



SQFlash Utility

User Manual

Version v3.1.16477

Advantech Co. Ltd.

No. 1, Alley 20, Lane 26,

Rueiguang Road, Neihu District,

Taipei 114, Taiwan

Copyright Notice

This document is copyrighted, 2021, by Advantech Co., Ltd. All rights reserved. Advantech Co., Ltd. Reserves the right to make improvements to the products described in this manual at any time. Specifications are thus subject to change without notice.

No part of this manual may be reproduced, copied, translated, or transmitted in any form or by any means without prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd., assumes no responsibility for its use, or for any infringements upon the rights of third parties which may result from its use.

All the trade marks of products and companies mentioned in this data sheet belong to their respective owners.

Copyright © 1983-2021 Advantech Co., Ltd. All Rights Reserved

Version History

Date	Version	Remark
2008-11-20	0.1	First release
2008-12-03	0.2	Update Tool
2008-12-15	0.3	Update naming & PN
2009-01-06	1.0	Add UnInitializeCF API
2009-01-19	1.1	Update SMART attribute
2009-01-20	1.2	Update PN table
2009-02-12	1.3	Add Flash Lock Feature
2010-08-17	2.0.0812.1	1. Version 2.0 release
2011-05-19	2.0.1621.1	1. Add New Model Support
2011-06-10	2.0.1808.1	1. Temporarily remove SMART support on models with FW version "Ver7.M0K" for bug solving
2013-07-19	2.0.4211.1	1. Add New Model Support
2015-01-02	2.1.218.1	1. Add New Model Support
2016-02-15	3.0.13738.0	Version 3.0 release
2021-09-09	3.1.16477.0	Support NVMe disk

Table of Contents

Introduction	4
Environment Requirements	5
Operating Systems	5
SQFlash API	6
Status code	6
Information Functions	8
Vendor SMART functions	16
Self-Test Functions	19
Crypto Erase Functions	21
Flash Lock V2 Functions	22
Security Initialize Functions	24
Security ID Functions	26
Flash Vault Functions	29

Introduction

SQFlash Utility software package is a flash management package contains utility and API to access Advantech flash storages and provides Software Protection and Life Monitoring (S.M.A.R.T.) features. Benefitting from an AES-256 encryption engine, the integrated TCG-OPAL self-encryption disk functions seamlessly in the SQFlash via the Flash Lock (patented) feature. It also provides a Crypto Erase function that can purge the disk within 10 milliseconds. Through the utility, the disk security function focuses on protecting customer's software (Security ID), drive accessibility (Flash Lock), and data accessibility (Flash Vault). With the function enabled, data can be protected from hacking or misuse by unauthorized users. In addition, user can easily get disk status, endurance estimation, and real-time status updates including operating information such as power-on time, temperature, power-cycle counts, and any error logs for diagnostic purposes.

Benefits

■ **Faster Time to Market**

The Utility is ready to run without modifications. System developers can use it to control the SQFlash without knowing the controller specs of the SQFlash. API and sample code are ready for software developers to implement flash management mechanism into their applications.

■ **Protect Your Intellectual Property**

In order to help protect customers' intellectual property, Advantech has designed in Security ID feature for Advantech SQFlash. Customers can easily implement security functions on their applications based on encrypted utility and library. Flash Lock feature helps user to lock SQFlash to protect flash data not to be read.

■ **Monitor SQFlash Health**

Advantech SQFlash management package provides utility and API to get Self-Monitoring, Analysis, and Reporting Technology (S.M.A.R.T.) information from SQFlash cards. Customer can monitor the flash storage health and design an early warning mechanism by life-span detection.

Environment Requirements

Operating Systems

- Windows XP Embedded
- Windows XP Professional or Home Edition
- Windows 7
- Windows 8 Desktop
- Windows 10

SQFlash API

sqflib.h file includes the API declaration, constants and flags that are required for programming.

Status code

All SQFlash API functions will return a status code to indicate the possible errors. Any function may return any of the defined status codes.

#define SQF_SUCCESS 0

The operation is successful.

#define SQF_ERROR -1

Generic error message. No further error details are available.

#define SQF_ACCESS_FAILED -2

Failed to access the SQFlash device.

#define SQF_CMD_FAILED -3

Failed to execute command.

#define SQF_SIZE_TOO_SMALL -5

The given length is too small.

#define SQF_INVALID_PARAMETER -6

The given parameter is invalid.

#define SQF_NOT_FOUND_RESOURCE -7

Does not find the resource in the specified region.

#define SQF_NOT_FOUND_DISK -10

Does not find specified disk.

#define SQF_INVALID_SQFLASH -11

The specified disk is not a SQFlash device.

#define SQF_INVALID_APKEY -12

The given apkey is invalid.

#define SQF_INVALID_ACCESS_CODE -13

The given access code is invalid.

#define SQF_INVALID_PASSWORD -14

The given password is invalid.

#define SQF_INVALID_ACTION -15

The specified action is invalid.

#define SQF_UNSUPPORTED_FEATURE -20

The specified feature is unsupported.

#define SQF_ACCESS_CODE_EXIST 10

The operation is successful, and the access code exists.

#define SQF_ACCESS_CODE_NOT_EXIST 11

The operation is successful, but the access code does not exist.

#define SQF_ACCESS_CODE_NOT_MATCH 12

The given access code does not match with the existing access code of the SQFlash device.

Information Functions

SQFlash_GetModelName

Syntax:

```
int SQFlash_GetModelName(int index, char ModelNameStr[41])
```

Description:

Get model name of the SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

ModelNameStr

[out] Model name of the SQFlash device.

Return:

Refer to the section of status code.

SQFlash_GetSerialNumber

Syntax:

```
int SQFlash_GetSerialNumber(int index, char SerialNumberStr[21])
```

Description:

Get serial number of the SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

SerialNumberStr

[out] Serial number of the SQFlash device.

Return:

Refer to the section of status code.

SQFlash_GetFirmwareRevision

Syntax:

```
int SQFlash_GetFirmwareRevision(int index, char FirmwareRevisionStr[9])
```

Description:

Get firmware version of the SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

FirmwareRevisionStr

[out] Firmware revision of the SQFlash device.

Return:

Refer to the section of status code.

SQFlash_GetIdentifyDevice

Syntax:

```
int SQFlash_GetIdentifyDevice(int index, unsigned char IdentifyDevice[512])
```

Description:

Get raw identify data of the SQFlash device.

Please refer to ATA / NVMe spec of identify device to convert raw data into readable information.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

IdentifyDevice

[out] Raw identify device of the SQFlash device.

Return:

Refer to the section of status code.

SQFlash_GetDiskType

Syntax:

```
int SQFlash_GetDiskType(int index, char DiskType[4])
```

Description:

Get disk type of the SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

DiskType

[out] Disk Type of the SQFlash device, should be “SSD” or “HDD”

Return:

Refer to the section of status code.

SQFlash_GetInformation

Syntax:

```
int SQFlash_GetInformation(int index, uint32_t infoType, uint32_t *result)
```

Table infoType

ID	Description
SQF_INFO_TYPE_INTERFACE	Query interface type. Result See Table Interface Type .
SQF_INFO_TYPE_ENDURANCE	Query endurance
SQF_INFO_TYPE_OPALSTATUS	Query OPAL status Result: <ul style="list-style-type: none"> ● Bit 0: locking Supported ● Bit 1: locking Enabled ● Bit 2: locked or not

Table Interface Type

ID	Description
SQF_INTERFACE_UNKNOWN	Unknown interface
SQF_INTERFACE_SATA	SATA interface
SQF_INTERFACE_NVME	NVMe interface

Description:

Get information of the SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

infoType

[in] Specific which information you want to query. See [Table infoType](#).

result

[out] According to infoType to return query result. See [Table infoType](#).

Return:

Refer to the section of status code.

SQFlash_ReadSMARTAttribute

Syntax:

```
int SQFlash_ReadSMARTAttribute(int index, unsigned char SmartData[512])
```

Description:

Read raw SMART attributes of the SATA SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list.

SmartData

[out] Raw smart data of the SATA SQFlash device.

Return:

Refer to the section of status code.

SQFlash_ReadSMARTAttributeThresholds

Syntax:

```
int SQFlash_ReadSMARTAttributeThresholds(int index, unsigned char
AttributeThresholds[512])
```

Description:

Read thresholds of the SMART attributes of the SATA SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

AttributeThresholds

[out] The threshold of the attributes of the SATA SQFlash device.

Return:

Refer to the section of status code.

SQFlash_ReadParsedSMARTAttribute

Syntax:

```
int SQFlash_ReadParsedSMARTAttribute(int index, PSATASmart_t
sataSmart)
```

SATASmart Structure:

```
typedef struct SATASmart {
    uint64_t UncorrectableECCCnt;
    uint64_t PowerOnHours;
    uint64_t PowerCycleCnt;
    uint32_t DeviceCapacity;
    uint32_t UserCapacity;
    uint32_t TotalAvailSpareBlk;
    uint32_t RemainingSpareBlk;
    uint32_t TotalEraseCnt;
    uint64_t PHYErrCnt;
```

```

uint16_t LaterBadBlkCnt;
uint16_t EarlyBadBlkCnt;
uint16_t AvgEraseCnt;
uint16_t MaxEraseCnt;
uint32_t UnexpectedPwrLostCnt;
uint32_t VolStabilizerTriggerCnt;
uint8_t GuaranteedFlush;
uint8_t DriveStatus;
uint16_t UnexpectedPwrLostCnt2;
uint16_t MaxTemperature;
uint16_t MinTemperature;
uint16_t CurTemperature;
uint8_t SSDLifeUsed;
uint64_t CRCErrCnt;
uint8_t SSDLifeLeft;
uint64_t TotalNANDRead;
uint64_t TotalNANDWritten;
uint64_t HostWrite;
uint64_t HostRead;
} SATASmart_t, *PSATASmart_t;

```

Description:

Read parsed SMART attributes of the SATA SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

sataSmart

[out] Parsed SMART attributes of the SATA SQFlash device.

Return:

Refer to the section of status code.

SQFlash_NVMe_StdSMARTRaw

Syntax:

```
int SQFlash_NVMe_StdSMARTRaw(int index, uint8_t stdSmartRaw[512])
```

Description:

Read raw SMART attributes of the NVMe SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

stdSmartRaw

[out] Raw smart data of the NVMe SQFlash device.

Return:

Refer to the section of status code.

SQFlash_NVMe_StdSMART

Syntax:

```
int SQFlash_NVMe_StdSMART(int index, PNVMStdSMART_t  
stdSmart)
```

NVMeStdSMART Structure:

```
typedef struct NVMeStdSMART {
    uint8_t CriticalWarning;
    uint16_t CompositeTemperature;
    uint8_t AvailableSpare;
    uint8_t AvailableSpareThreshold;
    uint8_t PercentageUsed;
    uint8_t EnduGrpCritWarnSumry;
    uint8_t DataUnitsRead[16];
    uint8_t DataUnitsWritten[16];
    uint8_t HostReadCommands[16];
    uint8_t HostWriteCommands[16];
    uint8_t ControllerBusyTime[16];
}
```

```

uint8_t PowerCycles[16];
uint8_t PowerOnHours[16];
uint8_t UnsafeShutdowns[16];
uint8_t MediaErrors[16];
uint8_t ErrorLogNumber[16];
uint32_t WarningCompositeTemperatureTime;
uint32_t CriticalCompositeTemperatureTime;
uint16_t TemperatureSensor1;
uint16_t TemperatureSensor2;
uint16_t TemperatureSensor3;
uint16_t TemperatureSensor4;
uint16_t TemperatureSensor5;
uint16_t TemperatureSensor6;
uint16_t TemperatureSensor7;
uint16_t TemperatureSensor8;
uint32_t ThmTemp1TransCnt;
uint32_t ThmTemp2TransCnt;
uint32_t ThmTemp1TotalTime;
uint32_t ThmTemp2TotalTime;
} NVMeStdSMART_t, *PNVMeStdSMART_t;

```

Description:

Read parsed SMART attributes of the NVMe SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

sataSmart

[out] Parsed SMART attributes of the NVMe SQFlash device.

Return:

Refer to the section of status code.

Vendor SMART functions

SQFlash_ReadVendorSMARTAttributes

Syntax:

```
int SQFlash_ReadVendorSMARTAttributes(int index, PVSMARTAttr_t Attr, int AttrSize)
```

VSMARTAttr Structure:

```
typedef struct VSMARTAttr {
    unsigned int MaxProgram;
    unsigned int AverageProgram;
    unsigned int EnduranceCheck;
    unsigned int PowerOnTime;
    unsigned int EccCount;
    unsigned int MaxReservedBlock;
    unsigned int CurrentReservedBlock;
    unsigned int GoodBlockRate;
} VSMARTAttr_t, *PVSMARTAttr_t;
```

Description:

Read vendor SMART attributes of the SATA SQFlash device. The vendor attribute contains 8 data:

- MaxProgram
Max program and erase cycles.
- AverageProgram
Average program and erase cycles.
- EnduranceCheck
The percentage of (Average P/E cycles) / (Max P/E cycles).
- PowerOnTime
Accumulated times of power on.
- EccCount
Error correct code number of times counting.
- MaxReservedBlock
Max amount of reserved block

- CurrentReservedBlock
Current amount of reserved block
- GoodBlockRate
Rate of good blocks

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

Attr

[out] The attribute of vendor SMART of the SATA SQFlash device.

AttrSize

[in] Size of VSMARTAttr Structure.

Return:

Refer to the section of status code.

SQFlash_NVMe_VendorSMARTRaw

Syntax:

```
int SQFlash_NVMe_VendorSMARTRaw(int index, uint8_t  
vendorSmartRaw[512])
```

Description:

Read raw vendor SMART attributes of the NVMe SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

stdSmartRaw

[out] Raw vendor smart data of the NVMe SQFlash device.

Return:

Refer to the section of status code.

SQFlash_NVMe_VendorSMART

Syntax:

```
int SQFlash_NVMe_VendorSMART(int index, PNVMVenSMART_t  
vendorSmart)
```

NVMeVenSMART Structure:

```
typedef struct NVMeVenSMART {  
    uint8_t FlashReadSector[8];  
    uint8_t FlashWriteSector[8];  
    uint8_t UNCErrors[8];  
    uint32_t PyhError;  
    uint32_t EarlyBadBlock;  
    uint32_t LaterBadBlock;  
    uint32_t MaxEraseCount;  
    uint32_t AvgEraseCount;  
    uint8_t CurPercentSpares[8];  
    uint16_t CurTemperature;  
    uint16_t LowestTemperature;  
    uint16_t HighestTemperature;  
    uint16_t ChipInternalTemperature;  
    uint16_t SpareBlocks;  
} NVMeVenSMART_t, *PNVMVenSMART_t;
```

Description:

Read parsed vendor SMART attributes of the NVMe SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

vendorSmart

[out] Parsed vendor SMART attributes of the NVMe SQFlash device.

Return:

Refer to the section of status code.

Self-Test Functions

SQFlash_ExecuteSMARTSelfTest

Syntax:

```
int SQFlash_ExecuteSMARTSelfTest(int index, unsigned char Subcommand)
```

Table SATA Self-Test Type

ID	Description
SMART_SHORT_SELF_TEST_OFFLINE_MODE	Short self-test offline mode
SMART_EXTENDED_SELF_TEST_OFFLINE_MODE	Extended self-test offline mode

Table NVMe Self-Test Type

ID	Description
NVME_SELF_TEST_SHORT_TEST	Short test
NVME_SELF_TEST_EXTENDED_TEST	Extended test
NVME_SELF_TEST_ABORT_TEST	Abort test

Description:

Execute SMART self-test. Support short self-test offline mode and extended self-test offline mode for SATA device; short test and extended test for NVMe device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

Subcommand

[out] Command of self-test. See [Table SATA Self-Test Type](#) and [Table NVMe Self-Test Type](#).

Return:

Refer to the section of status code.

SQFlash_ReadSMARTSelfTestStatus

Syntax:

```
int SQFlash_ReadSMARTSelfTestStatus(int index, uint8_t *progress, uint8_t *status)
```

Table Self-Test Status

ID	Description
SELF_TEST_STATUS_IN_PROGRESS	Self-Test in progress
SELF_TEST_STATUS_SUCCESS	Self-Test succeed

Description:

Read the executed status of the SMART self-test.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

progress

[out] Progress of the self-test.

status

[out] Status of the self-test. See [Table Self-Test Status](#).

Return:

Refer to the section of status code.

Crypto Erase Functions

SQFlash_QuickErase

Syntax:

int SQFlash_QuickErase(int index, int option)

Table Crypto Erase Option

ID	Description
QUICK_ERASE_OPTION_NONE	Do crypto erase
QUICK_ERASE_OPTION_CHECK_SUPPORT	Check support crypto erase or not

Description:

Do quick erase (a.k.a. crypto erase) to the SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

option

[in] Options for erase operation. See [Table Crypto Erase Option](#).

Return:

Refer to the section of status code.

Flash Lock V2 Functions

SQFlash_LockV2GetStatus

Syntax:

```
int SQFlash_LockV2GetStatus(int index, unsigned char *Status)
```

Table Lock Status

ID	Description
LOCK_STATUS_ENABLE	Flash lock is enable
LOCK_STATUS_DISABLE	Flash lock is disable

Description:

Get the status of LockV2 feature.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

Status

[out] Status of flash lock v2 feature in the SQFlash device. See [Table Lock Status](#).

Return:

Refer to the section of status code.

SQFlash_LockV2Enable

Syntax:

```
int SQFlash_LockV2Enable(int index, char *Password)
```

Description:

Enable LockV2 feature.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

Password

[in] Password for flash lock v2 feature to enable in the SQFlash device.

Return:

Refer to the section of status code.

SQFlash_LockV2Disable

Syntax:

```
int SQFlash_LockV2Disable(int index, char *Password)
```

Description:

Disable LockV2 feature.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

Password

[in] Password for flash lock v2 feature to disable in the SQFlash device.

Return:

Refer to the section of status code.

Security Initialize Functions

SQFlash_SendAccessCode

Syntax:

```
int SQFlash_SendAccessCode(int index, const char *SerialNumber)
```

Description:

Verify the correctness of access code to have the permission of controlling the SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

SerialNumber

[in] Serial number of the SQFlash device.

Return:

Refer to the section of status code.

SQFlash_SetAccessCode

Syntax:

```
int SQFlash_SetAccessCode(int index, const char *SerialNumber)
```

Description:

Set access code of the SQFlash device

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

SerialNumber

[in] Serial number of the SQFlash device.

Return:

Refer to the section of status code.

SQFlash_ChangeAccessCode

Syntax:

```
int SQFlash_ChangeAccessCode(int index, const char *orgsn, const char  
*newsn)
```

Description:

Change access code of the SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

orgsn

[in] Original serial number.

newsn

[in] New serial number.

Return:

Refer to the section of status code.

Security ID Functions

SQFlash_GetSecurityIDCap

Syntax:
int SQFlash_GetSecurityIDCap(int index, unsigned int Item, void *Content)

Table Security ID Capability

ID	Description
SID_CAP_ITEM_ID_MAX_LENGTH	Max length of security ID
SID_CAP_ITEM_ID2_MAX_LENGTH	Max length of security ID2

Description:
Get the maximum support length of security ID/ID2.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

Item

[in] Information item of security ID. See [Table Security ID Capability](#).

Content

[out] The return information of the specified item.

Return:
Refer to the section of status code.

SQFlash_SetSecurityID

Syntax:

```
int SQFlash_SetSecurityID(int index, const char *SecurityID)
```

Description:

Write Security ID into the hidden area of the SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

SecurityID

[in] Security ID of the SQFlash device.

Return:

Refer to the section of status code.

SQFlash_GetSecurityID

Syntax:

```
int SQFlash_GetSecurityID(int index, char *SecurityID, int Length)
```

Description:

Read Security ID from hidden area of the SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

SecurityID

[out] Buffer for the data of Security ID retrieved from SQFlash device.

Length

[in] Buffer size for the data of Security ID, the maximum size of the buffer can be retrieved with function SQFlash_GetSecurityIDCap().

Return:

Refer to the section of status code.

SQFlash_SetSecurityID2

Syntax:

```
int SQFlash_SetSecurityID2(int index, const char *SecurityID)
```

Description:

Write Security ID 2 into the SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

SecurityID

[in] Security ID2 of the SQFlash device.

Return:

Refer to the section of status code.

SQFlash_GetSecurityID2

Syntax:

```
int SQFlash_GetSecurityID2(int index, char *SecurityID, int Length)
```

Description:

Read Security ID 2 from the SQFlash device.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

SecurityID

[out] Buffer for the data of Security ID2 retrieved from SQFlash device.

Length

[in] Buffer size for the data of Security ID2, the maximum size of the buffer can be retrieved with function SQFlash_GetSecurityIDCap().

Return:

Refer to the section of status code.

Flash Vault Functions

SQFlash_SetVaultEnable

Syntax:

```
int SQFlash_SetVaultEnable(int index, char *Password)
```

Description:

Enable flash vault feature.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

Password

[in] Password for flash vault feature to enable in the SQFlash device.

Return:

Refer to the section of status code.

SQFlash_SetVaultDisable

Syntax:

```
int SQFlash_SetVaultDisable(int index, char *Password)
```

Description:

Disable flash vault feature.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

Password

[in] Password for flash vault feature to disable in the SQFlash device.

Return:

Refer to the section of status code.

SQFlash_SetVaultUnlock

Syntax:

```
int SQFlash_SetVaultUnlock(int index, char *Password)
```

Description:

Temporary unlock flash vault, the SQFlash device will be re-lock after power reset.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

Password

[in] Password for flash vault feature to temporary unlock in the SQFlash device, it will be re-locked after reset.

Return:

Refer to the section of status code.

SQFlash_GetVaultStatus

Syntax:

```
int SQFlash_GetVaultStatus(int index, unsigned char *Status)
```

Table Flash Vault Status

ID	Description
VAULT_STATUS_ENABLE	Flash vault enable
VAULT_STATUS_DISABLE	Flash vault disable
VAULT_STATUS_UNLOCKED	Temporary unlock, re-lock after power reset
VAULT_STATUS_CONFLICT	Conflict with other feature

Description:

Get the status of flash vault.

Parameters:

index

[in] Index of the specified SQFlash device in the disk list, range from 0 to total disk number – 1.

Status

[Out] Status of flash vault feature in the SQFlash device. See [Table Flash Vault Status](#).

Return:

Refer to the section of status code.

