4-bytes addressing scheme. Function issues all required commands and polls for completion. If P8E_4BOp is attempted on the protected area, the function returns SLLD_OK but erase operation is not executed.

Function Name	slld_BEOp
Purpose	Performs a Bulk Erase Operation
Parameters	
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function erases the data in the chip. Function issues all required commands and polls for completion. If BEOp is attempted on the protected area, the function returns SLLD_OK but erase operation is not executed.

Function Name	slld_WRSROp
Purpose	Performs a Write Status Register Operation
Parameters	
data_buf	variable containing data to program
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function writes a new value to status register. Function issues all required commands and polls for completion. If WRSROp is attempted with status register (or part of it) protected, the function returns SLLD_OK but status register is not updated.

Function Name	slld_WRROp
Purpose	Writes to Registers.
Parameters	
status_val	variable containing data to program to the status register
config_val	variable containing data to program to the configuration register
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WRR command to SPI Flash. Function issues all required commands and polls for completion. If WRROp is attempted with status/configuration registers protected (or part of them), the function returns SLLD_OK but the registers are not updated.

Function Name	slld_WASPOp
Purpose	Writes to ASP register.
Parameters	
asp_val	variable containing data to program to the ASP register
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WASP command to SPI Flash. Function issues all required commands and polls for completion.

Function Name	slld_WBNKOp
Purpose	Writes to bank addressing register.
Parameters	
bnk_val	variable containing data to program to the bank addressing register
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WBNK command to SPI Flash. Function issues all required commands and polls for completion.

Function Name	slld_WABTOp
Purpose	Writes to Auto Boot register.
Parameters	
abt_val	variable containing data to program to the Auto Boot register
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WABT command to SPI Flash. Function issues all required commands and polls for completion.

Function Name	slld_WPWDOp
Purpose	Writes to the password.
Parameters	
target	variable containing data to program to the password
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WPWD command to SPI Flash. Function issues all required commands and polls for completion.

Function Name	slld_BlockProtectOp
Purpose	Performs a Block Protect Operation
Parameters	
bpb_value	value of block protect bits. Valid value for bpb_value is: 0x0 - 0x3 : for S25FL001D, S25FL002D 0x0 - 0x7 : for S25FL004D *For detail, please refer to data sheet of target device.
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	Function sets Block Protect bits to protect specified memory area. Function issues all required commands and polls for completion.

Function Name	slld_PPB_PGOp
Purpose	Performs a PPB Programming Operation
Parameters	
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function programs the PPB at the specified location and polls for completion.

Function Name	slld_DYB_PGOp
Purpose	Performs a DYB Programming Operation
Parameters	
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function programs the DYB at the specified location and polls for completion.

7.2 Command Functions

Function Name	slld_ReadCmd
Purpose	Reads from SPI Flash.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Read command to SPI Flash and reads data from the array. Data size is specified by len_in_bytes.

Function Name	slld_Read_4BCmd
Purpose	Reads from SPI Flash using 4-bytes addressing scheme.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Read command to SPI Flash and reads data from the array using 4-bytes addressing scheme. Data size is specified by len_in_bytes.

Function Name	slld_Fast_ReadCmd
Purpose	Fast_Read from SPI Flash.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Fast_Read command to SPI Flash and reads data from the array. Data size is specified by len_in_bytes.

Function Name	slld_Fast_Read_4BCmd
Purpose	Fast_Read from SPI Flash using 4-bytes addressing scheme.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Fast_Read command to SPI Flash and reads data from the array using 4-bytes addressing scheme. Data size is specified by len_in_bytes.

Function Name	slld_DualIOReadCmd
Purpose	Reads from SPI Flash in Dual I/O mode.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Dual I/O Read command to SPI Flash and reads data from the array. Data size is specified by len_in_bytes.

Function Name	slld_DualIORead_4BCmd
Purpose	Reads from SPI Flash in Dual I/O mode using 4-bytes addressing scheme.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Dual I/O Read command to SPI Flash and reads data from the array using 4-bytes addressing scheme. Data size is specified by len_in_bytes.

Function Name	slld_DualIOHPReadCmd
Purpose	Reads from SPI Flash in Dual I/O high performance mode.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
mode	read mode
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Dual I/O High Performance Read command to SPI Flash and reads data from the array. Data size is specified by len_in_bytes.

Function Name	slld_DualIOHPRead_4BCmd
Purpose	Reads from SPI Flash in Dual I/O high performance mode using 4-bytes addressing scheme.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
mode	read mode
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Dual I/O High Performance Read command to SPI Flash and reads data from the array using 4-bytes addressing scheme. Data size is specified by len_in_bytes.

Function Name	slld_QuadIOReadCmd
Purpose	Reads from SPI Flash in Quad I/O mode.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Quad I/O Read command to SPI Flash and reads data from the array. Data size is specified by len_in_bytes.

Function Name	slld_QuadIORead_4BCmd
Purpose	Reads from SPI Flash in Quad I/O mode using 4-bytes addressing scheme.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Quad I/O Read command to SPI Flash and reads data from the array using 4-bytes addressing scheme. Data size is specified by len_in_bytes.

Function Name	slld_QuadIOHPReadCmd
Purpose	Reads from SPI Flash in Quad I/O high performance mode.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
mode	read mode
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Quad I/O High Performance Read command to SPI Flash and reads data from the array. Data size is specified by len_in_bytes.

Function Name	slld_QuadIOHPRead_4BCmd
Purpose	Reads from SPI Flash in Quad I/O high performance mode using 4-bytes addressing scheme.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
mode	read mode
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Quad I/O High Performance Read command to SPI Flash and reads data from the array using 4-bytes addressing scheme. Data size is specified by len_in_bytes.

Function Name	slld_Read_IDCmd
Purpose	Reads ID from SPI Flash
Parameters	
target	variable in which to store read data
Return Values	SLLD_OK or SLLD_E_HAL_ERROR
Details	This function issues the Read_ID command to SPI Flash and reads the device ID

Function Name	slld_RDIDCmd
Purpose	Reads Identification from SPI flash
Parameters	
target	variable in which to store read data
Return Values	SLLD_OK or SLLD_E_HAL_ERROR
Details	This function issues the RDID command to SPI Flash and reads the identification. Note the identification consists of 3 bytes (manufacturer identification byte, memory type byte and memory capacity byte.)

Function Name	slld_Read_IdentificationCmd
Purpose	Reads Identification from SPI flash
Parameters	
target	variable in which to store read data
addr	address offset for the command
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the Read-ID command to SPI Flash and reads the identification. Note the identification consists of 2 bytes (manufacturer identification byte, and Device ID.)

Function Name	slld_RDSRCmd
Purpose	Reads from Status Register.
Parameters	
target	variable in which to store read data
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the RDSR command to SPI Flash and reads the value of status register.

Function Name	slld_SRSTCmd
Purpose	Writes software reset to the flash.
Parameters	
~~Void~~	
Return Values	SLLD_OK or SLLD_E_HAL_ERROR
Details	This function issues the SRST command to the SPI Flash.

Function Name	slld_RASPCmd
Purpose	Reads from ASP Register.
Parameters	
target	variable in which to store read data
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the RASP command to SPI Flash and reads the value of the ASP register.

Function Name	slld_RBNKCmd
Purpose	Reads from bank addressing Register.
Parameters	
target	variable in which to store read data
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the RBNK command to SPI Flash and reads the value of the bank addressing register.

Function Name	slld_RABTCmd
Purpose	Reads from Auto Boot Register.
Parameters	
target	variable in which to store read data
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the RABT command to SPI Flash and reads the value of the Auto Boot register.

Function Name	slld_RECCCmd
Purpose	Reads from ECC Register.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the RECC command to SPI Flash and reads the value of the ECC register per cache-line.

Function Name:	slld_RPWDCmd
Purpose:	Reads the Password.
Parameters:	
target	variable in which to store read data
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the RPWD command to SPI Flash and reads the value of the password.

Function Name	slld_RCRCmd
Purpose	Reads from Configuration Register.
Parameters	
target	variable in which to store read data
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the RCR command to SPI Flash and reads the value of configuration register.

Function Name	slld_WRENCmd
Purpose	Writes the Write Enable command.
Parameters	
~~Void~~	
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WREN command to SPI Flash.

Function Name	slld_WRDICmd
Purpose	Writes the Write Disable command.
Parameters	
~~Void~~	
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WRDI command to SPI Flash.

Function Name	slld_WRSRCmd
Purpose	Write to Status Register.
Parameters	
data_buf	variable containing data to program
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WRSR command to SPI Flash.

Function Name	slld_WRRCmd
Purpose	Writes to Registers.
Parameters	
status_val	variable containing data to program to the status register
config_val	variable containing data to program to the configuration register
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WRR command to SPI Flash.

Function Name	slld_WASPCmd
Purpose	Writes to ASP register.
Parameters	
asp_val	variable containing data to program to the ASP register
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WASP command to SPI Flash.

Function Name	slld_WBNKCmd
Purpose	Writes to bank addressing register.
Parameters	
bnk_val	variable containing data to program to the bank addressing register
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WBNK command to SPI Flash.

Function Name	slld_WABTCmd
Purpose	Writes to Auto Boot register.
Parameters	
abt_val	variable containing data to program to the Auto Boot register
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WABT command to SPI Flash.

Function Name	slld_WPWDCmd
Purpose	Writes to password.
Parameters	
target	variable containing data to program to the password
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the WPWD command to SPI Flash.

Function Name	slld_PPCmd
Purpose	Page Program.
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
len_in_bytes	number of bytes to program
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the PP command to SPI Flash.

Function Name	slld_PP_4BCmd
Purpose	Page Program using 4-bytes addressing scheme.
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
len_in_bytes	number of bytes to program
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the PP command to SPI Flash using 4-bytes addressing scheme.

Function Name	slld_QPPCmd
Purpose	Quad I/O Page Program.
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
len_in_bytes	number of bytes to program
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the QPP command to SPI Flash.

Function Name	slld_QPP_4BCmd
Purpose	Quad I/O Page Program using 4-bytes addressing scheme.
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
len_in_bytes	number of bytes to program
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the QPP command to SPI Flash using 4-bytes addressing scheme.

Function Name	slld_SECmd
Purpose	Sector Erase.
Parameters	
sys_addr	device address given by system
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the SE command to SPI Flash.

Function Name	slld_SE_4BCmd
Purpose	Sector Erase using 4-bytes addressing scheme.
Parameters	
sys_addr	device address given by system
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the SE command to SPI Flash using 4-bytes addressing scheme.

Function Name	slld_ERS_SSPCmd
Purpose	Suspend sector erase operation.
Parameters	
~~Void~~	
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the ERS_SSP command to SPI Flash.

Function Name	slld_ERS_RESCmd
Purpose	Resumes suspended sector erase operation.
Parameters	
~~Void~~	
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the ERS_RES command to SPI Flash.

Function Name	slld_RCVRCmd
Purpose	Initiates recovery mode.
Parameters	
~~Void~~	
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the RCVR command to SPI Flash.

Function Name	slld_RCSPCmd
Purpose	Suspends the Recovery mode.
Parameters	
~~Void~~	
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the RCSP command to SPI Flash.

Function Name	slld_RCRSCmd
Purpose	Resumes the suspended Recovery mode.
Parameters	
~~Void~~	
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the RCRS command to SPI Flash.

Function Name	slld_P4ECmd
Purpose	4KB Parameter Sector Erase.
Parameters	
sys_addr	device address given by system
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the P4E command to SPI Flash.

Function Name	slld_P8ECmd
Purpose	8KB Parameter Sector Erase.
Parameters	
sys_addr	device address given by system
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the P8E command to SPI Flash.

Function Name	slld_P8E_4BCmd
Purpose	8KB Parameter Sector Erase using 4-bytes addressing scheme.
Parameters	
sys_addr	device address given by system
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the P8E command to SPI Flash using 4-bytes addressing scheme.

Function Name	slld_BECmd
Purpose	Bulk (Chip) Erase.
Parameters	
~~Void~~	
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the BE command to SPI Flash.

Function Name	slld_OTPPCmd
Purpose	Programs OTP area.
Parameters	
sys_addr	device address given by system
data_buf	variable containing data to program
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the OTPP command to SPI Flash.

Function Name	slld_OTPRCmd
Purpose	Reads data from OTP region.
Parameters	
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the OTP Read command to SPI Flash and reads data from OTP region. Data size is specified by len_in_bytes.

Function Name	slld_SPCmd
Purpose	Software Protect (Deep power-Down).
Parameters	
~~Void~~	
Return Values	SLLD_OK or SLLD_E_HAL_ERROR
Details	This function issues the DP command to SPI Flash.

Function Name	slld_RESCmd
Purpose	Exit Software Protection mode (Release from Deep Power-Down mode).
Parameters	
~~Void~~	
Return Values	SLLD_OK or SLLD_E_HAL_ERROR
Details	This function issues the RES command to SPI Flash.

Function Name	slld_ClearStatusRegisterCmd
Purpose	Clears the status register.
Parameters	
~~Void~~	
Return Values	SLLD_OK or SLLD_E_HAL_ERROR
Details	This function issues the CLSR command to SPI Flash.

Function Name	slld_PPB_PGCmd
Purpose	Program PPB.
Parameters	
sys_addr	device address given by system
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the program PPB command to SPI Flash.

Function Name	slld_DYB_PGCmd
Purpose	Program DYB.
Parameters	
sys_addr	device address given by system
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function issues the program DYB command to SPI Flash.

7.3 Utility Functions

Function Name	slld_Poll
Purpose	Polls flash device for embedded operation completion.
Parameters	
~~Void~~	
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function polls the flash device to determine when an embedded operation has finished.

Function Name	slld_StatusGet
Purpose	Determines Flash Status
Parameters	
dev_status_ptr	variable to store device status
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function reads the status register of specified device and sets value of dev_status_ptr to the current device status (dev_busy, dev_program_error, dev_erase_error, dev_suspend or dev_not_busy.)

Function Name	slld_SoftwareProtectStatusGet
Purpose	Gets Flash Software protect Status.
Parameters	
~~dev_~~softwareprotect	
_status_ptr	variable to store device software protect status
Return Values	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
Details	This function gets the software protect status of the specified device from the variable stored in RAM.

7.4 HAL Functions

Function Name	FLASH_READ
Purpose	Basic device read (one CS# cycle)
Parameters	
command	Command byte to be written to the flash
sys_addr	System address to be read from
Data_buffer	Pointer to a data buffer where the read data will be stored
Number_Of_Bytes	Number of bytes to read
Return Values	SLLD_OK or SLLD_E_HAL_ERROR
Details	A call to the read function corresponds to a CS# cycle. The source code of this function must be adapted according to the system platform requirements.

Function Name	FLASH_WRITE
Purpose	Basic device write (one CS# cycle)
Parameters	
command	Command byte to be written to the flash
sys_addr	System address to be written to
Data_buffer	Pointer to a data buffer containing data to be written
Number_Of_Bytes	Number of bytes to write
Return Values	SLLD_OK or SLLD_E_HAL_ERROR
Details	A call to the write function corresponds to a CS# cycle. The source code of this function must be adapted according to the system platform requirements.

8. Revision History

Section	Description
Revision 01 (October 5, 2010)	
	Initial revision
Revision 02 (December 8, 2011)	
General	Changed from Release 9.3.0 to Release Version 11.4.1
Files	Updated source code

Colophon

The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for any use that includes fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for any use where chance of failure is intolerable (i.e., submersible repeater and artificial satellite). Please note that Spansion will not be liable to you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products. Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the US Export Administration Regulations or the applicable laws of any other country, the prior authorization by the respective government entity will be required for export of those products.

Trademarks and Notice

The contents of this document are subject to change without notice. This document may contain information on a Spansion product under development by Spansion. Spansion reserves the right to change or discontinue work on any product without notice. The information in this document is provided as is without warranty or guarantee of any kind as to its accuracy, completeness, operability, fitness for particular purpose, merchantability, non-infringement of third-party rights, or any other warranty, express, implied, or statutory. Spansion assumes no liability for any damages of any kind arising out of the use of the information in this document.