



SafeFLASH NOR Driver for CFI and SFDP SPI Flash User Guide

Version 2.00

For use with SafeFLASH NOR Driver for CFI and SFDP
SPI Flash versions 1.11 R2 and above

Table of Contents

1. System Overview	3
1.1. Introduction	4
1.2. Feature Check	6
1.3. Fail-safety	7
1.4. Packages and Documents	8
1.5. Change History	9
2. Source File List	10
3. Configuration Options	12
4. PSP Porting	14
4.1. psp_safe_nor_cfi_sfdp_delay	15
4.2. config_ftl_nor_cfi_sfdp.h	16
4.3. QSPI Functions	19
psp_qspi_init	19
psp_qspi_delete	20
psp_qspi_stop	21
psp_qspi_start	22
psp_qspi_register_isr	23
psp_qspi_get_baudrate	24
psp_qspi_set_baudrate	25
psp_qspi_lock	26
psp_qspi_unlock	27
psp_qspi_transfer	28
psp_qspi_wait_status	29
t_qspi_transfer	30
t_qspi_wait_status	30
Transfer Modes	31
5. Version	32

1. System Overview

This chapter contains the fundamental information for this module.

The component sections are as follows:

- [Introduction](#) - describes the main elements of the module.
- [Feature Check](#) - summarizes the main features of the module as bullet points.
- [Fail-safety](#) - defines fail-safety and describes the quality of service that SafeFLASH provides.
- [Packages and Documents](#) - the *Packages* section lists the packages that you need in order to use this module. The *Documents* section lists the relevant user guides.
- [Change History](#) - lists the earlier versions of this manual, giving the software version that each manual describes.

All rights reserved. This document and the associated software are the sole property of HCC Embedded. Reproduction or duplication by any means of any portion of this document without the prior written consent of HCC Embedded is expressly forbidden.

HCC Embedded reserves the right to make changes to this document and to the related software at any time and without notice. The information in this document has been carefully checked for its accuracy; however, HCC Embedded makes no warranty relating to the correctness of this document.

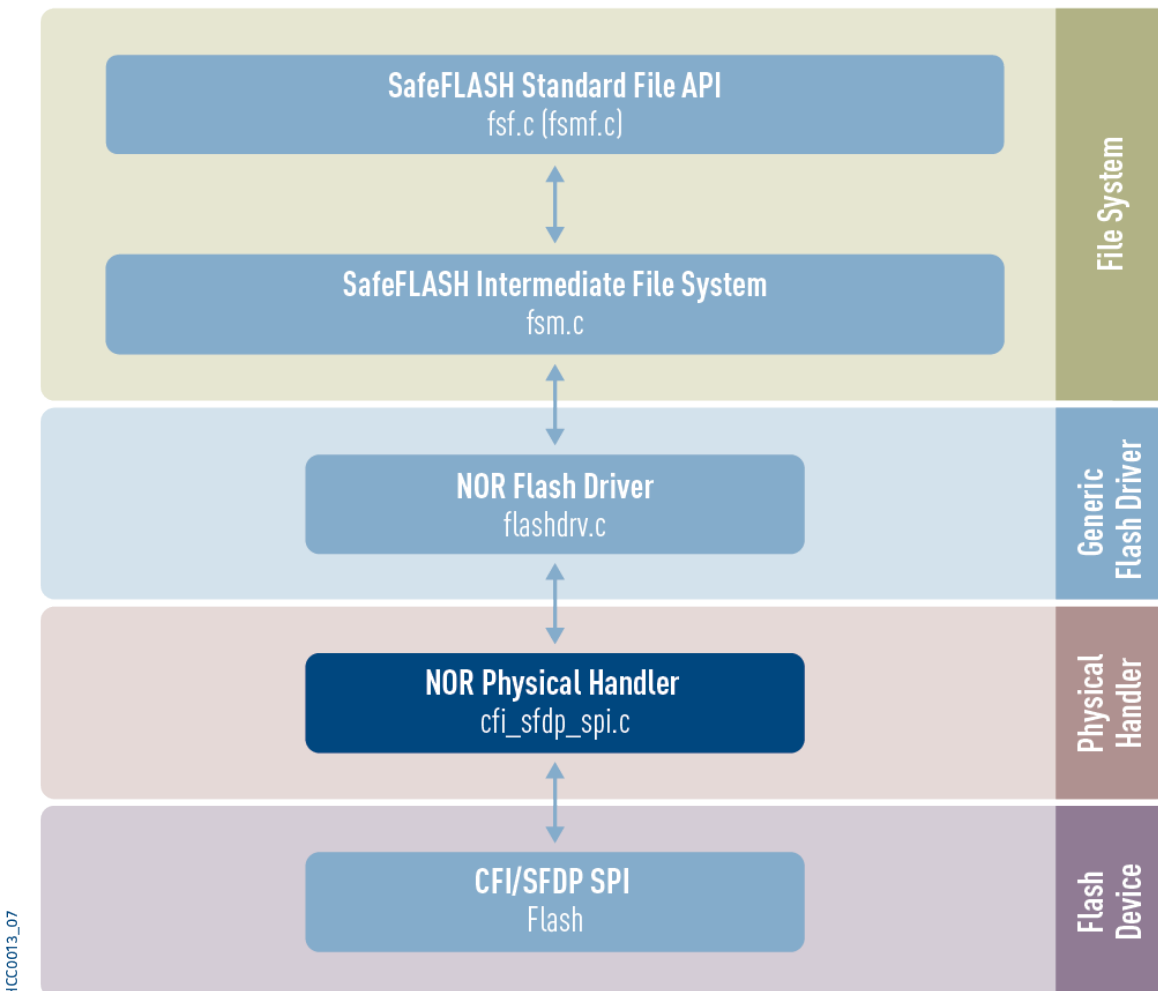
1.1. Introduction

This guide is for those who want to implement a SafeFLASH NOR driver for use with CFI/SFDP SPI flash devices. This is for use with HCC's SafeFLASH file system. These flash devices support the Common Flash Memory Interface (CFI) and Serial Flash Discoverable Parameters (SFDP) standards on a Serial Peripheral Interface (SPI) or Quad Serial Peripheral Interface (QSPI).

CFI and SFDP are standards widely used by flash manufacturers that allow software to obtain detailed information about the internal configuration of the flash device. This driver uses that information to configure the system automatically to use the attached flash type.

The SafeFLASH file system driver design is highly portable while still maintaining excellent performance. The basic device architecture includes a high level driver for each general media type that shares some common properties. This driver handles issues of FAT maintenance, wear leveling, and so on.

The following diagram shows the structure of the file system software:



This diagram shows:

- The main SafeFLASH package - this provides the file API and intermediate file system. This is described in the [HCC SafeFLASH File System User Guide](#).
- The NOR flash driver - the generic device driver for NOR flash, provided by the base NOR package.

This driver handles issues of FAT maintenance, wear leveling, and so on. It is described in the [HCC SafeFLASH File System NOR Drive User Guide](#).

- The NOR physical handler - the NOR device driver for the CFI/SFDP SPI flash. Provided by this module, this performs the translation between the driver and the physical flash hardware. This guide shows how to add this to the build. Using the available sample drivers as a model, you can create a driver that meets your specific needs.
- The CFI/SFDP SPI flash device.

Note: HCC Embedded offers hardware and firmware development consultancy to assist developers with the implementation of flash file systems.

1.2. Feature Check

The main features of the module are the following:

- Conforms to the HCC Advanced Embedded Framework.
- Designed for integration with both RTOS and non-RTOS based systems.
- Supports NOR flash connected by SPI or Quad SPI (QSPI).
- Supports all common NOR devices.
- Can handle any array of NOR flash.
- Supports the Common Flash Memory Interface (CFI) standard for NOR flash devices.
- Supports the Serial Flash Discoverable Parameters (SFDP) standard for serial NOR flash devices.

1.3. Fail-safety

This driver for serial NOR flash is designed as part of HCC's SafeFLASH file system. SafeFLASH guarantees a defined level of fail-safety (see the [SafeFLASH File System User Guide](#)). For the system to be able to guarantee fail-safety, each component must provide a defined quality of service.

For this driver the following must be guaranteed to ensure the system is fail-safe:

- All write operations must be committed to flash in the sequence in which they are provided to the driver.
- Any write operation that fails must return an error.
- Any erase operation that fails must return an error.
- The system must ensure that there is at most one partially complete write or erase operation. At this point the file system should be restarted so that it can be recovered.

To achieve this, the hardware should ensure that, in the event of a falling voltage approaching the specified minimum programming level of the flash, the system either resets or provides a signal to the software to block access to the flash.

An alternative solution is to add capacitance to the design. This must provide sufficient power that, after a hardware error or reset condition is detected, the active operation on the flash can be completed.

Only by using one of these techniques can the system guarantee correct operation even after an unexpected system reset.

1.4. Packages and Documents

Packages

The table below lists the packages that you need in order to use this module:

Package	Description
<code>hcc_base_doc</code>	This contains the two guides that will help you get started.
<code>fs_safe_nor</code>	The SafeFLASH NOR flash driver.
<code>fs_safe_nor_drv_cfi_sfdp_spi</code>	The NOR driver for CFI and SFDP package described in this document.
<code>psp_template_spi</code>	The SPI Platform Support Package (PSP).

Documents

For an overview of HCC file systems and guidance on choosing a file system, see [Product Information](#) on the main HCC website.

Readers should note the points in the [HCC Documentation Guidelines](#) on the HCC documentation website.

HCC Firmware Quick Start Guide

This document describes how to install packages provided by HCC in the target development environment. Also follow the *Quick Start Guide* when HCC provides package updates.

HCC Source Tree Guide

This document describes the HCC source tree. It gives an overview of the system to make clear the logic behind its organization.

HCC SafeFLASH File System User Guide

This document describes the base SafeFLASH System.

HCC SafeFLASH File System NOR Drive User Guide

This document describes the SafeFLASH NOR driver that is used with the NOR CFI/SFDP SPI package.

HCC SafeFLASH NOR Driver for CFI and SFDP SPI Flash User Guide

This is this document.

1.5. Change History

This section describes past changes to this manual.

- To download this manual or a PDF describing an [earlier software version, see File System PDFs](#).
- For the history of changes made to the package code, see [History: fs_safe_nor_drv_cfi_sfdp_spi](#).

The current version of this manual is 2.00. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
2.00	2020-02-03	1.11 R2	New template.
1.50	2019-11-19	1.11 R2	Added QSPI reference PSP for Nucleo-F722ZE board.
1.40	2019-10-31	1.11	Added QSPI references, mainly in <i>Configuration Options</i> .
1.30	2019-01-23	1.09	Corrected diagram in <i>Introduction</i> .
1.20	2017-08-31	1.09	Corrected <i>Packages</i> list.
1.10	2017-06-26	1.08	New <i>Change History</i> format.
1.00	2017-04-24	1.08	First online version.

2. Source File List

The following sections describe all the source code files included in the system. These files follow the HCC Embedded standard source tree system, described in the [HCC Source Tree Guide](#). All references to file pathnames refer to locations within this standard source tree, not within the package you initially receive.

Note: Do not modify any files except the configuration file and PSP files.

Configuration File

The file `src/config/config_safe_nor_cfi_sfdp.h` contains the configurable parameters of the system. Configure these as required. For detailed explanation of these options, see [Configuration Options](#).

System Files

These files are in the directory `src/safe-flash/nor/phy/cfi_sfdp`. **These files should only be modified by HCC.**

File	Description
<code>cfi_sfdp_spi.c</code>	SPI driver source code.
<code>cfi_sfdp_spi.h</code>	SPI driver header file.

Version File

The file `src/version/ver_safe_nor_cfi_sfdp.h` contains the version number of this module. This version number is checked by all modules that use this module to ensure system consistency over upgrades.

Platform Support Package (PSP) Files

There are two sets of files.

`psp_template`

These files in the directory `src/psp/target/safe_nor_cfi_sfdp` define the [`psp_safe_nor_cfi_sfdp_delay\(\)`](#) function. Modify these files as required for your hardware; see [PSP Porting](#).

Note:

- These are PSP implementations for the specific microcontroller and board; you may need to modify these to work with a different microcontroller and/or development board.
- In the package these files are offset to avoid overwriting an existing implementation. Copy them to the root `hcc` directory for use.

The files are as follows:

File	Description
psp_safe_cfi_sfdp.c	Source code defining the psp_safe_nor_cfi_sfdp_delay() function. This waits at least 100 ns, the chip-select deselect time.
psp_safe_cfi_sfdp.h	Header file.

This PSP also has a version file, **ver_psp_safe_nor_cfi_sfdp.h**.

Nucleo-F722ZE QSPI Board

These files in the directory **psp_stm32f222ze_nucleo_qspi/src** define the PSP for this board.

File	Description
config/config_ftl_nor_cfi_sfdp.h	Configuration file.
psp/include/psp_qspi.h	Transfer mode settings and psp_qspi_xxx() function definitions.
psp/target/ftl_nor_cfi_sfdp/psp_ftl_nor_cfi_sfdp.c and .h	Source code and header file defining the psp_safe_nor_cfi_sfdp_delay() function.
psp/target/include/hcc_stm32f7xx_regs.h	Registers file.
psp/target/qspi/psp_qspi.c	Source code of psp_qspi_xxx() functions.

This PSP has the following version files:

File	Description
ver_psp_ftl_nor_cfi_sfdp.c	psp_safe_nor_cfi_sfdp_delay() files version.
ver_psp_proc_reg.h	Registers file version.
ver_psp_qspi.h	PSP version.

3. Configuration Options

Set the configuration options in the file `src/config/config_safe_nor_cfi_sfdp.h`. This section lists the available options and their default values.

NOR_PAGE_SIZE

The maximum number of bytes for the page program command. The default is 256.

NOR_BLOCK_THRESHOLD

The block sector size threshold. Below this number of blocks, a 4K sector size is preferred. The default is 128.

Note: For full instructions on how to set up the following options, see the following sections in the *HCC SafeFLASH File System NOR Drive User Guide*:

- [Sectors and File Storage](#)
- [Physical Device Usage](#) - this describes all the block types in detail and gives examples.

NOR_BLOCKSTART

Specify the space before the block start that is not to be used by the file system. The default is 0.

NOR_SECTORSIZE

The logical sector size. This must be less than or equal to the block size ($g_block_size/NOR_BLOCK_SIZE$ or $NOR_VIRTUAL_BLOCK_SIZE$). The default is 512.

NOR_DESCSIZE

The descriptor size. This must be less than or equal to the block size ($g_block_size/NOR_BLOCK_SIZE$ or $NOR_VIRTUAL_BLOCK_SIZE$). The default is 16384.

NOR_CACHEDESCSIZE

The cache size. This must be less than the descriptor size ($NOR_DESCSIZE$). The default is 2048.

NOR_VIRTUAL_BLOCK_SIZE

Physical blocks can be joined to a larger virtual block (to allow use of a larger descriptor block). Set this option to 0 to disable virtual blocks, otherwise it must be a multiple of $g_block_size/NOR_BLOCK_SIZE$. The default is 16384.

Note: The following three parameters are only used if flash identification is not possible using SFDP or CFI.

NOR_BLOCK_SIZE

The block size. The default is 4096.

NOR_NUM_BLOCKS

The number of blocks. The default is 128.

NOR_CMD_ERASE

The erase command for the block size specified by `NOR_BLOCK_SIZE`. The default is 0x20.

NOR_TBPARAM_TOP

If this is non-zero, the driver checks whether the TBPARAM bit is set (that is, whether the device is already configured to have parameter sectors on the top of the address space). If this OTP bit is not set, it sets it. The default is 1.

NOR_ENABLE_QSPI

Set this to 1 to use QSPI instead of SPI. The default is 0. In QSPI mode the driver uses four lines to send/receive data, but only one line (MOSI) to send command and address information.

NOR_SPI_UNIT

The ID of the SPI or QSPI to use. The default is 0.

NOR_SPI_BAUDRATE

The SPI/QSPI clock frequency in Hz . The default is 20000000.

NOR_TMO_ERASE_MS

The erase timeout in milliseconds. The default is 3000.

NOR_TMO_PROGRAM_MS

The program timeout in milliseconds. The default is 500.

NOR_TMO_WREN_MS

The write enable timeout in milliseconds. The default is 5.

NOR_TMO_WRITE_SR_MS

The write status register timeout in milliseconds. The default is 8.

NOR_CFI_SFDP_TMO_RESET_US

The reset timeout in microseconds. The default is 260.

4. PSP Porting

The Platform Support Package (PSP) is designed to hold all platform-specific functionality, either because it relies on specific features of a target system, or because this provides the most efficient or flexible solution for the developer.

The file **config_ftl_nor_cfi_sfdp.h** defines configuration options for the Nucleo-F722ZE QSPI board.

The files **psp_safe_cfi_sfdp.c** and **psp_safe_cfi_sfdp.h** define the [psp_safe_nor_cfi_sfdp_delay\(\)](#) function. Modify these files as required for your hardware.

QSPI Functions

The functions used by the Nucleo-F722ZE QSPI Board PSP are as follows. For details, see the [detailed descriptions](#).

Function	Description
psp_qspi_init()	Initializes a QSPI port and allocates the required resources.
psp_qspi_start()	Starts a QSPI port.
psp_qspi_stop()	Stops a QSPI port.
psp_qspi_delete()	Deletes a QSPI port and releases the resources it used.
psp_qspi_get_baudrate()	Gets the baud rate.
psp_qspi_set_baudrate()	Sets the baud rate.
psp_qspi_lock()	Locks the QSPI for the specific unit.
psp_qspi_unlock()	Unlocks the QSPI for the specific unit.
psp_qspi_register_isr()	Registers the ISR ID for a QSPI unit. This only applies if ISRs are supported.
psp_qspi_transfer()	Transmits/receives data over the QSPI.
psp_qspi_wait_status()	Waits until the status register matches a value.

4.1. psp_safe_nor_cfi_sfdp_delay

Use this function to wait for at least 100 ns, which is the chip select/deselect time.

Format

```
void psp_safe_nor_cfi_sfdp_delay ( void )
```

Arguments

None.

Return Values

None.

4.2. config_ftl_nor_cfi_sfdp.h

Set the configuration options for the Nucleo-F722ZE QSPI board in the PSP file `src/config/config_ftl_nor_cfi_sfdp.h`. This section lists the available options and their default values.

NOR_CFI_SFDP_PAGE_SIZE

The maximum number of bytes for the page program command. This is purely logical on NOR flashes. The default is 256.

NOR_CFI_SFDP_BLOCK_THRESHOLD

The block sector size threshold. Below this number of blocks, a 4K sector size is preferred. The default is 128.

NOR_CFI_SFDP_RESERVED_BLOCKS

The number of reserved blocks. FTL will not access blocks below this index. The default is 0.

Note: These following two options control the FTL's static wear leveling.

NOR_CFI_SFDP_STATIC_LIMIT

The maximum allowed difference between maximum and minimum wear count. The default is 1024.

NOR_CFI_SFDP_STATIC_COUNT

The number of merge operations before static wear checking must be run. The default is 128.

Note: The following parameters are only used if flash identification is not possible using SFDP of CFI.

NOR_CFI_SFDP_BLOCK_SIZE

The block size. The default is 4096.

NOR_CFI_SFDP_PAGE_PER_BLOCK

The number of pages in a block. The default is 16.

NOR_CFI_SFDP_NUM_BLOCKS

The number of blocks. The default is 4096.

NOR_CFI_SFDP_TBPARAM_TOP

If this is non-zero, the driver checks whether the TBPARAM bit is set (that is, whether the device is already configured to have parameter sectors on the top of the address space). If this OTP bit is not set, it sets it. The default is 1.

NOR_CFI_SFDP_LEGACY_MODE

There are two options:

- 0: Use automatic map and log block calculation (recommended for new designs). This is the default.
- 1: The layout (map and log block numbers) is compatible with **media_drv_ftl_nor_cfi_sfdp_spi** version 1.5.

NOR_CFI_SFDP_SPI_UNIT

The ID of the SPI or QSPI to use. The default is 0.

NOR_CFI_SFDP_QSPI

Keep the default of 1 to use QSPI instead of SPI. In QSPI mode the driver uses four lines to send/receive data, but only one line (MOSI) to send command and address information.

NOR_CFI_SFDP_SPI_BAUDRATE

The SPI/QSPI clock frequency in Hz . The default is 50000000.

NOR_CFI_SFDP_TMO_ERASE_MS

The erase timeout in milliseconds. The default is 3000.

NOR_CFI_SFDP_TMO_PROGRAM_MS

The program timeout in milliseconds. The default is 500.

NOR_CFI_SFDP_TMO_WREN_MS

The write enable timeout in milliseconds. The default is 5.

NOR_CFI_SFDP_TMO_WRITE_SR_MS

The write status register timeout in milliseconds. The default is 8.

NOR_CFI_SFDP_TMO_RESET_US

The reset timeout in microseconds. The default is 260.

NOR_CFI_SFDP_DIVIDED

Set this to 1 if NOR is split into two separated media drives. The default is 0.

Note: The following option only applies if NOR_CFI_SFDP_DIVIDED is set.

NOR_CFI_SFDP_BLOCK_NUM1

The number of blocks in the first part of flash if flash is divided into two parts. The default is 0x80.

NOR_CFI_SFDP_DRV_NUM

The number of drives. If NOR_CFI_SFDP_DIVIDED is set this is 2, otherwise it is 1.

4.3. QSPI Functions

This section contains the QSPI Functions.

psp_qspi_init

Use this function to initialize a QSPI port.

Note: You must call this before any other QSPI function.

Format

```
t_qspi_ret psp_qspi_init ( uint8_t uid )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t

Return Values

Return value	Description
PSP_QSPI_SUCCESS	Successful execution.
PSP_QSPI_ERROR	Operation failed.

psp_qspi_delete

Use this function to delete a QSPI port and release the associated resources.

Format

```
t_qspi_ret psp_qspi_delete ( uint8_t uid )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t

Return Values

Return value	Description
PSP_QSPI_SUCCESS	Successful execution.
PSP_QSPI_ERROR	Operation failed.

psp_qspi_stop

Use this function to stop a QSPI port.

Format

```
t_qspi_ret psp_qspi_stop ( uint8_t uid )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t

Return Values

Return value	Description
PSP_QSPI_SUCCESS	Successful execution.
PSP_QSPI_ERROR	Operation failed.

psp_qspi_start

Use this function to start a QSPI port.

Note: You must call **psp_qspi_init()** before this to initialize the module.

Format

```
t_qspi_ret psp_qspi_start ( uint8_t uid )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t

Return Values

Return value	Description
PSP_QSPI_SUCCESS	Successful execution.
PSP_QSPI_ERROR	Operation failed.

psp_qsapi_register_isr

Use this function to register the ISR ID for a QSPI unit.

Note: This only applies if ISRs are supported.

Format

```
t_qsapi_ret psp_qsapi_register_isr (  
    uint8_t      uid,  
    oal_isr_id_t isr_id )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t
isr_id	The ISR ID.	oal_isr_id_t

Return Values

Return value	Description
PSP_QSPI_SUCCESS	Successful execution.
PSP_QSPI_ERROR	Operation failed.

psp_qspi_get_baudrate

Use this function to get the baud rate.

Format

```
t_qspi_ret psp_qspi_get_baudrate (  
    uint8_t    uid,  
    uint32_t * p_br )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t
p_br	The baud rate in Hz.	uint32_t*

Return Values

Return value	Description
PSP_QSPI_SUCCESS	Successful execution.
PSP_QSPI_ERROR	Operation failed.

psp_qspi_set_baudrate

Use this function to set the baud rate.

Format

```
t_qspi_ret psp_qspi_set_baudrate (  
    uint8_t    uid,  
    uint32_t   p_br )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t
p_br	The baud rate in Hz.	uint32_t

Return Values

Return value	Description
PSP_QSPI_SUCCESS	Successful execution.
PSP_QSPI_ERROR	Operation failed.

psp_qspi_lock

Use this function to lock the QSPI for the specific unit.

This can be useful if multiple units are attached to the same QSPI bus.

Format

```
t_qspi_ret psp_qspi_lock (  
    uint8_t  uid,  
    uint8_t  from_isr )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t
from_isr	The function is called from the interrupt flag.	uint8_t

Return Values

Return value	Description
PSP_QSPI_SUCCESS	Successful execution.
PSP_QSPI_ERROR	Operation failed.

psp_qspi_unlock

Use this function to unlock the QSPI for the specific unit.

This can be useful if multiple units are attached to the same QSPI bus.

Format

```
t_qspi_ret psp_qspi_unlock (  
    uint8_t  uid,  
    uint8_t  from_isr )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t
from_isr	The function is called from the interrupt flag.	uint8_t

Return Values

Return value	Description
PSP_QSPI_SUCCESS	Successful execution.
PSP_QSPI_ERROR	Operation failed.

psp_qspi_transfer

Use this function to transmit and receive a number of bytes.

Format

```
t_psp_qspi_ret psp_qspi_transfer (  
    uint8_t      uid,  
    t_qspi_transfer * p_transfer)
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t
p_transfer	A pointer to the transfer parameters.	t_qspi_transfer *

Return Values

Return value	Description
PSP_QSPI_SUCCESS	Successful execution.
PSP_QSPI_ERROR	Operation failed.

psp_qspi_wait_status

Use this function to wait until the status register matches a value.

Format

```
t_psp_qspi_ret psp_qspi_transfer (  
    uint8_t      uid,  
    t_qspi_wait_status * p_wait_status)
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t
p_wait_status	A pointer to the wait parameters.	t_qspi_wait_status *

Return Values

Return value	Description
PSP_QSPI_SUCCESS	Successful execution; the status register changed to the predefined value.
PSP_QSPI_ERROR	Operation failed.

t_qspi_transfer

The *t_qspi_transfer* structure is as follows:

Element	Type	Description
mode	uint8_t	The transfer mode, PSP_QSPI_TRANSFER_MODE_XXX
cmd	uint8_t	The command to execute.
addr	uint32_t	The address to send.
addr_size	uint8_t	The number of address bytes, 3 or 4 bytes.
dummy_cycles	uint8_t	The number of dummy clocks between address and data bytes .
p_data	uint8_t*	A pointer to the data buffer.
data_size	uint32_t	The number of bytes in the data buffer.

t_qspi_wait_status

The content of the *t_qspi_wait_status* structure is as follows:

Element	Type	Description
cmd	uint8_t	The command to send to get the register's value.
status_size	uint8_t	The number of bytes in the status register.
status_mask	uint32_t	The status register mask.
status_value	uint32_t	The masked status register compared to this value.
timeout_ms	uint32_t	The timeout in milliseconds.

Transfer Modes

The possible transfer modes are as follows:

Mode	Value	Description
PSP_QSPI_TRANSFER_MODE_ADDR_1L	0x01	Send address on a single line.
PSP_QSPI_TRANSFER_MODE_ADDR_2L	0x02	Send address on two lines.
PSP_QSPI_TRANSFER_MODE_ADDR_4L	0x04	Send address on four lines.
PSP_QSPI_TRANSFER_MODE_DATA_WR	0x08	Send data.
PSP_QSPI_TRANSFER_MODE_DATA_RD	0x10	Receive data.
PSP_QSPI_TRANSFER_MODE_DATA_1L	0x20	Transfer data on a single line.
PSP_QSPI_TRANSFER_MODE_DATA_2L	0x40	Transfer data on two lines.
PSP_QSPI_TRANSFER_MODE_DATA_4L	0x80	Transfer data on four lines.

The address mask is defined as follows:

```
PSP_QSPI_TRANSFER_MODE_ADDR_MASK \
    ( (uint8_t) ( PSP_QSPI_TRANSFER_MODE_ADDR_1L \
                | PSP_QSPI_TRANSFER_MODE_ADDR_2L \
                | PSP_QSPI_TRANSFER_MODE_ADDR_4L ) )
```

For sending data the mode is:

```
PSP_QSPI_TRANSFER_MODE_ADDR_1L | PSP_QSPI_TRANSFER_MODE_DATA_WR |
PSP_QSPI_TRANSFER_MODE_DATA_4L
```

For receiving data the mode is:

```
PSP_QSPI_TRANSFER_MODE_ADDR_1L | PSP_QSPI_TRANSFER_MODE_DATA_RD
| PSP_QSPI_TRANSFER_MODE_DATA_4
```

5. Version

Version 2.00

For use with SafeFLASH NOR Driver for CFI and SFDP SPI Flash versions 1.11 R2 and above