SPI Flash Library Reference Guide

V1.00.001

Publication Release Date: Aug. 2011

Support Chips:

ISD9160

Support Platforms:

NuvotonPlatform_Keil

The information in this document is subject to change without notice.

The Nuvoton Technology Corp. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

Table of Contents

1. Introduction

1.1.SPI Flash Introduction

Serial Flash memories provide a storage solution for systems with limited space, pin, and power. They are ideal for code download applications as well as storing voice, text, and data. The device operates on a single 2.7V to 3.6 V power supply with current consumption as low as 4mA active and 1uA for power-down.

1.2.SPI Flash Feature

The SPI Flash includes following features:

- 4-pin SPI Serial Interface
- Low Power Consumption. Wide Temperature Range
- Fast and Flexible Serial Data Access
- Programming Features
- Software and Hardware Write Protection
- Parameter Page
- Ideal for systems with limited pins, space, and power

1.3.SPI Flash Support List

Winbond – W25Q10, W25Q20, W25Q40, W25Q80, W25Q16, W25Q32, W25Q64, W25Q128

1.4.SPI Flash Limitation

Before SPI flash programming, the operation area should be erased first. The minimum erase size is a sector 4KB. Programmer need to handle the remained valid data in a sector if operation data size is not the multiple of 4KB.
The read/writer functions need word (4 bytes) alignment for buffer.

2. APIs Specification

2.1.Data Type Definition

```
typedef struct tagSFLASH_CTX
{
   unsigned long spi_base;
   unsigned long spi_chnl; // 1, 2
   void (*spi_setcs)(const struct tagSFLASH_CTX *ctx, int csOn);
}SFLASH_CTX;
```

2.2.API Functions

LibSPIFlash_Open

Prototype

```
void    LibSPIFlash_Open(
   const SFLASH_CTX *ctx
);
```

Description

SPI flash initialization.

Parameter

ctx [in]

The SPI flash structure handler.

Include

LibSPIFlash.h

Return Value

None

LibSPIFlash_GetID

Prototype

```
void    LibSPIFlash_GetID(
```

```
    const SFLASH_CTX *ctx
);
```

Description

Erase whole SPI flash.

Parameter

ctx [in]

The SPI flash structure handler.

Include

LibSPIFlash.h

Return Value

0

LibSPIFlash_WriteEnable

Prototype

```
void    LibSPIFlash_WriteEnable(
    const SFLASH_CTX *ctx
);
```

Description

Set SPI-flash WEL bit for flash erase and program.

Parameter

ctx [in]

The SPI flash structure handler.

Include

LibSPIFlash.h

Return Value

None

LibSPIFlash_Erase

Prototype

```
int     LibSPIFlash_Erase(
    const SFLASH_CTX *ctx,
    unsigned long addr,
    unsigned long len
```

);

Description

Erase SPI Flash memory size with specified address and length (unit: byte). This API uses sector erase (4KB)

Parameter

ctx[in]

The SPI flash structure handler.

addr[in]

The start address for erase, should be the multiple of 4096.

len[in]

The length for erase, should be the multiple of 4096.

Include

LibSPIFlash.h

Return Value

0

LibSPIFlash_EraseAll

Prototype

void LibSPIFlash_EraseAll(

const SFLASH_CTX *ctx

);

Description

Erase whole SPI flash.

Parameter

ctx [in]

The SPI flash structure handler.

Include

LibSPIFlash.h

Return Value

None

LibSPIFlash_Write

Prototype

```
int     LibSPIFlash_Write(
    const SFLASH_CTX *ctx,
    unsigned long addr,
    const unsigned long* p,
    unsigned long bytes
);
```

Description

Copy the data from a buffer to specified address of SPI flash with page programming.

Parameter

ctx[in]

The SPI flash structure handler.

addr[in]

The start SPI-flash address for programming, should be the multiple of 256.

p[in]

Specify the buffer start pointer to be transferred to SPI flash.

bytes[in]

The length for programming, unit is byte. Should be the multiple of 256.

Include

LibSPIFlash.h

Return Value

0

LibSPIFlash_Read

Prototype

```
int     LibSPIFlash_Read(
    const SFLASH_CTX *ctx,
    unsigned long addr,
    const unsigned long* p,
    unsigned long bytes
);
```

Description

Copy the data from specified address of SPI flash to a buffer with repeating the read command.

Parameter

ctx[in]

The SPI flash structure handler.

addr[in]

The start SPI-flash address for reading.

p[in]

Specify the buffer start pointer to be written for reading data.

bytes[in]

The length for reading, unit is byte. Should be the multiple of 4.

Include

LibSPIFlash.h

Return Value

0

LibSPIFlash_FastRead

Prototype

```
int     LibSPIFlash_FastRead(
    const SFLASH_CTX *ctx,
    unsigned long addr,
    const unsigned long* p,
    unsigned long bytes
);
```

Description

Copy the data from specified address of SPI flash to a buffer with the command of continuous reading.

Parameter

ctx[in]

The SPI flash structure handler.

addr[in]

The start SPI-flash address for reading.

p[in]

Specify the buffer start pointer to be written for reading data.

bytes[in]

The length for reading, unit is byte. Should be the multiple of 4.

Include

LibSPIFlash.h

Return Value

0

3. Revision History

Version	Date	Description
V1.00.001	Aug.30, 2011	● Created

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in equipment or systems intended for surgical implantation, atomic energy control instruments, aircraft or spacecraft instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for any other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications whereby failure could result or lead to personal injury, death or severe property or environmental damage.

Nuvoton customers using or selling these products for such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from their improper use or sales.