



## SPI Low Level Drivers Guide

Doc. No. 002-00997 Rev. \*C

Cypress Semiconductor  
198 Champion Court  
San Jose, CA 95134-1709  
Phone (USA): 800.858.1810  
Phone (Intl): +1.408.943.2600  
<http://www.cypress.com>

## Copyrights

© Cypress Semiconductor Corporation, 2006-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you under its copyright rights in the Software, a personal, non-exclusive, nontransferable license (without the right to sublicense) (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units. Cypress also grants you a personal, non-exclusive, nontransferable, license (without the right to sublicense) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely to the minimum extent that is necessary for you to exercise your rights under the copyright license granted in the previous sentence. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and Company shall and hereby does release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. Company shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit [cypress.com](http://cypress.com). Other names and brands may be claimed as property of their respective owners.

# Contents



<b>Contents .....</b>	<b>3</b>
<b>1. Overview .....</b>	<b>4</b>
1.1 Function Categories .....	4
1.2 Parameter Description .....	4
1.3 Data Types .....	5
1.4 Functions Returned Value Description .....	5
1.5 Hardware Abstraction Layer (HAL) .....	6
<b>2. Functions List .....</b>	<b>7</b>
2.1 Operation Functions .....	8
2.2 Command Functions .....	9
2.3 Utility Functions .....	10
2.4 Files .....	10
2.5 Porting Considerations .....	11
2.6 Typedefs .....	11
<b>3. Appendix – API Details .....</b>	<b>12</b>
3.1 Operation Functions .....	12
3.2 Command Functions .....	27
3.3 Utility Functions .....	61
3.4 HAL Functions .....	63
<b>Revision History .....</b>	<b>64</b>

# 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. Refer to [Section 2.5, Porting Considerations on page 12](#) for more details regarding the SLLD system integration.

## 1.1 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: `sll_d_xxxxOp`

For example, `sll_d_PPOp`

This function implements Page Programming operation

### ■ Command Functions:

These functions send flash command sequences to the device.

The command function naming convention is: `sll_d_xxxxCmd`

For example, `sll_d_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.

For example, `sll_d_StatusGet`

This function determines the flash device status and returns the information to the caller (for example, device is busy or device is not busy).

## 1.2 Parameter Description

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

Parameter	Description
<code>device_num</code>	device number, this parameter will be used on device with multiple chip select.
<code>sys_addr</code>	device address given by system: must be 32 bits
<code>source</code>	a single byte to write to flash
<code>target</code>	variable in which to store read data

Parameter	Description
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

### 1.3 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_SOFTWARE_PROTECTED,
                  SLLD_E_HAL_ERROR,
                  SLLD_ERROR.
```

### 1.4 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., `sllid_ppop`, `sllid_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 the software protect mode.

SLLD\_E\_HAL\_ERROR Error occurs during HAL function.

SLLD\_ERROR Error occurs during operation function.

## 1.5 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 device\_num, /\* device number \*/

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 in 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.

## 2. Functions List



For the details, refer to [Appendix – API Details chapter on page 12](#).

### 2.1 Operation Functions

Command	Description
<a href="#">slll_WriteOp</a>	Performs a Single / Quad Page Programming Operation
<a href="#">slll_ReadOp</a>	Performs a Single / Fast / Dual / Quad Read Operation
<a href="#">slll_PPOp</a>	Performs a Page Programming Operation
<a href="#">slll_PP_4BOP</a>	Performs a Page Programming Operation using 4-bytes addressing scheme
<a href="#">slll_QPPOp</a>	Performs a Quad input Page Programming Operation
<a href="#">slll_QPP_4BOP</a>	Performs a Quad input Page Programming Operation using 4-bytes addressing scheme
<a href="#">slll_BufferedProgramOp</a>	Performs a Programming Operation. Unlike <a href="#">slll_PPOp</a> , this function enables program operation over page boundary
<a href="#">slll_BufferedProgram_4BOP</a>	Performs a Programming Operation using 4-bytes addressing scheme. Unlike <a href="#">slll_PP_4BOP</a> , this function enables program operation over page boundary
<a href="#">slll_OTPPOp</a>	Performs a OTP Programming Operation
<a href="#">slll_SEOp</a>	Performs a Sector Erase Operation
<a href="#">slll_SE_4BOP</a>	Performs a Sector Erase Operation using 4-bytes addressing scheme
<a href="#">slll_P4EOp</a>	Performs a Parameter sector Erase Operation. This function erases one of the 4 KB sectors
<a href="#">slll_P8EOp</a>	Performs a Parameter sector Erase Operation. This function erases two of the 4 KB sectors
<a href="#">slll_P8E_4BOP</a>	Performs a Parameter sector Erase Operation using 4-bytes addressing scheme. This function erases two of the 4 KB sectors
<a href="#">slll_BEOp</a>	Performs a Bulk Erase Operation
<a href="#">slll_WRSROp</a>	Performs a Write Status Register Operation
<a href="#">slll_WRROp</a>	Writes a Write Registers Command Sequence to Flash Device
<a href="#">slll_WASPOp</a>	Writes a Write ASP Command Sequence to Flash Device
<a href="#">slll_WBNKOp</a>	Writes a Write bank addressing Command Sequence to Flash Device
<a href="#">slll_WABTOp</a>	Writes a Write Auto Boot Command Sequence to Flash Device
<a href="#">slll_WPWDOp</a>	Writes a Write password Command Sequence to Flash Device
<a href="#">slll_BlockProtectOp</a>	Performs a Block Protect Operation
<a href="#">slll_PPB_PGOp</a>	Performs a PPB programming Operation
<a href="#">slll_DYB_PGOp</a>	Performs a DYB programming Operation
<a href="#">slll_BE32KBOp</a>	Performs a 32KB Block Erase Operation, used for FLK devices
<a href="#">slll_WRAR_Op</a>	Performs a Write Any Register Operation
<a href="#">slll_PPBP_Op</a>	Performs a PPB programming Operation for FSS devices
<a href="#">slll_DYBWR_Op</a>	Performs a DYB Write Operation for FSS devices

Command	Description
<a href="#">sllid_IRPPOP</a>	Performs a Write IRP Register Operation
<a href="#">sllid_SECRPOP</a>	Performs a Secure Region Programming Operation
<a href="#">sllid_SECREOp</a>	Performs a Secure Region Erase Operation
<a href="#">sllid_HBEOp</a>	Performs a Half Block Erase Operation
<a href="#">sllid_CEOp</a>	Performs a Chip Erase Operation
<a href="#">sllid_CE1Op</a>	Performs an alternative Chip Erase Operation
<a href="#">sllid_IBLOp</a>	Performs a IBL lock Operation
<a href="#">sllid_IBULOp</a>	Performs a IBL unlock Operation
<a href="#">sllid_GBLOp</a>	Performs a GBL lock Operation
<a href="#">sllid_GBULOp</a>	Performs a GBL unlock Operation
<a href="#">sllid_SPRPOP</a>	Performs Set Pointer Region Protection Operation
<a href="#">sllid_BlockEraseOp</a>	Performs Block Erase Operation

## 2.2 Command Functions

All Command Functions except [sllid\\_Read\\_IDCmd](#), [sllid\\_SPCmd](#) and [sllid\\_RESCmd](#) check the software protect status of the target device (this information is stored in RAM) before issuing command sequences.

Command	Description
<a href="#">sllid_ReadCmd</a>	Writes a Read Command to Flash Device and reads data
<a href="#">sllid_Read_4BCmd</a>	Writes a Read Command to Flash Device and reads data using 4-bytes addressing scheme
<a href="#">sllid_Fast_ReadCmd</a>	Writes a Flash Read Command Sequence to Flash Device and read data
<a href="#">sllid_Fast_Read_4BCmd</a>	Writes a Flash Read Command Sequence to Flash Device and reads data using 4-bytes addressing scheme
<a href="#">sllid_DualIOReadCmd</a>	Writes a Dual I/O Read Command to Flash Device and reads data
<a href="#">sllid_DualIORead_4BCmd</a>	Writes a Dual I/O Read Command to Flash Device and reads data using 4-bytes addressing scheme
<a href="#">sllid_DualIOHPReadCmd</a>	Writes a Dual I/O High Performance Read Command to Flash Device and reads data
<a href="#">sllid_DualIOHPRead_4BCmd</a>	Writes a Dual I/O High Performance Read Command to Flash Device and read data using 4-bytes addressing scheme
<a href="#">sllid_QuadIOReadCmd</a>	Writes a Quad I/O Read Command to Flash Device and reads data
<a href="#">sllid_QuadIORead_4BCmd</a>	Writes a Quad I/O Read Command to Flash Device and reads data using 4-bytes addressing scheme
<a href="#">sllid_QuadIOHPReadCmd</a>	Writes a Quad I/O High Performance Read Command to Flash Device and reads data
<a href="#">sllid_QuadIOHPRead_4BCmd</a>	Writes a Quad I/O High Performance Read Command to Flash Device and reads data using 4-bytes addressing scheme
<a href="#">sllid_Read_IDCmd</a>	Writes a Read ID Command Sequence to Flash Device and reads Device_ID
<a href="#">sllid_RDIDCmd</a>	Writes a RDID Command Sequence to Flash Device and reads Device_ID
<a href="#">sllid_Read_IdentificationCmd</a>	Writes a Read Electronic ID Command Sequence to Flash Device and reads Device_ID
<a href="#">sllid_RDSRCmd</a>	Writes a Read from Status Register Command Sequence to Flash Device and reads status register
<a href="#">sllid_SRSTCmd</a>	Writes the software reset command to the flash device
<a href="#">sllid_RASPCmd</a>	Writes a Read from ASP Register Command Sequence to Flash Device and reads ASP register
<a href="#">sllid_RBNKCmd</a>	Writes a Read from Bank Addressing Register Command Sequence to Flash Device and reads the bank addressing register
<a href="#">sllid_RABTCmd</a>	Writes a Read from Auto Boot Register Command Sequence to Flash Device and reads Auto Boot register



Command	Description
<i>slld_RECCCmd</i>	Writes a Read from ECC Register Command Sequence to Flash Device and reads ECC register
<i>slld_RPWDCmd</i>	Writes a Read from password Command Sequence to Flash Device and reads the password
<i>slld_RCRCmd</i>	Writes a Read Configuration Register Command Sequence to Flash Device and reads configuration register
<i>slld_WRENCmd</i>	Writes a Write Enable Command Sequence to Flash Device
<i>slld_WRDICmd</i>	Writes a Write Disable Command Sequence to Flash Device
<i>slld_WRSRCmd</i>	Writes a Write Status Register Command Sequence to Flash Device
<i>slld_WRRCmd</i>	Writes a Write Registers Command Sequence to Flash Device
<i>slld_WASPCmd</i>	Writes a Write ASP register Command Sequence to Flash Device
<i>slld_WBNKCmd</i>	Writes a Write bank addressing Command Sequence to Flash Device
<i>slld_WABTCmd</i>	Writes a Write Auto Boot Register Command Sequence to Flash Device
<i>slld_WPWDCmd</i>	Writes a write password Command Sequence to Flash Device
<i>slld_PPCmd</i>	Writes a Page Program Command Sequence to Flash Device
<i>slld_PP_4BCmd</i>	Writes a Page Program Command Sequence to Flash Device using 4-bytes addressing scheme
<i>slld_QPPCmd</i>	Writes a Quad input Page Program Command Sequence to Flash Device
<i>slld_QPP_4BCmd</i>	Writes a Quad input Page Program Command Sequence to Flash Device using 4-bytes addressing scheme
<i>slld_SECmd</i>	Writes a Sector Erase Command Sequence to Flash Device
<i>slld_SE_4BCmd</i>	Writes a Sector Erase Command Sequence to Flash Device using 4-bytes addressing scheme
<i>slld_ERS_SSPCmd</i>	Writes a Sector Erase Suspend command to Flash Device
<i>slld_ERS_RESCmd</i>	Writes a Sector Erase Resume command to Flash Device
<i>slld_RCVRCmd</i>	Writes a Initiate Recovery mode command to Flash Device
<i>slld_RCSPCmd</i>	Writes a Recovery Suspend command to Flash Device
<i>slld_RCRSCmd</i>	Writes a Recovery Resume command to Flash Device
<i>slld_P4ECmd</i>	Writes a 4KB Parameter Sector Erase Command Sequence to Flash Device
<i>slld_P8ECmd</i>	Writes an 8KB Parameter Sector Erase Command Sequence to Flash Device
<i>slld_P8E_4BCmd</i>	Writes an 8 KB Parameter Sector Erase Command Sequence to Flash Device using 4-bytes addressing scheme
<i>slld_BECmd</i>	Writes a Bulk Erase Command Sequence to Flash Device
<i>slld_OTPPCmd</i>	Writes an OTP Program Command Sequence to Flash Device
<i>slld_OTPRCmd</i>	Writes an OTP Read Command Sequence to Flash Device and reads OTP
<i>slld_SPCmd</i>	Writes a Software Protect Command Sequence to Flash Device
<i>slld_RESCmd</i>	Writes a RES Command Sequence to Flash Device
<i>slld_ClearStatusRegisterCmd</i>	Writes a Clear Status Register Command Sequence to Flash Device
<i>slld_PPB_PGCmd</i>	Writes a PPB program Command Sequence to Flash Device
<i>slld_DYB_PGCmd</i>	Writes a DYB program Command Sequence to Flash Device
<i>slld_DPCmd</i>	Writes a Deep Power down Command Sequence to Flash Device
<i>slld_BE32KBCmd</i>	Writes a Block Erase 32KB Command Sequence to Flash Device
<i>slld_WRARCmd</i>	Writes a Write Any Register Command Sequence to Flash Device
<i>slld_PPBP_Cmd</i>	Writes a PPB Program Command Sequence to Flash Device
<i>slld_DYBWR_Cmd</i>	Writes a DYB write Command Sequence to Flash Device
<i>slld_ReadSFDPCmd</i>	Writes a Read SFDPCmd Command Sequence to Flash Device, and read SFDPC values

Command	Description
<a href="#">sllid_RDCR2Cmd</a>	Writes a Read Configuration register 2 Command Sequence to Flash Device, and read register value
<a href="#">sllid_RDCR3Cmd</a>	Writes a Read Configuration register 3 Command Sequence to Flash Device, and read register value
<a href="#">sllid_IRPRDCmd</a>	Writes a Read IRP register Command Sequence to Flash Device, and read register value
<a href="#">sllid_IRPPCmd</a>	Writes a Write IRP register Command Sequence to Flash Device
<a href="#">sllid_QPIENCmd</a>	Writes a Enter QPI Command Sequence to Flash Device
<a href="#">sllid_QPIEXCmd</a>	Writes a Exit QPI Command Sequence to Flash Device
<a href="#">sllid_4BENCmd</a>	Writes a Enter 4 Bytes Address Mode Command Sequence to Flash Device
<a href="#">sllid_4BEXCmd</a>	Writes a Exit 4 Bytes Address Mode Command Sequence to Flash Device
<a href="#">sllid_DORCmd</a>	Writes a Dual Output Command Sequence to Flash Device and read data
<a href="#">sllid_4DORCmd</a>	Writes a 4Byte Dual Output Command Sequence to Flash Device and read data
<a href="#">sllid_DIORCmd</a>	Writes a Dual IO Command Sequence to Flash Device and read data
<a href="#">sllid_4DIORCmd</a>	Writes a 4B Dual IO Command Sequence to Flash Device and read data
<a href="#">sllid_QORCmd</a>	Writes a Quad Output Command Sequence to Flash Device and read data
<a href="#">sllid_4QORCmd</a>	Writes a 4B Quad Output Command Sequence to Flash Device and read data
<a href="#">sllid_QIORCmd</a>	Writes a Quad IO Command Sequence to Flash Device and read data
<a href="#">sllid_4QIORCmd</a>	Writes a 4B Quad IO Command Sequence to Flash Device and read data
<a href="#">sllid_DDRQIORCmd</a>	Writes a DDR Quad IO Command Sequence to Flash Device and read data
<a href="#">sllid_4DDRQIORCmd</a>	Writes a 4B DDR Quad IO Command Sequence to Flash Device and read data
<a href="#">sllid_SECRRCmd</a>	Writes a Secure Region read Command Sequence to Flash Device and read secure region data
<a href="#">sllid_SECRPCmd</a>	Writes a Secure Region program Command Sequence to Flash Device
<a href="#">sllid_SECRECmd</a>	Writes a Secure Region erase Command Sequence to Flash Device
<a href="#">sllid_RUIDCmd</a>	Writes a Read UID Command Sequence to Flash Device and read UID value
<a href="#">sllid_WRENVCmd</a>	Writes a Write Enable for Volatile Register Command Sequence to Flash Device
<a href="#">sllid_HBECmd</a>	Writes a Half Block Erase Command Sequence to Flash Device
<a href="#">sllid_4HBECmd</a>	Writes a 4B Half Block Erase Command Sequence to Flash Device
<a href="#">sllid_4BECmd</a>	Writes a 4B Block Erase Command Sequence to Flash Device
<a href="#">sllid_CECmd</a>	Writes a Chip Erase Command Sequence to Flash Device
<a href="#">sllid_CE1Cmd</a>	Writes an alternative Chip Erase Command Sequence to Flash Device
<a href="#">sllid_IBLRDCmd</a>	Writes a IBL read Command Sequence to Flash Device and read IBL bit
<a href="#">sllid_4IBLRDCmd</a>	Writes a 4B IBL read Command Sequence to Flash Device and read IBL bit
<a href="#">sllid_IBLCmd</a>	Writes a IBL lock Command Sequence to Flash Device
<a href="#">sllid_4IBLCmd</a>	Writes a 4B IBL lock Command Sequence to Flash Device
<a href="#">sllid_IBULCmd</a>	Writes a IBL unlock Command Sequence to Flash Device
<a href="#">sllid_4IBULCmd</a>	Writes a 4B IBL unlock Command Sequence to Flash Device
<a href="#">sllid_GBLCmd</a>	Writes a Global IBL lock Command Sequence to Flash Device
<a href="#">sllid_GBULCmd</a>	Writes a Global IBL unlock Command Sequence to Flash Device
<a href="#">sllid_SPRPCmd</a>	Writes a Set Pointer Region Protection Command Sequence to Flash Device
<a href="#">sllid_4SPRPCmd</a>	Writes a 4B Set Pointer Region Protection Command Sequence to Flash Device
<a href="#">sllid_PRLCmd</a>	Writes a Protection Register lock Command Sequence to Flash Device
<a href="#">sllid_PRRDCmd</a>	Writes a Protection Register Read Command Sequence to Flash Device and read data
<a href="#">sllid_SBLCmd</a>	Writes a Set Burst Length Command Sequence to Flash Device

Command	Description
<a href="#">sllid_MBRCmd</a>	Writes Mode Bit Reset Command Sequence to Flash Device
<a href="#">sllid_BlockEraseCmd</a>	Write Block Erase Command Sequence to Flash Device
<a href="#">sllid_RDQIDCmd</a>	Writes a RDQID Command Sequence to Flash Device and reads Device_ID in Quad All mode
<a href="#">sllid_RDSR2Cmd</a>	Writes a Read from Status Register 2 Command Sequence to Flash Device and reads status register 2
<a href="#">sllid_RDARCmd</a>	Writes a Read Any Register Command Sequence to Flash Device and reads register value
<a href="#">sllid_DLPRDCmd</a>	Writes a DLPRD Command Sequence to Flash Device and read data
<a href="#">sllid_PNVDLRCmd</a>	Writes a Program NV Data Learning Register Command Sequence to Flash Device
<a href="#">sllid_WVDLRCmd</a>	Writes a Write Volatile Data Learning Register Command Sequence to Flash Device
<a href="#">sllid_EPS_Cmd</a>	Writes a Sector Erase suspend Command Sequence to Flash Device
<a href="#">sllid_EPR_Cmd</a>	Writes a Sector Erase resume Command Sequence to Flash Device
<a href="#">sllid_4BAM_Cmd</a>	Writes a 4-Byte Address Mode Command Sequence to Flash Device
<a href="#">sllid_PASSRDCmd</a>	Writes a Password Read Command Sequence to Flash Device and read password
<a href="#">sllid_PASSPCmd</a>	Writes a Password Write Command Sequence to Flash Device
<a href="#">sllid_PASSUCmd</a>	Writes a Password Unlock Command Sequence to Flash Device
<a href="#">sllid_RSTCmd</a>	Writes a Software Reset Command Sequence to Flash Device
<a href="#">sllid_RSTENCmd</a>	Writes a Software Reset Enable Command Sequence to Flash Device

## 2.3 Utility Functions

*sllld\_Poll*

Polls flash device for embedded operation completion

*sllld\_StatusGet*

Determines Flash Status

*sllld\_SoftwareProtectStatusGet*

Gets the software protect status from the variable in RAM

*sllld\_GetDevNumFromAddr*

Get device number from address

## 2.4 Files

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

File	Description
sllld.h	header file containing SLLD function prototypes
sllld_hal.h	header file containing HAL function prototypes
sllld_target_specific.h	header file containing code customization macros
sllld.c	C file containing SLLD function definitions
sllld_hal_example.c	C example file containing HAL function definitions

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

## 2.5 Porting Considerations

Cypress'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 `sllld_target_specific.h`, `sllld_hal.h` and `sllld.h` files contain all defines that must be changed when porting the SLLD to your system. You can also find in the `sllld_hal.h` header file the API parameters and the HAL function prototypes.

The `sllld_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 are worried about the code footprint. You should enable the correspondent macro to the device you are using in your system (FL-A, FL-D, FL-P...).

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

## 2.6 Typedefs

The following typedefs, located in `sllld.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 are convenient for your target system, no modification is required.

On the other hand, `FLASH_READ()` and `FLASH_WRITE()` functions in `sllld_hal_example.c` have to be adapted 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 has to be inserted.

## 3. Appendix – API Details



### 3.1 Operation Functions

<b>Function Name</b>	<b>slll_WriteOp</b>
<b>Purpose</b>	Performs a Single / Quad Page Programming Operation
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	<p>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.</p> <p>NOTE: If the page boundary is encountered during page programming, additional bytes are wrapped around to the start of the same page.</p> <p>If WriteOp is attempted on the protected area, the function returns SLLD_OK but program operation is not executed.</p>

<b>Function Name</b>	<b>slll_ReadOp</b>
<b>Purpose</b>	Performs a Single / Fast, Dual / Quad Read Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	<p>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.</p>

<b>Function Name</b>	<b>sllld_PPOp</b>
<b>Purpose</b>	Performs a Page Programming Operation
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	<p>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 does not execute operation.</p> <p><b>Note:</b> If the page boundary is encountered during page programming, additional bytes are wrapped around to the start of the same page.</p> <p>If PPOp is attempted on the protected area, the function returns SLLD_OK but program operation is not executed.</p>

<b>Function Name</b>	<b>sllld_PP_4BOP</b>
<b>Purpose</b>	Performs a Page Programming Operation using 4-bytes addressing scheme
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	<p>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 does not execute operation.</p> <p><b>Note:</b> If the page boundary is encountered during page programming, additional bytes are wrapped around to the start of the same page.</p> <p>If PP_4BOP is attempted on the protected area, the function returns SLLD_OK but program operation is not executed.</p>

<b>Function Name</b>	<b>slld_QPPOp</b>
<b>Purpose</b>	Performs a Quad input Page Programming Operation
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	<p>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 does not execute operation.</p> <p><b>Note:</b> If the page boundary is encountered during page programming, additional bytes are wrapped around to the start of the same page.</p> <p>If QPPOp is attempted on the protected area, the function returns SLLD_OK but program operation is not executed.</p>

<b>Function Name</b>	<b>slld_QPP_4BOP</b>
<b>Purpose</b>	Performs a Quad input Page Programming Operation using 4-bytes addressing scheme
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	<p>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 does not execute operation.</p> <p><b>Note:</b> If the page boundary is encountered during page programming, additional bytes are wrapped around to the start of the same page.</p> <p>If QPP_4BOP is attempted on the protected area, the function returns SLLD_OK but program operation is not executed.</p>

<b>Function Name</b>	<b>sllid_BufferedProgramOp</b>
<b>Purpose</b>	Performs a Programming Operation
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllid_BufferedProgram_4BOp</b>
<b>Purpose</b>	Performs a Programming Operation using 4-bytes addressing scheme
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllid_OTPPop</b>
<b>Purpose</b>	Performs a OTP Programming Operation
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function programs specified data in the OTP region, which is in a different address space from the main array data.



<b>Function Name</b>	<b>sllid_SEOp</b>
<b>Purpose</b>	Performs a Sector Erase Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllid_SE_4BOp</b>
<b>Purpose</b>	Performs a Sector Erase Operation using 4-bytes addressing scheme.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllid_P4EOp</b>
<b>Purpose</b>	Performs a 4 KB Parameter Sector Erase Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllid_P8EOp</b>
<b>Purpose</b>	Performs a 8 KB Parameter Sector Erase Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllid_P8E_4BOp</b>
<b>Purpose</b>	Performs a 8KB Parameter Sector Erase Operation using 4-bytes addressing scheme
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllid_BEOp</b>
<b>Purpose</b>	Performs a Bulk Erase Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllld_WRSROp</b>
<b>Purpose</b>	Performs a Write Status Register Operation
<b>Parameters</b>	
device_num	device number
data_buf	variable containing data to program
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllld_WRROp</b>
<b>Purpose</b>	Writes to Registers.
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

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

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

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

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

<b>Function Name</b>	<b>slll_BlockProtectOp</b>
<b>Purpose</b>	Performs a Block Protect Operation
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	Function sets Block Protect bits to protect specified memory area. Function issues all required commands and polls for completion.

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

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

<b>Function Name</b>	<b>sllid_GBLOp</b>
<b>Purpose</b>	Performs a Global IBL lock Operation
<b>Parameters</b>	
device_num	device number
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Global IBL lock command to SPI Flash, and then polls for completion.

<b>Function Name</b>	<b>sllid_GBULOOp</b>
<b>Purpose</b>	Performs a Global IBL unlock Operation
<b>Parameters</b>	
device_num	device number
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Global IBL unlock command to SPI Flash, and then polls for completion

<b>Function Name</b>	<b>sllid_SPRPOp</b>
<b>Purpose</b>	Performs a Set Pointer Region Protection Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the SPRP command to SPI Flash, and then polls for completion.

<b>Function Name</b>	<b>sllid_BlockEraseOp</b>
<b>Purpose</b>	Performs a Block Erase Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Block Erase command to SPI Flash, and then polls for completion.

<b>Function Name</b>	<b>sllid_BE32KBOp</b>
<b>Purpose</b>	Performs a 32KB Block Erase Operation for FLK device
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function sends a 32-KB Block erase command to FLK device and polls for completion.

<b>Function Name</b>	<b>sllid_WRAR_Op</b>
<b>Purpose</b>	Performs a Write any register Operation
<b>Parameters</b>	
device_num	device number
reg_addr	register address given by system
data_buf	variable containing data to program
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the write any register command and write 1-byte.

<b>Function Name</b>	<b>slll_PPBP_Op</b>
<b>Purpose</b>	Performs a PPB program Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the PPB Program command to SPI Flash and programs it, then polls for completion.

<b>Function Name</b>	<b>slll_DYBWR_Op</b>
<b>Purpose</b>	Performs a DYB write Operation
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the DYB write command to SPI Flash and programs it, then polls for completion.

<b>Function Name</b>	<b>slll_IRPPOp</b>
<b>Purpose</b>	Performs a write to IRP register Operation
<b>Parameters</b>	
device_num	device number
irp_val	variable containing data to program to the IRP register
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the write IRP register command to SPI Flash and programs it, then polls for completion.



<b>Function Name</b>	<b>sllid_SECRPOp</b>
<b>Purpose</b>	Performs a Security Region program Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
program_buf	variable containing data to program
len_in_bytes	number of bytes
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the write IRP register command to SPI Flash and programs it, then polls for completion.

<b>Function Name</b>	<b>sllid_SECRPOp</b>
<b>Purpose</b>	Performs a Security Region program Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
program_buf	variable containing data to program
len_in_bytes	number of bytes
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Security Region program command to SPI Flash and programs it, then polls for completion.

<b>Function Name</b>	<b>sllid_SECREOp</b>
<b>Purpose</b>	Performs a Security Region erase Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Secure Region erase command to SPI Flash, and then polls for completion.

<b>Function Name</b>	<b>sllid_HBEOp</b>
<b>Purpose</b>	Performs a Half Block erase Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Half Block erase command to SPI Flash, and then polls for completion.

<b>Function Name</b>	<b>sllid_CEOp</b>
<b>Purpose</b>	Performs a Chip erase Operation
<b>Parameters</b>	
device_num	device number
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Chip erase command to SPI Flash, and then polls for completion.

<b>Function Name</b>	<b>sllid_CE1Op</b>
<b>Purpose</b>	Performs an Chip erase Operation with alternative instruction
<b>Parameters</b>	
device_num	device number
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Chip erase command to SPI Flash, and then polls for completion.

<b>Function Name</b>	<b>slld_IBLOp</b>
<b>Purpose</b>	Performs a IBL lock Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the IBL lock command to SPI Flash, and then polls for completion.

<b>Function Name</b>	<b>slld_IBULOp</b>
<b>Purpose</b>	Performs a IBL unlock Operation
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
dev_status_ptr	Pointer to the device status value after polling end
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the IBL unlock command to SPI Flash, and then polls for completion.

## 3.2 Command Functions

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

<b>Function Name</b>	<b>sllid_Read_4BCmd</b>
<b>Purpose</b>	Reads from SPI Flash using 4-bytes addressing scheme.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

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

<b>Function Name</b>	<b>sllid_Fast_Read_4BCmd</b>
<b>Purpose</b>	Fast_Read from SPI Flash using 4-bytes addressing scheme.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllid_DualIOReadCmd</b>
<b>Purpose</b>	Reads from SPI Flash in Dual I/O mode.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllid_DualIORead_4BCmd</b>
<b>Purpose</b>	Reads from SPI Flash in Dual I/O mode using 4-bytes addressing scheme.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllld_DualIOHPReadCmd</b>
<b>Purpose</b>	Reads from SPI Flash in Dual I/O high performance mode.
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllld_DualIOHPRead_4BCmd</b>
<b>Purpose</b>	Reads from SPI Flash in Dual I/O high performance mode using 4-bytes addressing scheme.
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>sllld_QuadIOReadCmd</b>
<b>Purpose</b>	Reads from SPI Flash in Quad I/O mode.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>slll_QuadIORead_4BCmd</b>
<b>Purpose</b>	Reads from SPI Flash in Quad I/O mode using 4-bytes addressing scheme.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>slll_QuadIOHPReadCmd</b>
<b>Purpose</b>	Reads from SPI Flash in Quad I/O high performance mode.
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>slll_QuadIOHPRead_4BCmd</b>
<b>Purpose</b>	Reads from SPI Flash in Quad I/O high performance mode using 4-bytes addressing scheme.
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.

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

<b>Function Name</b>	<b>slll_RDIDCmd</b>
<b>Purpose</b>	Reads Identification from SPI flash
<b>Parameters</b>	
device_num	device number
target	variable in which to store read data
<b>Return Values</b>	SLLD_OK or SLLD_E_HAL_ERROR
<b>Details</b>	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.)

<b>Function Name</b>	<b>slll_Read_IdentificationCmd</b>
<b>Purpose</b>	Reads Identification from SPI flash
<b>Parameters</b>	
device_num	device number
target	variable in which to store read data
addr	address offset for the command
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.)



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

<b>Function Name</b>	<b>slld_SRSTCmd</b>
<b>Purpose</b>	Writes software reset to the flash.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the SRST command to the SPI Flash.

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

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

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

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

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

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

<b>Function Name</b>	<b>slld_WRENCmd</b>
<b>Purpose</b>	Writes the Write Enable command.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the WREN command to SPI Flash.

<b>Function Name</b>	<b>slld_WRDICmd</b>
<b>Purpose</b>	Writes the Write Disable command.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the WRDI command to SPI Flash.

<b>Function Name</b>	<b>slld_WSRCmd</b>
<b>Purpose</b>	Write to Status Register.
<b>Parameters</b>	
device_num	device number
data_buf	variable containing data to program
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the WRSR command to SPI Flash.

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

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

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

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

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

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

<b>Function Name</b>	<b>sllid_PP_4BCmd</b>
<b>Purpose</b>	Page Program using 4-bytes addressing scheme.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
data_buf	variable containing data to program
len_in_bytes	number of bytes to program
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the PP command to SPI Flash using 4-bytes addressing scheme.

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

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

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

<b>Function Name</b>	<b>slld_SE_4BCmd</b>
<b>Purpose</b>	Sector Erase using 4-bytes addressing scheme.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system

<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the SE command to SPI Flash using 4-bytes addressing scheme.

<b>Function Name</b>	<b>sllid_ERS_SSPCmd</b>
<b>Purpose</b>	Suspend sector erase operation.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the ERS_SSP command to SPI Flash.

<b>Function Name</b>	<b>sllid_ERS_RESCmd</b>
<b>Purpose</b>	Resumes suspended sector erase operation.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the ERS_RES command to SPI Flash.

<b>Function Name</b>	<b>sllid_RCVRCmd</b>
<b>Purpose</b>	Initiates recovery mode.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the RCVR command to SPI Flash.

<b>Function Name</b>	<b>sllid_RCSPCmd</b>
<b>Purpose</b>	Suspends the Recovery mode.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the RCSP command to SPI Flash.



<b>Function Name</b>	<b>sllid_RCRSCmd</b>
<b>Purpose</b>	Resumes the suspended Recovery mode.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the RCRS command to SPI Flash.

<b>Function Name</b>	<b>sllid_P4ECmd</b>
<b>Purpose</b>	4KB Parameter Sector Erase.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the P4E command to SPI Flash.

<b>Function Name</b>	<b>sllid_P8ECmd</b>
<b>Purpose</b>	8KB Parameter Sector Erase.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the P8E command to SPI Flash.

<b>Function Name</b>	<b>sllid_P8E_4BCmd</b>
<b>Purpose</b>	8KB Parameter Sector Erase using 4-bytes addressing scheme.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the P8E command to SPI Flash using 4-bytes addressing scheme.

<b>Function Name</b>	<b>sllid_BECmd</b>
<b>Purpose</b>	Bulk (Chip) Erase.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the BE command to SPI Flash.

<b>Function Name</b>	<b>sllid_OTPPCmd</b>
<b>Purpose</b>	Programs OTP area.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
data_buf	variable containing data to program
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the OTP command to SPI Flash.

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

<b>Function Name</b>	<b>sllid_SPCmd</b>
<b>Purpose</b>	Software Protect (Deep power-Down).
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK or SLLD_E_HAL_ERROR

<b>Details</b>	This function issues the DP command to SPI Flash.

<b>Function Name</b>	<b>sllid_RESCmd</b>
<b>Purpose</b>	Exit Software Protection mode (Release from Deep Power-Down mode).
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the RES command to SPI Flash.

<b>Function Name</b>	<b>sllid_ClearStatusRegisterCmd</b>
<b>Purpose</b>	Clears the status register.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the CLSR command to SPI Flash.

<b>Function Name</b>	<b>sllid_PPB_PGCmd</b>
<b>Purpose</b>	Program PPB.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the program PPB command to SPI Flash.

<b>Function Name</b>	<b>sllid_DYB_PGCmd</b>
<b>Purpose</b>	Program DYB.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the program DYB command to SPI Flash.

<b>Function Name</b>	<b>slld_PASSUCmd</b>
<b>Purpose</b>	Password Unlock.
<b>Parameters</b>	
device_num	device number
data_buf	variable containing data to program
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the PASSU command to SPI Flash.

<b>Function Name</b>	<b>slld_RSTCmd</b>
<b>Purpose</b>	Software Reset
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Software Reset command immediately following a RSTEN command initiates the software reset process.

<b>Function Name</b>	<b>slld_RSTENCmd</b>
<b>Purpose</b>	Software Reset Enable
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Reset Enable command which required immediately before a Reset command (RST).

<b>Function Name</b>	<b>slld_DPCmd</b>
<b>Purpose</b>	Deep Power Down
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Deep Power-down command to SPI Flash. This command sets the target device in deep power-down state to reduce power consumption.

<b>Function Name</b>	<b>sllid_PRLCmd</b>
<b>Purpose</b>	Protection register lock (NVLOCK bit write).
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the PRL command to SPI Flash.

<b>Function Name</b>	<b>sllid_PRRDCmd</b>
<b>Purpose</b>	Protection register read.
<b>Parameters</b>	
device_num	device number
data_buf	variable containing data to program
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the PRRD command to SPI Flash.

<b>Function Name</b>	<b>sllid_SBLCmd</b>
<b>Purpose</b>	Set Burst Length.
<b>Parameters</b>	
device_num	device number
wrapbit_buf	variable in which to store wrap bit data
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Set Burst Length command to SPI Flash.

<b>Function Name</b>	<b>sllid_MBRCmd</b>
<b>Purpose</b>	Mode Bit Reset.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the MBR command to return the device from continuous high performance read mode back to normal standby awaiting any new command.

<b>Function Name</b>	<b>slld_BlockEraseCmd</b>
<b>Purpose</b>	Block Erase.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Block Erase command to SPI Flash. The Block Erase command sets all bits in the addressed 64 KB block to 1 (all bytes are FFh).

<b>Function Name</b>	<b>slld_RDQIDCmd</b>
<b>Purpose</b>	Read manufacturer ID in Quad All mode.
<b>Parameters</b>	
device_num	device number
target	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the RDQID command to SPI Flash and read ID data.

<b>Function Name</b>	<b>slld_RDSR2Cmd</b>
<b>Purpose</b>	Read Status Register-2.
<b>Parameters</b>	
device_num	device number
target	variable in which to store read data
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the RDSR2 command to SPI Flash and read status register 2.

<b>Function Name</b>	<b>sllld_RDARCmd</b>
<b>Purpose</b>	Read any device register.
<b>Parameters</b>	
device_num	device number
reg_addr	register address given by system
target	variable in which to store read data
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the read any register command and reads the requested data in 1-byte.

<b>Function Name</b>	<b>sllld_DLPRDCmd</b>
<b>Purpose</b>	Read Data Learning register.
<b>Parameters</b>	
device_num	device number
data_buf	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the read any register command and reads the requested data in 1-byte.

<b>Function Name</b>	<b>sllld_PNVDLRCmd</b>
<b>Purpose</b>	Program NV Data Learning register.
<b>Parameters</b>	
device_num	device number
data_buf	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the PNVDRL command to SPI Flash.

<b>Function Name</b>	<b>sllid_WVDLRCmd</b>
<b>Purpose</b>	Write Volatile Data Learning register.
<b>Parameters</b>	
device_num	device number
data_buf	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the WVDRL command to SPI Flash.

<b>Function Name</b>	<b>sllid_EPS_Cmd</b>
<b>Purpose</b>	Sector Erase suspend.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the EPS command to SPI Flash.

<b>Function Name</b>	<b>sllid_EPR_Cmd</b>
<b>Purpose</b>	Sector Erase resume.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the EPR command to SPI Flash.

<b>Function Name</b>	<b>sllid_4BAM_Cmd</b>
<b>Purpose</b>	4-Byte Address Mode.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the 4BAM command to SPI Flash. Used for FSS devices.



<b>Function Name</b>	<b>sllid_PASSRDCmd</b>
<b>Purpose</b>	Password Read.
<b>Parameters</b>	
device_num	device number
data_buf	variable in which to store read data
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the PASSRD command to SPI Flash and read password data.

<b>Function Name</b>	<b>sllid_PASSPCmd</b>
<b>Purpose</b>	Program Password.
<b>Parameters</b>	
device_num	device number
data_buf	variable in which to store read data
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the PASSP command to SPI Flash.

<b>Function Name</b>	<b>sllid_WRENVCmd</b>
<b>Purpose</b>	Write Enable for Volatile Status and Configure register.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the WRENV command to SPI Flash.

<b>Function Name</b>	<b>sllid_HBECmd</b>
<b>Purpose</b>	Half Block Erase.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the HBE command to SPI Flash.

<b>Function Name</b>	<b>sllid_4HBECmd</b>
<b>Purpose</b>	Half Block Erase 4 bytes addressing scheme.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the 4HBE command to SPI Flash.

<b>Function Name</b>	<b>sllid_4BECmd</b>
<b>Purpose</b>	Block Erase 4 bytes addressing scheme.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the 4BE command to SPI Flash.

<b>Function Name</b>	<b>sllid_CECmd</b>
<b>Purpose</b>	Chip erase.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the CE command to SPI Flash.

<b>Function Name</b>	<b>sllid_CE1Cmd</b>
<b>Purpose</b>	Chip erase using alternate instruction.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the CE1 command to SPI Flash.

<b>Function Name</b>	<b>sIld_IBLRDCmd</b>
<b>Purpose</b>	IBL read.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the reading the state of each IBL bit protection.

<b>Function Name</b>	<b>sIld_4IBLRDCmd</b>
<b>Purpose</b>	IBL read - 4 Bytes address.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the reading the state of each IBL bit protection using 4 Bytes address scheme.

<b>Function Name</b>	<b>sIld_IBLCmd</b>
<b>Purpose</b>	IBL lock.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the IBL lock command to sets the selected IBL bit to "0" protecting each related sector / block.

<b>Function Name</b>	<b>sllid_4IBLCmd</b>
<b>Purpose</b>	IBL lock using 4 Bytes address scheme.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the IBL lock command to sets the selected IBL bit to "0" protecting each related sector / block.

<b>Function Name</b>	<b>sllid_IBULCmd</b>
<b>Purpose</b>	IBL unlock.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the IBL unlock command to sets the selected IBL bit to "1" unprotecting each related sector / block.

<b>Function Name</b>	<b>sllid_4IBLCmd</b>
<b>Purpose</b>	IBL unlock using 4 Bytes address scheme.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the IBL unlock command to sets the selected IBL bit to "1" unprotecting each related sector / block.

<b>Function Name</b>	<b>sllid_GBLCmd</b>
<b>Purpose</b>	Global IBL lock.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR

<b>Details</b>	This function issues the global IBL lock command to sets all IBL bit to "0" protecting all sectors / blocks.

<b>Function Name</b>	<b>sllid_GBULCmd</b>
<b>Purpose</b>	IBL unlock.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the global IBL unlock command to sets all IBL bit to "1" unprotecting all sectors / blocks.

<b>Function Name</b>	<b>sllid_SPRPCmd</b>
<b>Purpose</b>	Set Pointer Region Protection.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the SPRP command to sets pointer region protection.

<b>Function Name</b>	<b>sllid_4SPRPCmd</b>
<b>Purpose</b>	Set Pointer Region Protection - 4 Bytes address
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the 4SPRP command to sets pointer region protection.

<b>Function Name</b>	<b>sllid_DIORCmd</b>
<b>Purpose</b>	Read flash using Dual IO.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system

target	variable in which to store read data
modebit	the read mode to be passed to the device
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the dual IO read command and reads the requested data.

<b>Function Name</b>	<b>sllid_4DIORCmd</b>
<b>Purpose</b>	Read flash using 4 Byte Dual IO.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
modebit	the read mode to be passed to the device
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the 4 byte dual IO read command and reads the requested data.

<b>Function Name</b>	<b>sllid_QIORCmd</b>
<b>Purpose</b>	Read flash using Quad IO.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
modebit	the read mode to be passed to the device
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the quad IO read command and reads the requested data.

<b>Function Name</b>	<b>sllid_4QIORCmd</b>
<b>Purpose</b>	Read flash using 4 Byte Quad IO.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
modebit	the read mode to be passed to the device
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the 4 byte quad IO read command and reads the requested data.

<b>Function Name</b>	<b>sllid_DDRQIORCmd</b>
<b>Purpose</b>	Read flash using DDR Quad IO.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
modebit	the read mode to be passed to the device
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the DDR quad IO read command and reads the requested data.

<b>Function Name</b>	<b>sllid_4DDRQIORCmd</b>
<b>Purpose</b>	Read flash using 4 Byte DDR Quad IO.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
modebit	the read mode to be passed to the device
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the 4 byte DDR quad IO read command and reads the requested data.

<b>Function Name</b>	<b>slld_SECRRCmd</b>
<b>Purpose</b>	Security Region Read.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
read_buf	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the security region read command and reads the requested data.

<b>Function Name</b>	<b>slld_SECRPCmd</b>
<b>Purpose</b>	Security Region program.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
program_buf	data buffer
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the security region program command to SPI flash.

<b>Function Name</b>	<b>slld_SECRECmd</b>
<b>Purpose</b>	Security Region erase.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the security region erase command to SPI flash.



<b>Function Name</b>	<b>sllid_RUIDCmd</b>
<b>Purpose</b>	Read Unique ID number.
<b>Parameters</b>	
device_num	device number
read_buf	variable in which to store read data
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the read UID command and reads the requested data.

<b>Function Name</b>	<b>sllid_BE32KBCmd</b>
<b>Purpose</b>	Block Erase 32KB for FLK device.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the Block Erase 32KB command to SPI Flash.

<b>Function Name</b>	<b>sllid_WRARCmd</b>
<b>Purpose</b>	Write any device register command.
<b>Parameters</b>	
device_num	device number
reg_addr	register address given by system
data_buf	variable containing data to program
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the write any register command to SPI Flash.

<b>Function Name</b>	<b>sllid_PPBP_Cmd</b>
<b>Purpose</b>	PPB program.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the PPB Program command to SPI Flash and programs it. The command is followed by the 24 or 32-bit address, depending on the address length configuration CR2V[7].

<b>Function Name</b>	<b>sllid_DYBWR_Cmd</b>
<b>Purpose</b>	DYB write.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
data_buf	variable containing data to program
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the DYB write command to SPI Flash.

<b>Function Name</b>	<b>sllid_ReadSFDPCmd</b>
<b>Purpose</b>	Read Serial Flash Discoverable Parameter.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
read_buf	data buffer
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the read SFDPCmd command to SPI Flash.

<b>Function Name</b>	<b>slll_RDCR2Cmd</b>
<b>Purpose</b>	Read from configuration register-2.
<b>Parameters</b>	
device_num	device number
target	variable in which to store read data
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the RDCR2 command to SPI Flash.

<b>Function Name</b>	<b>slll_RDCR3Cmd</b>
<b>Purpose</b>	Read from configuration register-3.
<b>Parameters</b>	
device_num	device number
target	variable in which to store read data
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the RDCR3 command to SPI Flash.

<b>Function Name</b>	<b>slll_IRPRDCmd</b>
<b>Purpose</b>	Read IRP Register.
<b>Parameters</b>	
device_num	device number
target	variable in which to store read data
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the IRPRD command to SPI Flash.

<b>Function Name</b>	<b>sllld_IRPPCmd</b>
<b>Purpose</b>	Write to IRP Register.
<b>Parameters</b>	
device_num	device number
target	variable containing data to program to the IRP register
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the IRPP command to SPI Flash.

<b>Function Name</b>	<b>sllld_QPIENCmd</b>
<b>Purpose</b>	Enter QPI
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the QPIEN command to SPI Flash

<b>Function Name</b>	<b>sllld_QPIEXCmd</b>
<b>Purpose</b>	Exit QPI.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the QPIEX command to SPI Flash.

<b>Function Name</b>	<b>sllld_4BENCmd</b>
<b>Purpose</b>	Enter 4 byte Address Mode.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the 4BEN command to SPI Flash. This will sets the volatile Address Length bit (CR2V[0]) to 1 to change most 3-byte address commands to require 4 bytes of address.

<b>Function Name</b>	<b>sllid_4BEXCmd</b>
<b>Purpose</b>	Exit 4 byte Address Mode.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the 4BEX command to SPI Flash. This will sets the volatile Address Length bit (CR2V[0]) to 0.

<b>Function Name</b>	<b>sllid_DORCmd</b>
<b>Purpose</b>	Read flash using Dual Output.
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the dual output read command and reads the requested data.

<b>Function Name</b>	<b>sllid_4DORCmd</b>
<b>Purpose</b>	Read flash using 4 Byte Dual Output
<b>Parameters</b>	
device_num	device number
sys_addr	device address given by system
target	variable in which to store read data
len_in_bytes	number of bytes to read
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function issues the 4 byte dual output read command and reads the requested data.

### 3.3 Utility Functions

<b>Function Name</b>	<b>sllid_Poll</b>
<b>Purpose</b>	Polls flash device for embedded operation completion.
<b>Parameters</b>	
device_num	device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function polls the flash device to determine when an embedded operation has finished.

<b>Function Name</b>	<b>sllid_StatusGet</b>
<b>Purpose</b>	Determines Flash Status
<b>Parameters</b>	
device_num	device number
dev_status_ptr	variable to store device status
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	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.)

<b>Function Name</b>	<b>sllid_SoftwareProtectStatusGet</b>
<b>Purpose</b>	Gets Flash Software protect Status.
<b>Parameters</b>	
device_num	device number
softwareprotect	
dev_status_ptr	variable to store device software protect status
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function gets the software protect status of the specified device from the variable stored in RAM.

<b>Function Name</b>	<b>sllid_GetDevNumFromAddr</b>
<b>Purpose</b>	Helps user to get device number from address when target device with multiple chip select.
<b>Parameters</b>	
sys_addr	device address given by system
device_num	pointer to device number
<b>Return Values</b>	SLLD_OK, SLLD_E_DEVICE_SOFTWARE_PROTECTED or SLLD_E_HAL_ERROR
<b>Details</b>	This function can help the user to get the device number from the address when target device with multiple chip select. Before using this function, user must ensure the target device has multiple chip selects. The user must also modify BASE_ADDR_MASK and BASE_ADDR_SHIFT_BIT according to their own system configuration.

## 3.4 HAL Functions

<b>Function Name</b>	<b>FLASH_READ</b>
<b>Purpose</b>	Basic device read (one CS# cycle)
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK or SLLD_E_HAL_ERROR
<b>Details</b>	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.

<b>Function Name</b>	<b>FLASH_WRITE</b>
<b>Purpose</b>	Basic device write (one CS# cycle)
<b>Parameters</b>	
device_num	device number
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
<b>Return Values</b>	SLLD_OK or SLLD_E_HAL_ERROR
<b>Details</b>	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.



# Revision History



## Document Revision History

Document Title: SPI Low Level Drivers Guide				
Document Number: 002-00997				
Revision	ECN#	Issue Date	Origin of Change	Description of Change
**	–	10/05/2010	–	Initial version
*A	4969108	10/16/2015	MSWI	Updated in Cypress template
*B	5028171	11/26/2015	MSWI	Updated in User Guide template
*C	5211120	04/07/2016	PZHU	Updated API description to support FL-L devices