



SCIGATE AUTOMATION (S) PTE LTD

No.1 Bukit Batok Street 22 #01-01 Singapore 659592

Tel: (65) 6561 0488

Fax: (65) 6562 0588

Email: [sales@scigate.com.sg](mailto:sales@scigate.com.sg)

Web: [www.scigate.com.sg](http://www.scigate.com.sg)

Business Hours: Monday - Friday 8.30am - 6.15pm

# Software API 4.0

# Library

## User's Manual

V1.0



# Contents

<b>CONTENTS</b> .....	<b>3</b>
<b>LIST OF TABLES</b> .....	<b>5</b>
<b>1 INTRODUCTION</b> .....	<b>6</b>
1.1 BENEFITS.....	6
1.2 ENVIRONMENT REQUIREMENTS.....	7
1.2.1 <i>Operating Systems</i> .....	7
<b>2 PROFACE SOFTWARE API DEFINITION</b> .....	<b>8</b>
2.1 STATUS CODES.....	8
2.2 ID.....	12
<b>3 PROFACE SOFTWARE AP</b> .....	<b>17</b>
3.1 INITIALIZATION FUNCTIONS.....	17
3.1.1 <i>SusiLibInitialize</i> .....	17
3.1.2 <i>SusiLibUninitialize</i> .....	17
3.2 INFORMATION FUNCTIONS.....	19
3.2.1 <i>SusiBoardGetValue</i> .....	19
3.2.2 <i>SusiBoardGetStringA</i> .....	21
3.3 BACKLIGHT FUNCTIONS.....	23
3.3.1 <i>SusiVgaGetCaps</i> .....	23
3.3.2 <i>SusiVgaGetBacklightEnable</i> .....	24
3.3.3 <i>SusiVgaSetBacklightEnable</i> .....	24
3.3.4 <i>SusiVgaGetBacklightBrightness</i> .....	25
3.3.5 <i>SusiVgaSetBacklightBrightness</i> .....	25
3.3.6 <i>SusiVgaGetBacklightLevel</i> .....	26
3.3.7 <i>SusiVgaSetBacklightLevel</i> .....	27
3.3.8 <i>SusiVgaGetPolarity</i> .....	27
3.3.9 <i>SusiVgaSetPolarity</i> .....	28
3.3.10 <i>SusiVgaGetFrequency</i> .....	28
3.3.11 <i>SusiVgaSetFrequency</i> .....	29
3.4 I2C FUNCTIONS.....	30
3.4.1 <i>SusiI2CGetCaps</i> .....	30
3.4.2 <i>SusiI2CWriteReadCombine</i> .....	31
3.4.3 <i>SusiI2CReadTransfer</i> .....	32

3.4.4	<i>SusiI2CWriteTransfer</i> .....	33
3.4.5	<i>SusiI2CProbeDevice</i> .....	34
3.4.6	<i>SusiI2CGetFrequency</i> .....	34
3.4.7	<i>SusiI2CSetFrequency</i> .....	35
3.5	SMBUS FUNCTIONS.....	36
3.5.1	<i>SusiSMBReadByte</i> .....	36
3.5.2	<i>SusiSMBWriteByte</i> .....	37
3.5.3	<i>SusiSMBReadWord</i> .....	38
3.5.4	<i>SusiSMBWriteWord</i> .....	38
3.5.5	<i>SusiSMBReceiveByte</i> .....	39
3.5.6	<i>SusiSMBSendByte</i> .....	40
3.5.7	<i>SusiSMBReadQuick</i> .....	41
3.5.8	<i>SusiSMBWriteQuick</i> .....	41
3.5.9	<i>SusiSMBReadBlock</i> .....	42
3.5.10	<i>SusiSMBWriteBlock</i> .....	43
3.5.11	<i>SusiSMBI2CReadBlock</i> .....	44
3.5.12	<i>SusiSMBI2CWriteBlock</i> .....	45
3.6	WATCHDOG FUNCTIONS.....	47
3.6.1	<i>SusiWDogGetCaps</i> .....	48
3.6.2	<i>SusiWDogStart</i> .....	49
3.6.3	<i>SusiWDogStop</i> .....	50
3.6.4	<i>SusiWDogTrigger</i> .....	50
3.6.5	<i>SusiWDogSetCallBack</i> .....	51
3.7	STORAGE FUNCTIONS.....	52
3.7.1	<i>SusiStorageGetCaps</i> .....	52
3.7.2	<i>SusiStorageAreaRead</i> .....	53
3.7.3	<i>SusiStorageAreaWrite</i> .....	53
3.7.4	<i>SusiStorageAreaSetLock</i> .....	54
3.7.5	<i>SusiStorageAreaSetUnlock</i> .....	55
3.8	THERMAL PROTECTION FUNCTIONS.....	56
3.8.1	<i>SusiThermalProtectionGetCaps</i> .....	57
3.8.2	<i>SusiThermalProtectionSetConfig</i> .....	58
3.8.3	<i>SusiThermalProtectionGetConfig</i> .....	58

## List of Tables

Table 1	Board information value ID .....	19
Table 2	Board voltage value ID.....	19
Table 3	Board temperature value ID.....	20
Table 4	Board support information value ID .....	20
Table 5	Board information string ID .....	21
Table 6	Backlight ID.....	23
Table 7	Backlight Enable Values .....	23
Table 8	VGA capabilities item Id.....	24
Table 9	Brightness level range definition .....	26
Table 10	Brightness polarity definition .....	28
Table 11	I <sup>2</sup> C ID.....	30
Table 12	I <sup>2</sup> C command encode .....	30
Table 13	I <sup>2</sup> C capabilities item Id .....	31
Table 14	SMBus ID.....	36
Table 15	Watchdog ID .....	48
Table 17	Watchdog capabilities item Id .....	48
Table 17	Watchdog Support Flags .....	50
Table 18	Watchdog timer event type .....	50
Table 19	Storage ID.....	52
Table 20	Storage capabilities item Id.....	52
Table 27	Storage Lock Status.....	53
Table 22	Thermal Protection Event Type.....	56
Table 23	Thermal Protection ID .....	56
Table 24	Thermal Protection capabilities item Id.....	57
Table 25	Thermal Protection Support Flags .....	57

## 1 Introduction

### **Proface Software API – A Bridge to Simplify & Enhance H/W & Application Implementation Efficiency**

When developers want to write an application that involves hardware access, they have to study the specifications to write the drivers. This is a time-consuming job and requires lots of expertise.

**Proface Software API** has done all the hard work for our customers with the release of a suite of Software APIs (Application Programming Interfaces), called **Proface Software API**.

**Proface Software API** provides not only the underlying drivers required but also a rich set of user-friendly, intelligent and integrated interfaces, which speeds development, enhances security and offers add-on value for Proface platforms. **Proface Software API** plays the role of catalyst between developer and solution, and makes Proface platforms easier and simpler to adopt and operate with customer applications.

### 1.1 Benefits

- ✓ **Faster Time to Market**  
**Proface Software API** helps developers write applications to control the hardware without knowing the hardware specs of the chipsets and driver architecture.
- ✓ **Reduced Project Effort**  
When customers have their own devices connected to the onboard bus, they can either: study the data sheet and write the driver & API from scratch, or they can use **Proface Software API** to start the integration with a 50% head start. Developers can reference the sample program on the CD to see and learn more about the software development environment.
- ✓ **Enhances Hardware Platform Reliability**  
**Proface Software API** provides a trusted custom ready solution which combines chipset and library function support, controlling application development through **Proface Software API** enhances reliability and brings peace of mind.
- ✓ **Flexible Upgrade Possibilities**  
**Proface Software API** supports an easy upgrade solution for customers. Customers just need to install the new version SUSI that supports the new functions.

## **1.2 Environment Requirements**

### **1.2.1 Operating Systems**

Windows 7 (x86 / x64)

WES7 (x86 / x64)

Windows 8.1 Desktop (x86 / x64)

## 2 Proface Software API definition

**Susi4.h** file includes the API declaration, constants and flags that are required for programming.

### 2.1 Status Codes

All **Proface Software API** functions immediately return a status code from a common list of possible errors. Any function may return any of the defined status codes. See the Appendix for more detailed information.

```
#define SUSI_STATUS_NOT_INITIALIZED 0xFFFFFFFF
```

#### Description

The **Proface Software API** library is not yet or unsuccessfully initialized. `SusiLibInitialize` needs to be called prior to the first access of any other **Proface Software API** functions.

#### Actions

Call `SusiLibInitialize`.

```
#define SUSI_STATUS_INITIALIZED 0xFFFFFFFFE
```

#### Description

Library is initialized.

#### Actions

None.

```
#define SUSI_STATUS_ALLOC_ERROR 0xFFFFFFFFD
```

#### Description

Memory Allocation Error.

#### Actions

Free memory and try again.

```
#define SUSI_STATUS_DRIVER_TIMEOUT 0xFFFFFFFFC
```

#### Description

Time out in driver. This is Normally caused by hardware/software semaphore timeout.

#### Actions

Retry.

```
#define SUSI_STATUS_INVALID_PARAMETER 0xFFFFFFFFF
```

#### Description



One or more of the **Proface Software API** functions call parameters are out of defined range.

## Actions

Verify Function Parameters.

```
#define SUSI_STATUS_INVALID_BLOCK_ALIGNMENT 0xFFFFFEFE
```

## Description

The Block Alignment is incorrect.

## Actions

Use Inputs and Outputs to correctly select input and outputs.

```
#define SUSI_STATUS_INVALID_BLOCK_LENGTH 0xFFFFFEFD
```

## Description

This means that the Block length is too long.

## Actions

Use Alignment Capabilities information to correctly align write access.

```
#define SUSI_STATUS_INVALID_DIRECTION 0xFFFFFEFC
```

## Description

The current Direction Argument attempts to set GPIOs to a unsupported direction. I.E. Setting GPI to Output.

## Actions

Use Inputs and Outputs to correctly select input and outputs.

```
#define SUSI_STATUS_INVALID_BITMASK 0xFFFFFEFB
```

## Description

The Bitmask Selects bits/GPIOs which are not supported for the current ID.

## Actions

Use Inputs and Outputs to probe supported bits.

```
#define SUSI_STATUS_RUNNING 0xFFFFFEFA
```

## Description

Watchdog timer already started.

## Actions

Call SusiWDogStop before retrying.

```
#define SUSI_STATUS_UNSUPPORTED 0xFFFFFCFF
```

## Description

This function or ID is not supported at the actual hardware environment.

## Actions

None.

```
#define SUSI_STATUS_NOT_FOUND 0xFFFFFBFF
```

## Description

Selected device was not found

## Actions

None.

```
#define SUSI_STATUS_TIMEOUT 0xFFFFBFE
```

## Description

Device has no response.

## Actions

None.

```
#define SUSI_STATUS_BUSY_COLLISION 0xFFFFBFD
```

## Description

The selected device or ID is busy or a data collision is detected.

## Actions

Retry.

```
#define SUSI_STATUS_READ_ERROR 0xFFFFFAFF
```

## Description

An error is detected during a read operation.

## Actions

Retry.

```
#define SUSI_STATUS_WRITE_ERROR 0xFFFFFAFE
```

## Description

An error is detected during a write operation.

## Actions

Retry.

```
#define SUSI_STATUS_MORE_DATA 0xFFFF9FF
```

## Description

The amount of available data exceeds the buffer size. Storage buffer overflow was prevented.  
Read count was larger than the defined buffer length.

## Actions

Either increase the buffer size or reduce the block length.

```
#define SUSI_STATUS_ERROR 0xFFFF0FF
```

## Description

Generic error message. No further error details are available.

## Actions

None.

```
#define SUSI_STATUS_SUCCESS 0
```

## Description

The operation is successful.

## Actions

None.

## 2.2 ID

```
#define SUSI_ID_UNKNOWN 0xFFFFFFFF
```

### Description

Undefined/Unknown ID

```
#define SUSI_ID_BOARD_MANUFACTURER_STR 0
```

```
#define SUSI_ID_BOARD_NAME_STR 1
```

```
#define SUSI_ID_BOARD_REVISION_STR 2
```

```
#define SUSI_ID_BOARD_SERIAL_STR 3
```

```
#define SUSI_ID_BOARD_BIOS_REVISION_STR 4
```

```
#define SUSI_ID_BOARD_HW_REVISION_STR 5
```

```
#define SUSI_ID_BOARD_PLATFORM_TYPE_STR 6
```

### Description

Board information string ID, use in SusiBoardGetStringA.

```
#define SUSI_ID_GET_SPEC_VERSION 0x00000000
```

```
#define SUSI_ID_BOARD_BOOT_COUNTER_VAL 0x00000001
```

```
#define SUSI_ID_BOARD_RUNNING_TIME_METER_VAL 0x00000002
```

```
#define SUSI_ID_BOARD_PNPID_VAL 0x00000003
```

```
#define SUSI_ID_BOARD_PLATFORM_REV_VAL 0x00000004
```

```
#define SUSI_ID_BOARD_DRIVER_VERSION_VAL 0x00010000
```

```
#define SUSI_ID_BOARD_LIB_VERSION_VAL 0x00010001
```

```
#define SUSI_ID_BOARD_FIRMWARE_VERSION_VAL 0x00010002
```

### Description

Board information value ID, use in SusiBoardGetValue.

```
#define SUSI_ID_HWM_TEMP_CPU 0x00020000
```

```
#define SUSI_ID_HWM_TEMP_CHIPSET 0x00020001
```

```
#define SUSI_ID_HWM_TEMP_SYSTEM 0x00020002
```

```
#define SUSI_ID_HWM_TEMP_CPU2 0x00020003
```

```
#define SUSI_ID_HWM_TEMP_OEM0 0x00020004
```

```
#define SUSI_ID_HWM_TEMP_OEM1 0x00020005
```

```
#define SUSI_ID_HWM_TEMP_OEM2 0x00020006
```

```
#define SUSI_ID_HWM_TEMP_OEM3 0x00020007
```

```
#define SUSI_ID_HWM_TEMP_OEM4 0x00020008
```

```
#define SUSI_ID_HWM_TEMP_OEM5 0x00020009
```

### Description

Board temperature value ID, use in SusiBoardGetValue.

<code>#define SUSI_ID_HWM_VOLTAGE_VCORE</code>	<code>0x00021000</code>
<code>#define SUSI_ID_HWM_VOLTAGE_VCORE2</code>	<code>0x00021001</code>
<code>#define SUSI_ID_HWM_VOLTAGE_2V5</code>	<code>0x00021002</code>
<code>#define SUSI_ID_HWM_VOLTAGE_3V3</code>	<code>0x00021003</code>
<code>#define SUSI_ID_HWM_VOLTAGE_5V</code>	<code>0x00021004</code>
<code>#define SUSI_ID_HWM_VOLTAGE_12V</code>	<code>0x00021005</code>
<code>#define SUSI_ID_HWM_VOLTAGE_5VSB</code>	<code>0x00021006</code>
<code>#define SUSI_ID_HWM_VOLTAGE_3VSB</code>	<code>0x00021007</code>
<code>#define SUSI_ID_HWM_VOLTAGE_VBAT</code>	<code>0x00021008</code>
<code>#define SUSI_ID_HWM_VOLTAGE_5NV</code>	<code>0x00021009</code>
<code>#define SUSI_ID_HWM_VOLTAGE_12NV</code>	<code>0x0002100A</code>
<code>#define SUSI_ID_HWM_VOLTAGE_VTT</code>	<code>0x0002100B</code>
<code>#define SUSI_ID_HWM_VOLTAGE_24V</code>	<code>0x0002100C</code>
<code>#define SUSI_ID_HWM_VOLTAGE_DC</code>	<code>0x0002100D</code>
<code>#define SUSI_ID_HWM_VOLTAGE_DCSTBY</code>	<code>0x0002100E</code>
<code>#define SUSI_ID_HWM_VOLTAGE_OEM3</code>	<code>0x0002100F</code>
<code>#define SUSI_ID_HWM_VOLTAGE_OEM0</code>	<code>0x00021010</code>
<code>#define SUSI_ID_HWM_VOLTAGE_OEM1</code>	<code>0x00021011</code>
<code>#define SUSI_ID_HWM_VOLTAGE_OEM2</code>	<code>0x00021012</code>
<code>#define SUSI_ID_HWM_VOLTAGE_OEM3</code>	<code>0x00021013</code>
<code>#define SUSI_ID_HWM_VOLTAGE_1V05</code>	<code>0x00021014</code>
<code>#define SUSI_ID_HWM_VOLTAGE_1V5</code>	<code>0x00021015</code>
<code>#define SUSI_ID_HWM_VOLTAGE_1V8</code>	<code>0x00021016</code>

### Description

Board voltage value ID, use in SusiBoardGetValue.

<code>#define SUSI_ID_HWM_FAN_CPU</code>	<code>0x00022000</code>
<code>#define SUSI_ID_HWM_FAN_SYSTEM</code>	<code>0x00022001</code>
<code>#define SUSI_ID_HWM_FAN_CPU2</code>	<code>0x00022002</code>
<code>#define SUSI_ID_HWM_FAN_OEM0</code>	<code>0x00022003</code>
<code>#define SUSI_ID_HWM_FAN_OEM1</code>	<code>0x00022004</code>
<code>#define SUSI_ID_HWM_FAN_OEM2</code>	<code>0x00022005</code>
<code>#define SUSI_ID_HWM_FAN_OEM3</code>	<code>0x00022006</code>
<code>#define SUSI_ID_HWM_FAN_OEM4</code>	<code>0x00022007</code>
<code>#define SUSI_ID_HWM_FAN_OEM5</code>	<code>0x00022008</code>

`#define SUSI_ID_HWM_FAN_OEM6` 0x00022009

## Description

Board fan speed value ID, use in SusiBoardGetValue and **Error! Reference source not found..**

`#define SUSI_ID_HWM_CURRENT_OEM0` 0x00023000

`#define SUSI_ID_HWM_CURRENT_OEM1` 0x00023001

`#define SUSI_ID_HWM_CURRENT_OEM2` 0x00023002

## Description

Board current value ID, use in SusiBoardGetValue.

`#define SUSI_ID_SMBUS_SUPPORTED` 0x00030000

`#define SUSI_ID_I2C_SUPPORTED` 0x00030100

## Description

Board supported information value ID, use in SusiBoardGetValue.

`#define SUSI_ID_SMBUS_EXTERNAL` 0

`#define SUSI_ID_SMBUS_OEM0` 1

`#define SUSI_ID_SMBUS_OEM1` 2

`#define SUSI_ID_SMBUS_OEM2` 3

`#define SUSI_ID_SMBUS_OEM3` 4

## Description

SMBus device ID, use in SMBus Functions.

`#define SUSI_ID_I2C_EXTERNAL` 0

`#define SUSI_ID_I2C_OEM0` 1

`#define SUSI_ID_I2C_OEM1` 2

`#define SUSI_ID_I2C_OEM2` 3

## Description

I<sup>2</sup>C device ID, use in I2C Functions.

`#define SUSI_ID_BACKLIGHT_1` 0

`#define SUSI_ID_BACKLIGHT_2` 1

`#define SUSI_ID_BACKLIGHT_3` 2

## Description

Backlight device ID, use in Backlight Functions.

`#define SUSI_ID_STORAGE_STD` 0x00000000

`#define SUSI_ID_STORAGE_OEM0` 0x00000001

`#define SUSI_ID_STORAGE_OEM1` 0x00000002

## Description

Storage device ID, use in Storage Functions.

`#define SUSI_ID_THERMAL_PROTECT_1` 0

`#define SUSI_ID_THERMAL_PROTECT_2` 1

`#define SUSI_ID_THERMAL_PROTECT_3` 2

`#define SUSI_ID_THERMAL_PROTECT_4` 3

## Description

Thermal protection device ID, use in Thermal Protection Functions

`#define SUSI_ID_WATCHDOG_1` 0

`#define SUSI_ID_WATCHDOG_2` 1

`#define SUSI_ID_WATCHDOG_3` 2

## Description

Watchdog device ID, use in Watchdog Functions.

Item ID

`#define SUSI_ID_I2C_MAXIMUM_BLOCK_LENGTH` 0x00000000

## Description

Watchdog capabilities item ID, use in SusiI2CGetCaps.

`#define SUSI_ID_VGA_BRIGHTNESS_MAXIMUM` 0x00010000

`#define SUSI_ID_VGA_BRIGHTNESS_MINIMUM` 0x00010001

## Description

VGA capabilities item ID, use in SusiVgaGetCaps.

`#define SUSI_ID_STORAGE_TOTAL_SIZE` 0x00000000

`#define SUSI_ID_STORAGE_BLOCK_SIZE` 0x00000001

`#define SUSI_ID_STORAGE_LOCK_STATUS` 0x00010000

`#define SUSI_ID_STORAGE_PSW_MAX_LEN` 0x00010000

## Description

Storage capabilities item ID, use in SusiStorageGetCaps.

`#define SUSI_ID_WDT_SUPPORT_FLAGS` 0x00000000

`#define SUSI_ID_WDT_DELAY_MAXIMUM` 0x00000001

`#define SUSI_ID_WDT_DELAY_MINIMUM` 0x00000002

`#define SUSI_ID_WDT_EVENT_MAXIMUM` 0x00000003

## Pro-face

<code>#define SUSI_ID_WDT_EVENT_MINIMUM</code>	<code>0x00000004</code>
<code>#define SUSI_ID_WDT_RESET_MAXIMUM</code>	<code>0x00000005</code>
<code>#define SUSI_ID_WDT_RESET_MINIMUM</code>	<code>0x00000006</code>
<code>#define SUSI_ID_WDT_UNIT_MINIMUM</code>	<code>0x0000000F</code>
<code>#define SUSI_ID_WDT_DELAY_TIME</code>	<code>0x00010001</code>
<code>#define SUSI_ID_WDT_EVENT_TIME</code>	<code>0x00010002</code>
<code>#define SUSI_ID_WDT_RESET_TIME</code>	<code>0x00010003</code>
<code>#define SUSI_ID_WDT_EVENT_TYPE</code>	<code>0x00010004</code>

### Description

Watchdog capabilities item ID, use in SusiWDogGetCaps.

<code>#define SUSI_ID_TP_EVENT_SUPPORT_FLAGS</code>	<code>0x00000000</code>
<code>#define SUSI_ID_TP_EVENT_TRIGGER_MAXIMUM</code>	<code>0x00000001</code>
<code>#define SUSI_ID_TP_EVENT_TRIGGER_MINIMUM</code>	<code>0x00000002</code>
<code>#define SUSI_ID_TP_EVENT_CLEAR_MAXIMUM</code>	<code>0x00000003</code>
<code>#define SUSI_ID_TP_EVENT_CLEAR_MINIMUM</code>	<code>0x00000004</code>

### Description

Thermal protection capabilities item ID, use in SusiThermalProtectionGetCaps.



## 3 Proface Software AP

The **Proface Software API** provide functions to control Proface platforms. **Proface Software API** functions are based on a dynamic library. They can be implemented in various other programming languages.

### 3.1 Initialization Functions

#### 3.1.1 SusiLibInitialize

```
uint32_t SUSI_API SusiLibInitialize(void)
```

**Description:**

General initialization of the PROFACE SOFTWARE API . Prior to calling any **Proface Software API** functions, the library needs to be initialized by calling this function. The status code for all **Proface Software API** function will be SUSI\_STATUS\_NOT\_INITIALIZED unless this function is called.

**Parameters:**

None

**Return Status Code:**

Condition	Return Value
Library initialized	SUSI_STATUS_INITIALIZED
Fail	SUSI_STATUS_NOT_INITIALIZED
Success	SUSI_STATUS_SUCCESS

#### 3.1.2 SusiLibUninitialize

```
uint32_t SUSI_API SusiLibUninitialize(void)
```

**Description:**

General function to uninitialize the **Proface Software API** library that should be called before program exit. In a dynamic library environment this function is not expected to replace the native uninitialized routines. It is expected that in this environments this function has no functionality.

**Parameters:**

None

**Return Status Code:**

---

<b>Condition</b>	<b>Return Value</b>
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Success	SUSI_STATUS_SUCCESS

---

## 3.2 Information Functions

### 3.2.1 SusiBoardGetValue

```
uint32_t SUSI_API SusiBoardGetValue(uint32_t Id, uint32_t *pValue)
```

**Description:**

Getting information about the hardware platform in value format.

**Parameters:**

**Id**

Selects target value. See **Table 1** to **Table 4**.

**pValue**

Pointer to a buffer that receives the value's data.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pValue==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

**Table 1 Board information value ID**

Id	Description	Unit
SUSI_ID_GET_SPEC_VERSION	API Specification Version	
SUSI_ID_BOARD_BOOT_COUNTER_VAL	Boot Counter	boot
SUSI_ID_BOARD_RUNNING_TIME_METER_VAL	Running Time Meter	hour
SUSI_ID_BOARD_PNPID_VAL	Board Vendor PNPID	
SUSI_ID_BOARD_PLATFORM_REV_VAL	Platform revision	
SUSI_ID_BOARD_DRIVER_VERSION_VAL	Driver version	
SUSI_ID_BOARD_LIB_VERSION_VAL	Library version	
SUSI_ID_BOARD_FIRMWARE_VERSION_VAL	Firmware version	

**Table 2 Board voltage value ID**

Id	Description	Unit
----	-------------	------

SUSI_ID_HWM_VOLTAGE_VCORE	CPU Core voltage	millivolt
SUSI_ID_HWM_VOLTAGE_VCORE2	Second CPU Core voltage	millivolt
SUSI_ID_HWM_VOLTAGE_2V5	2.5V	millivolt
SUSI_ID_HWM_VOLTAGE_3V3	3.3V	millivolt
SUSI_ID_HWM_VOLTAGE_5V	5V	millivolt
SUSI_ID_HWM_VOLTAGE_12V	12V	millivolt
SUSI_ID_HWM_VOLTAGE_5VSB	5V Standby	millivolt
SUSI_ID_HWM_VOLTAGE_3VSB	3V Standby	millivolt
SUSI_ID_HWM_VOLTAGE_VBAT	CMOS Battery voltage	millivolt
SUSI_ID_HWM_VOLTAGE_5NV	-5V	millivolt
SUSI_ID_HWM_VOLTAGE_12NV	-12V	millivolt
SUSI_ID_HWM_VOLTAGE_VTT	DIMM voltage	millivolt
SUSI_ID_HWM_VOLTAGE_24V	24V	millivolt
SUSI_ID_HWM_VOLTAGE_OEM0~3	Other voltages	millivolt

**Table 3 Board temperature value ID**

<b>Id</b>	<b>Description</b>	<b>Unit</b>
SUSI_ID_HWM_TEMP_CPU	CPU temperature	0.1 Kelvin
SUSI_ID_HWM_TEMP_CHIPSET	Chipset temperature	0.1 Kelvin
SUSI_ID_HWM_TEMP_SYSTEM	System temperature	0.1 Kelvin
SUSI_ID_HWM_TEMP_CPU2	CPU2 temperature	0.1 Kelvin
SUSI_ID_HWM_TEMP_OEM0~5	Other temperatures	0.1 Kelvin

**Table 4 Board support information value ID**

<b>Id</b>	<b>Description</b>
SUSI_ID_SMBUS_SUPPORTED	Mask flags:
	SUSI_SMBUS_EXTERNAL_SUPPORTED
	SUSI_SMBUS_OEM0_SUPPORTED
	SUSI_SMBUS_OEM1_SUPPORTED
	SUSI_SMBUS_OEM2_SUPPORTED
SUSI_ID_I2C_SUPPORTED	Mask flags:
	SUSI_I2C_EXTERNAL_SUPPORTED
	SUSI_I2C_OEM0_SUPPORTED
	SUSI_I2C_OEM1_SUPPORTED
	SUSI_I2C_OEM2_SUPPORTED

### 3.2.2 SusiBoardGetStringA

```
uint32_t SUSI_API SusiBoardGetStringA(uint32_t Id, char *pBuffer, uint32_t *pBufLen)
```

**Description:**

Text information about the hardware platform.

**Parameters:**

**Id**

Selects target string. See **Table 5**.

**pBuffer**

Pointer to a buffer that receives the value's data.

**pBufLen**

Pointer to a variable that specifies the size, in bytes, of the buffer pointed to by the pBuffer parameter. When the function returns, this variable contains the size of the data copied to pBuffer including the terminating null character.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBufLen==NULL	SUSI_STATUS_INVALID_PARAMETER
pBufLen!=NULL&&*pBufLen&&pBuffer==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
strlen + 1 > *pBufLen	SUSI_STATUS_MORE_DATA
Success	SUSI_STATUS_SUCCESS

**Table 5 Board information string ID**

Id	Description
SUSI_ID_BOARD_MANUFACTURER_STR	Board Manufacturer Name
SUSI_ID_BOARD_NAME_STR	Board Name
SUSI_ID_BOARD_REVISION_STR	Board Revision
SUSI_ID_BOARD_SERIAL_STR	Board Serial Number
SUSI_ID_BOARD_BIOS_REVISION_STR	Board BIOS Revision
SUSI_ID_BOARD_HW_REVISION_STR	Hardware Revision
SUSI_ID_BOARD_PLATFORM_TYPE_STR	Platform type



### 3.3 Backlight Functions

This function sub set facilitates backlight control for Integrated flat panel displays, typically LVDS.

**Table 6 Backlight ID**

Id	Description
SUSI_ID_BACKLIGHT_1	Backlight Local Flat Panel 1
SUSI_ID_BACKLIGHT_2	Backlight Local Flat Panel 2
SUSI_ID_BACKLIGHT_3	Backlight Local Flat Panel 3

**Table 7 Backlight Enable Values**

Name	Description
SUSI_BACKLIGHT_SET_ON	Signifies that the Backlight be Enabled
SUSI_BACKLIGHT_SET_OFF	Signifies that the Backlight be Disabled

#### 3.3.1 SusiVgaGetCaps

```
uint32_t SUSI_API SusiVgaGetCaps(uint32_t Id, uint32_t ItemId, uint32_t *pValue);
```

**Description:**

Gets VGA capabilities.

**Parameters:**

**Id**

Selects target device. See **Table 6**.

**ItemId**

Selects target capability. See **Table 8**.

**pValue**

Pointer to a buffer that receives the target capability.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pValue==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id or ItemId	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

**Table 8 VGA capabilities item Id**

Item Id	Description
SUSI_ID_VGA_BRIGHTNESS_MAXIMUM	Maximum backlight value
SUSI_ID_VGA_BRIGHTNESS_MINIMUM	Minimum backlight value

### 3.3.2 SusiVgaGetBacklightEnable

```
uint32_t SUSI_API SusiVgaGetBacklightEnable(uint32_t Id, uint32_t *pEnable)
```

**Description:**

Gets current Backlight Enable state for specified Flat Panel.

**Parameters:**

**Id**

Selects target device. See **Table 6**.

**pEnable**

Pointer to a buffer that receives the current backlight enables state. See **Table 7**.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pEnable==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.3.3 SusiVgaSetBacklightEnable

```
uint32_t SUSI_API SusiVgaSetBacklightEnable(uint32_t Id, uint32_t Enable)
```

**Description:**

Enables or disable the backlight of the selected flat panel display

**Parameters:**

**Id**

Selects target device. See **Table 6**.

**Enable**

Backlight Enable options. See **Table 7**.



**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.3.4 SusiVgaGetBacklightBrightness

```
uint32_t SUSI_API SusiVgaGetBacklightBrightness(uint32_t Id, uint32_t *pBright)
```

**Description:**

Reads the current brightness of the selected flat panel display.

**Parameters:**

**Id**

Selects target device. See **Table 6**.

**pBright**

Pointer to a buffer that receives the current backlight brightness value.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBright==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.3.5 SusiVgaSetBacklightBrightness

```
uint32_t SUSI_API SusiVgaSetBacklightBrightness(uint32_t Id, uint32_t Bright)
```

**Description:**

Reads the current brightness of the selected flat panel display.

**Parameters:**

**Id**

Selects target device. See **Table 6**.

**Bright**

Backlight Brightness value.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Bright > MAX value    Bright < MIN value	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.3.6 SusiVgaGetBacklightLevel

```
uint32_t SUSI_API SusiVgaGetBacklightLevel(uint32_t Id, uint32_t *pLevel)
```

**Description:**

Reads the current brightness level of the selected flat panel display.

**Parameters:**

**Id**

Selects target device. See **Table 6**.

**pLevel**

Pointer to a buffer that receives the current backlight brightness level. See **Table 9**.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pLevel==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

**Table 9 Brightness level range definition**

Name	Description
SUSI_BACKLIGHT_LEVEL_MAXIMUM	Maximum backlight level is 9
SUSI_BACKLIGHT_LEVEL_MINIMUM	Minimum backlight level is 0

### 3.3.7 SusiVgaSetBacklightLevel

```
uint32_t SUSI_API SusiVgaSetBacklightLevel(uint32_t Id, uint32_t Level)
```

**Description:**

Sets the brightness level of the selected flat panel display.

**Parameters:**

**Id**

Selects target device. See Table 6.

**Level**

Backlight Brightness level. See Table 9.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Level>SUSI_BACKLIGHT_LEVEL_MAXIMUM	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.3.8 SusiVgaGetPolarity

```
uint32_t SUSI_API SusiVgaGetPolarity(uint32_t Id, uint32_t *pPolarity)
```

**Description:**

Reads the current backlight polarity of the selected flat panel display.

**Parameters:**

**Id**

Selects target device. See Table 6.

**pPolarity**

Pointer to a buffer that receives the current backlight polarity. See Table 10.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pPolarity==NULL	SUSI_STATUS_INVALID_PARAMETER

Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

**Table 10 Brightness polarity definition**

Name	Description
SUSI_BACKLIGHT_POLARITY_ON	Backlight signal polarity ON
SUSI_BACKLIGHT_POLARITY_OFF	Backlight signal polarity OFF

### 3.3.9 SusiVgaSetPolarity

```
uint32_t SUSI_API SusiVgaSetPolarity(uint32_t Id, uint32_t Polarity)
```

**Description:**

Sets the polarity of the selected flat panel display.

**Parameters:**

**Id**

Selects target device. See **Table 6**.

**Polarity**

Polarity state. See **Table 10**.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.3.10 SusiVgaGetFrequency

```
uint32_t SUSI_API SusiVgaGetFrequency(uint32_t Id, uint32_t *pFrequency)
```

**Description:**

Reads the current backlight frequency of the selected flat panel display.

**Parameters:**

**Id**

Selects target device. See **Table 6**.

**pFrequency**

Pointer to a buffer that receives the current backlight frequency. (Unit: Hz)

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pFrequency==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.3.11 SusiVgaSetFrequency

`uint32_t SUSI_API SusiVgaSetFrequency(uint32_t Id, uint32_t Frequency)`

**Description:**

Sets the frequency of the selected flat panel display.

**Parameters:**

**Id**

Selects target device. See **Table 6**.

**Polarity**

Frequency value. (Unit: Hz)

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.4 I2C Functions

I<sup>2</sup>C APIs support standard 7 and 10 bits slave address mode. I<sup>2</sup>C APIs also support word command, it needs encode before set to parameter, see Table 12

**Table 11 I<sup>2</sup>C ID**

Id	Description
SUSI_ID_I2C_EXTERNAL	Main I <sup>2</sup> C host device
SUSI_ID_I2C_OEM0~2	Other I <sup>2</sup> C host devices

**Table 12 I<sup>2</sup>C command encode**

Type	Description
Standard command	Byte command
Extend command	Word command   0x80000000 Ex. 0x8000FABC
No command	0x4000xxxx, ignore command parameter

#### 3.4.1 Susil2CGetCaps

```
uint32_t SUSI_API Susil2CGetCaps(uint32_t Id, uint32_t ItemId, uint32_t *pValue)
```

**Description:**

Gets I<sup>2</sup>C capabilities.

**Parameters:**

**Id**

Selects target device. See **Table 11**.

**ItemId**

Selects target capability. See **Table 13**.

**pValue**

Pointer to a buffer that receives the target capability.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pValue==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id or ItemId	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

**Table 13 I<sup>2</sup>C capabilities item Id**

Item Id	Description
SUSI_ID_I2C_MAXIMUM_BLOCK_LENGTH	I <sup>2</sup> C maximum block length

### 3.4.2 Susil2CWriteReadCombine

```
uint32_t SUSI_API Susil2CWriteReadCombine(uint32_t Id, uint8_t Addr,
uint8_t *pWBuffer, uint32_t WriteLen, uint8_t *pRBuffer, uint32_t ReadLen)
```

**Description:**

Universal function for read and write operations to the I<sup>2</sup>C bus.

**Parameters:**

**Id**

Selects target device. See **Table 13**.

**Addr**

First byte of I<sup>2</sup>C device address. 7-bit address only.

**pWBuffer**

Pointer to a buffer containing the data to be transferred. This parameter can be NULL if the data is not required.

**WriteLen**

Size in bytes of the information pointed to by the pWBuffer parameter. If pWBuffer is NULL this will be ignored.

**pRBuffer**

Pointer to a buffer that receives the read data. This parameter can be NULL if the data is not required.

**ReadLen**

Size in bytes of the buffer pointed to by the pRBuffer parameter. If pRBuffer is NULL this will be ignored.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
(WriteLen>1)&&(pWBuffer==NULL)	SUSI_STATUS_INVALID_PARAMETER
(RBufLen>1)&&(pRBuffer==NULL)	SUSI_STATUS_INVALID_PARAMETER
(WriteLen==0)&&(RBufLen==0)	SUSI_STATUS_INVALID_PARAMETER

Unknown Id	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.4.3 Susil2CReadTransfer

```
uint32_t SUSI_API Susil2CReadTransfer(uint32_t Id, uint32_t Addr, uint32_t Cmd,
uint8_t *pBuffer, uint32_t ReadLen)
```

**Description:**

Reads from a specific register in the selected I<sup>2</sup>C device. Reads from I<sup>2</sup>C device at the I<sup>2</sup>C address Addr the amount of ReadLen bytes to the buffer pBuffer while using the device specific command Cmd. Depending on the addressed I<sup>2</sup>C device Cmd can be a specific command or a byte offset.

**Parameters:**

**Id**

Selects target device. See **Table 13**.

**Addr**

Encoded 7/10 Bit I<sup>2</sup>C Device Address.

**Cmd**

Encoded I<sup>2</sup>C Device Command / Index. See **Table 12**.

**pBuffer**

Pointer to a buffer that receives the read data.

**ReadLen**

Size in bytes of the buffer pointed to by the pBuffer parameter.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBuffer==NULL    ReadLen==0	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED



Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.4.4 Susil2CWriteTransfer

```
uint32_t SUSI_API Susil2CWriteTransfer(uint32_t Id, uint32_t Addr, uint32_t Cmd,
uint8_t *pBuffer, uint32_t ByteCnt)
```

**Description:**

Write to a specific register in the selected I<sup>2</sup>C device. Writes to an I<sup>2</sup>C device at the I<sup>2</sup>C address Addr the amount of ByteCnt bytes from the buffer \*pBuffer while using the device specific command Cmd. Depending on the addressed I<sup>2</sup>C device Cmd can be a specific command or a byte offset

**Parameters:**

**Id**

Selects target device. See **Table 13**.

**Addr**

Encoded 7/10 Bit I<sup>2</sup>C Device Address.

**Cmd**

Encoded I<sup>2</sup>C Device Command / Index. See **Table 12**.

**pBuffer**

Pointer to a buffer that receives the write data.

**ByteCnt**

Size in bytes of the buffer pointed to by the pBuffer parameter.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBuffer==NULL    ByteCnt==0	SUSI_STATUS_INVALID_PARAMETER
ByteCnt > MaxLength	SUSI_STATUS_INVALID_BLOCK_LENGTH
Unknown Id	SUSI_STATUS_UNSUPPORTED

Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.4.5 Susil2CProbeDevice

```
uint32_t SUSI_API Susil2CProbeDevice(uint32_t Id, uint32_t Addr)
```

**Description:**

Probes I<sup>2</sup>C address to test I<sup>2</sup>C device present.

**Parameters:**

**Id**

Selects target device. See **Table 13**.

**Addr**

Encoded 7/10 Bit I<sup>2</sup>C Device Address.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.4.6 Susil2CGetFrequency

```
uint32_t SUSI_API Susil2CGetFrequency(uint32_t Id, uint32_t *pFreq)
```

**Description:**

Get I<sup>2</sup>C clock frequency.

**Parameters:**

**Id**

Selects target device. See **Table 13**.

**pFreq**

Pointer to a buffer that receives the I<sup>2</sup>C clock frequency value. (Unit: Hz)

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
pFreq == NULL	SUSI_STATUS_INVALID_PARAMETER
Success	SUSI_STATUS_SUCCESS

### 3.4.7 Susil2CSetFrequency

```
uint32_t SUSI_API Susil2CSetFrequency(uint32_t Id, uint32_t Freq)
```

**Description:**

Set I<sup>2</sup>C clock frequency.

**Parameters:**

**Id**

Selects target device. See **Table 13**.

**Freq**

I<sup>2</sup>C clock frequency value. (Unit: Hz)

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.5 SMBus Functions

SMBus is the System Management Bus defined by Intel® Corporation in 1995. It is used in personal computers and servers for low-speed system management communications.

**Table 14 SMBus ID**

Id	Description
SUSI_ID_SMBUS_EXTERNAL	Main SMBus host device
SUSI_ID_SMBUS_OEM0~3	Other SMBus host devices

#### 3.5.1 SusiSMBReadByte

```
uint32_t SUSI_API SusiSMBReadByte(uint32_t Id, uint8_t Addr, uint8_t Cmd, uint8_t *pData)
```

**Description:**

Read a byte of data from the target slave device in the SMBus.

**Parameters:**

**Id**

Selects target device. See **Table 14**.

**Addr**

Specifies the 8-bit device address, ranging from 0x00 to 0xFF. Whether to give a 1 (read) or 0 (write) to the LSB of slave address could be ignored.

**Cmd**

Specifies the offset or command of the device register to read data from.

**pData**

Pointer to a variable in which the function reads the byte data.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pData==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION

SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.5.2 SusiSMBWriteByte

`uint32_t SUSI_API SusiSMBWriteByte(uint32_t Id, uint8_t Addr, uint8_t Cmd, uint8_t Data)`

**Description:**

Write a byte of data to the target slave device in the SMBus.

**Parameters:**

**Id**

Selects target device. See **Table 14**.

**Addr**

Specifies the 8-bit device address, ranging from 0x00 to 0xFF. Whether to give a 1 (read) or 0 (write) to the LSB of slave address could be ignored.

**Cmd**

Specifies the offset or command of the device register to write data to.

**Data**

Specifies the byte data to be written.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.5.3 SusiSMBReadWord

```
uint32_t SUSI_API SusiSMBReadWord(uint32_t Id, uint8_t Addr, uint8_t Cmd, uint16_t *pData)
```

**Description:**

Read a word of data from the target slave device in the SMBus.

**Parameters:**

**Id**

Selects target device. See **Table 14**.

**Addr**

Specifies the 8-bit device address, ranging from 0x00 to 0xFF. Whether to give a 1 (read) or 0 (write) to the LSB of slave address could be ignored.

**Cmd**

Specifies the offset or command of the device register to read data from.

**pData**

Pointer to a variable in which the function reads the word data.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pData==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.5.4 SusiSMBWriteWord

```
uint32_t SUSI_API SusiSMBWriteWord(uint32_t Id, uint8_t Addr, uint8_t Cmd, uint16_t Data)
```

**Description:**

Write a word of data to the target slave device in the SMBus.

**Parameters:**

**Id**

Selects target device. See **Table 14**.

**Addr**

Specifies the 8-bit device address, ranging from 0x00 to 0xFF. Whether to give a 1 (read) or 0 (write) to the LSB of slave address could be ignored.

**Cmd**

Specifies the offset or command of the device register to write data to.

**Data**

Specifies the word data to be written.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.5.5 SusiSMBReceiveByte

```
uint32_t SUSI_API SusiSMBReceiveByte(uint32_t Id, uint8_t Addr, uint8_t *pData)
```

**Description:**

Receive a byte of data from the target slave device in the SMBus.

**Parameters:**

**Id**

Selects target device. See **Table 14**.

**Addr**

Specifies the 8-bit device address, ranging from 0x00 to 0xFF. Whether to give a 1 (read) or

0 (write) to the LSB of slave address could be ignored.

**pData**

Pointer to a variable in which the function receive the byte data.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBuffer==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.5.6 SusiSMBSendByte

```
uint32_t SUSI_API SusiSMBSendByte(uint32_t Id, uint8_t Addr, uint8_t Data)
```

**Description:**

Send a byte of data to the target slave device in the SMBus.

**Parameters:**

**Id**

Selects target device. See **Table 14**.

**Addr**

Specifies the 8-bit device address, ranging from 0x00 to 0xFF. Whether to give a 1 (read) or 0 (write) to the LSB of slave address could be ignored.

**Data**

Specifies the word data to be sent.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED



Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.5.7 SusiSMBReadQuick

```
uint32_t SUSI_API SusiSMBReadQuick(uint32_t Id, uint8_t Addr)
```

**Description:**

Turn SMBus device function off (on) or disable (enable) a specific device mode.

**Parameters:**

**Id**

Selects target device. See **Table 14**.

**Addr**

Specifies the 8-bit device address, ranging from 0x00 to 0xFF. Whether to give a 1 (read) or 0 (write) to the LSB of slave address could be ignored.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Success	SUSI_STATUS_SUCCESS

### 3.5.8 SusiSMBWriteQuick

```
uint32_t SUSI_API SusiSMBWriteQuick(uint32_t Id, uint8_t Addr)
```

**Description:**

Turn SMBus device function off (on) or disable (enable) a specific device mode.

**Parameters:**

**Id**

Selects target device. See **Table 14**.

**Addr**

Specifies the 8-bit device address, ranging from 0x00 to 0xFF. Whether to give a 1 (read) or 0 (write) to the LSB of slave address could be ignored.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Success	SUSI_STATUS_SUCCESS

### 3.5.9 SusiSMBReadBlock

```
uint32_t SUSI_API SusiSMBReadBlock(uint32_t Id, uint8_t Addr, uint8_t Cmd,
uint8_t *pBuffer, uint32_t *pLength)
```

**Description:**

Read multi-data from the target slave device in the SMBus.

**Parameters:**

**Id**

Selects target device. See **Table 14**.

**Addr**

Specifies the 8-bit device address, ranging from 0x00 to 0xFF. Whether to give a 1 (read) or 0 (write) to the LSB of slave address could be ignored.

**Cmd**

Specifies the offset or command of the device register to read data from.

**pBuffer**

Pointer to a byte array in which the function reads the block data.

**pLength**

Pointer to a byte in which specifies the number of bytes to be read and also return succeed bytes.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBuffer==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.5.10 SusiSMBWriteBlock

```
uint32_t Susi_API SusiSMBWriteBlock(uint32_t Id, uint8_t Addr, uint8_t Cmd,
uint8_t *pBuffer, uint32_t Length)
```

**Description:**

Write multi-data from the target slave device in the SMBus.

**Parameters:**

**Id**

Selects target device. See **Table 14**.

**Addr**

Specifies the 8-bit device address, ranging from 0x00 to 0xFF. Whether to give a 1 (read) or 0 (write) to the LSB of slave address could be ignored.

**Cmd**

Specifies the offset or command of the device register to write data to.

**pBuffer**

Pointer to a byte array in which the function writes the block data.

**Length**

Specifies the number of bytes to be write.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBuffer==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

**3.5.11 SusiSMBI2CReadBlock**

```
uint32_t SUSI_API SusiSMBI2CReadBlock(uint32_t Id, uint8_t Addr, uint8_t Cmd,
uint8_t *pBuffer, uint32_t *pLength)
```

**Description:**

Read multi-data using I<sup>2</sup>C block protocol from the target slave device in the SMBus.

**Parameters:**

**Id**

Selects target device. See **Table 14**.

**Addr**

Specifies the 8-bit device address, ranging from 0x00 to 0xFF. Whether to give a 1 (read) or 0 (write) to the LSB of slave address could be ignored.

**Cmd**

Specifies the offset or command of the device register to read data from.

**pBuffer**

Pointer to a byte array in which the function reads the block data.

**pLength**

Pointer to a byte in which specifies the number of bytes to be read and also return succeed

bytes.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBuffer==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.5.12 SusiSMBI2CWriteBlock

```
uint32_t SUSI_API SusiSMBI2CWriteBlock(uint32_t Id, uint8_t Addr, uint8_t Cmd,
uint8_t *pBuffer, uint32_t Length)
```

**Description:**

Write multi-data using I<sup>2</sup>C block protocol from the target slave device in the SMBus.

**Parameters:**

**Id**

Selects target device. See **Table 14**.

**Addr**

Specifies the 8-bit device address, ranging from 0x00 to 0xFF. Whether to give a 1 (read) or 0 (write) to the LSB of slave address could be ignored.

**Cmd**

Specifies the offset or command of the device register to write data to.

**pBuffer**

Pointer to a byte array in which the function writes the block data.

**Length**

Specifies the number of bytes to be write.

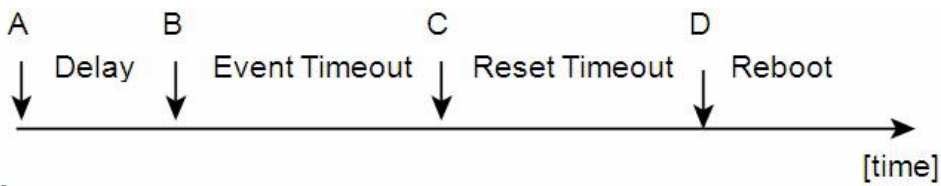
**Return Status Code:**

<b>Condition</b>	<b>Return Value</b>
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBuffer==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device unsupported	SUSI_STATUS_UNSUPPORTED
Bus Busy SDA/SDC low	SUSI_STATUS_BUSY_COLLISION
Arbitration Error/Collision Error	
On Write 1 write cycle	SUSI_STATUS_BUSY_COLLISION
SDA Remains low	
Time-out due to clock stretching	SUSI_STATUS_TIMEOUT
Address Non-ACK	SUSI_STATUS_NOT_FOUND
Write Non-ACK	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

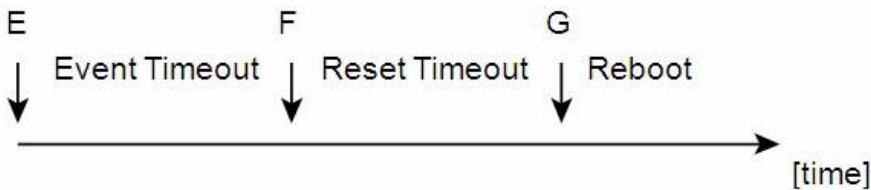
### 3.6 Watchdog Functions

After the watchdog timer has been start function it must be triggered within (Delay + Event Timeout) milliseconds as set with the start function, following the initial trigger every subsequent trigger must occur within (Event Timeout) milliseconds. Should trigger not be called within the relevant time limit a system reset will occur. The PROFACE SOFTWARE API watchdog timer may support two stages. If the watchdog is not triggered within the event timeout, an NMI, IRQ, or hardware output will be generated. Then the reset timeout becomes active. If the watchdog timer is not triggered within the reset timeout a reset will be generated

**Initial timing:**



**Timing after trigger:**



**Where:**

**Stage A**

Watchdog is started.

**Stage B**

Initial Delay Period is exhausted.

**Stage C/F**

Event is triggered, NMI, IRQ, or PIN is Triggered. To Allow for possible Software Recovery.

**Stage D/G**

System is reset.

**Stage E**

- Watchdog is Triggered.
- Trigger / Stop must be called before Stage C/F to prevent event from being generated.
- Trigger / Stop must be called before Stage D/G to prevent The system from being reset.

**Table 15 Watchdog ID**

Id	Description
SUSI_ID_WATCHDOG_1	First watchdog timer
SUSI_ID_WATCHDOG_2	Second watchdog timer
SUSI_ID_WATCHDOG_3	Third watchdog timer

### 3.6.1 SusiWDogGetCaps

```
uint32_t SUSI_API SusiWDogGetCaps(uint32_t Id, uint32_t ItemId, uint32_t *pValue)
```

**Description:**

Gets watchdog capabilities.

**Parameters:**

**Id**

Selects target device. See **Table 15**.

**ItemId**

Selects target capability. See **Table 16**.

**pValue**

Pointer to a buffer that receives the target capability.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pValue==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id or ItemId	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

**Table 16 Watchdog capabilities item Id**

Item Id	Description
SUSI_ID_WDT_SUPPORT_FLAGS	Event support flags ( <b>Table 17</b> )
SUSI_ID_WDT_DELAY_MAXIMUM	The maximum delay time value
SUSI_ID_WDT_DELAY_MINIMUM	The minimum delay time value
SUSI_ID_WDT_EVENT_MAXIMUM	The maximum event time value
SUSI_ID_WDT_EVENT_MINIMUM	The minimum event time value
SUSI_ID_WDT_RESET_MAXIMUM	The maximum reset time value



SUSI_ID_WDT_RESET_MINIMUM	The minimum reset time value
SUSI_ID_WDT_UNIT_MINIMUM	The minimum unit value
SUSI_ID_WDT_DELAY_TIME	Current delay time setting
SUSI_ID_WDT_EVENT_TIME	Current event time setting
SUSI_ID_WDT_RESET_TIME	Current reset time setting
SUSI_ID_WDT_EVENT_TYPE	Current event type ( <b>Table 18</b> )

### 3.6.2 SusiWDogStart

```
uint32_t SUSI_API SusiWDogStart(uint32_t Id, uint32_t DelayTime, uint32_t EventTime,
uint32_t ResetTime, uint32_t EventType)
```

**Description:**

Start the watchdog timer and set the parameters. To adjust the parameters, the watchdog must be stopped and then start again with the new values. If the hardware implementation of the watchdog timer does not allow a setting at the exact time selected, the PROFACE SOFTWARE API selects the next possible longer timing.

**Parameters:**

**Id**

Selects target device. See **Table 15**.

**DelayTime**

Initial delay for the watchdog timer in milliseconds.

**EventTime**

Watchdog timeout interval in milliseconds to trigger an event.

**ResetTime**

Watchdog timeout interval in milliseconds to trigger a reset.

**EventType**

To select one kind of event type. See **Table 18**.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Wrong time range	SUSI_STATUS_INVALID_PARAMETER
Success	SUSI_STATUS_SUCCESS

**Table 17 Watchdog Support Flags**

Flag Name	Description	Value
SUSI_WDT_FLAG_SUPPORT_IRQ	Support IRQ event	0x01
SUSI_WDT_FLAG_SUPPORT_SCI	Support SCI event	0x02
SUSI_WDT_FLAG_SUPPORT_PWRBTN	Support power button event	0x04

**Table 18 Watchdog timer event type**

Event Type	Description
SUSI_WDT_EVENT_TYPE_NONE	No event
SUSI_WDT_EVENT_TYPE_SCI	SCI event
SUSI_WDT_EVENT_TYPE_IRQ	IRQ event
SUSI_WDT_EVENT_TYPE_PWRBTN	Power button event

### 3.6.3 SusiWDogStop

```
uint32_t SUSI_API SusiWDogStop(uint32_t Id)
```

**Description:**

Stops the operation of the watchdog timer.

**Parameters:**

**Id**

Selects target device. See **Table 15**.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.6.4 SusiWDogTrigger

```
uint32_t SUSI_API SusiWDogTrigger(uint32_t Id)
```

**Description:**

Trigger the watchdog timer.

**Parameters:**

**Id**

Selects target device. See **Table 15**.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.6.5 SusiWDogSetCallback

```
uint32_t SUSI_API SusiWDogSetCallback(uint32_t Id,
SUSI_WDT_INT_CALLBACK pfnCallback, void *Context)
```

**Description:**

The call back function pointer can be transmit from Application when IRQ triggered.

**Parameters:**

**Id**

Selects target device. See **Table 15**.

**pfnCallback**

Call back function pointer, SUSI\_WDT\_INT\_CALLBACK is function pointer type, it can set NULL to clear. The type definition just like show below,

```
typedef void (*SUSI_WDT_INT_CALLBACK)(void*);
```

**Context**

Pointer to a user context structure for callback function.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.7 Storage Functions

Access storage information and read / write data to the selected user data area. Developers can use this area to store their own data.

**Table 19 Storage ID**

Id	Description
SUSI_ID_STORAGE_STD	Standard storage device
SUSI_ID_STORAGE_OEM0~1	Other storage devices

#### 3.7.1 SusiStorageGetCaps

```
uint32_t SUSI_API SusiStorageGetCaps(uint32_t Id, uint32_t ItemId, uint32_t *pValue)
```

**Description:**

Reads the capabilities of the current storage implementation from the selected storage interface.

**Parameters:**

**Id**

Selects target device. See **Table 19**.

**ItemId**

Selects target capability. See **Table 20**.

**pValue**

Pointer to a buffer that receives the target capability.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pValue==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device not support	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

**Table 20 Storage capabilities item Id**

Item Id	Description
SUSI_ID_STORAGE_TOTAL_SIZE	Get storage total size in bytes
SUSI_ID_STORAGE_BLOCK_SIZE	Get storage block size in bytes

SUSI_ID_STORAGE_LOCK_STATUS	Get storage lock status. See <b>Table 21</b> .
SUSI_ID_STORAGE_PSW_MAX_LEN	Get maximum length in byte of storage lock key

**Table 21 Storage Lock Status**

Lock Status	Description
SUSI_STORAGE_STATUS_LOCK	Storage is lock
SUSI_STORAGE_STATUS_UNLOCK	Storage is unlock

### 3.7.2 SusiStorageAreaRead

```
uint32_t SUSI_API SusiStorageAreaRead(uint32_t Id, uint32_t Offset, uint8_t *pBuffer,
uint32_t BufLen)
```

**Description:**

Reads data from the selected user data area.

**Parameters:**

**Id**

Selects target device. See **Table 19**.

**Offset**

Storage area start address offset in bytes.

**pBuffer**

Size in bytes of the buffer pointed to by the pBuffer parameter.

**BufLen**

Size in bytes of the information read to the buffer pointed to by the pBuffer parameter.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBuffer==NULL    BufLen==0	SUSI_STATUS_INVALID_PARAMETER
Offset+BufLen>TotalSize	SUSI_STATUS_INVALID_BLOCK_LENGTH
Unknown Id	SUSI_STATUS_UNSUPPORTED
Read error	SUSI_STATUS_READ_ERROR
Success	SUSI_STATUS_SUCCESS

### 3.7.3 SusiStorageAreaWrite

```
uint32_t SUSI_API SusiStorageAreaWrite(uint32_t Id, uint32_t Offset, uint8_t *pBuffer,
```

uint32\_t BufLen)

**Description:**

Writes data to the selected user data area.

**Parameters:**

**Id**

Selects target device. See **Table 19**.

**Offset**

Storage area start address offset in bytes.

**pBuffer**

Size in bytes of the buffer pointed to by the pBuffer parameter.

**BufLen**

Size in bytes of the information read to the buffer pointed to by the pBuffer parameter.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBuffer==NULL    BufLen==0	SUSI_STATUS_INVALID_PARAMETER
Offset+BufLen>TotalSize	SUSI_STATUS_INVALID_BLOCK_LENGTH
Unknown Id	SUSI_STATUS_UNSUPPORTED
Write error	SUSI_STATUS_WRITE_ERROR
Success	SUSI_STATUS_SUCCESS

**3.7.4 SusiStorageAreaSetLock**

uint32\_t SUSI\_API SusiStorageAreaSetLock(uint32\_t Id, uint8\_t \*pBuffer, uint32\_t BufLen)

**Description:**

Lock a storage area for write protect.

**Parameters:**

**Id**

Selects target device. See **Table 19**.

**pBuffer**

Lock of key buffer.

**BufLen**

Number of key buffer

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBuffer==NULL    BufLen==0	SUSI_STATUS_INVALID_PARAMETER
Lock error	SUSI_STATUS_WRITE_ERROR
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device not support	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.7.5 SusiStorageAreaSetUnlock

```
uint32_t SUSI_API SusiStorageAreaSetUnlock(uint32_t Id, uint8_t *pBuffer,
uint32_t BufLen)
```

**Description:**

Unlock a storage area for write protect.

**Parameters:**

**Id**

Selects target device. See **Table 19**.

**pBuffer**

Unlock of key buffer.

**BufLen**

Number of key buffer

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pBuffer==NULL    BufLen==0	SUSI_STATUS_INVALID_PARAMETER
Unlock error	SUSI_STATUS_WRITE_ERROR
Unknown Id	SUSI_STATUS_UNSUPPORTED
Device not support	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.8 Thermal Protection Functions

The Thermal Protection function call is used to set hardware base thermal monitoring and notify. It takes a pointer to an instance of structure *SusiThermalProtect*, which is defined as follows:

```
typedef struct _SusiThermalProtect{
    uint32_t SourceId;
    uint32_t EventType;
    uint32_t SendEventTemperature;
    uint32_t ClearEventTemperature;
} SusiThermalProtect, *PSusiThermalProtect;
```

**Where:**

**SourceId**

Setting thermal source ID here. See **Table 3**.

**EventType**

This byte can set up a thermal protect event, see **Table 22**. NOT every platform supports all event type.

**SendEventTemperature**

Unit is 0.1 Kelvins. When thermal source goes over this value, SUSI will send event according Event Type.

**ClearEventTemperature**

Unit is 0.1 Kelvins. When thermal source goes below this value and Event is sent, SUSI will clear event according Event Type

**Table 22 Thermal Protection Event Type**

Event Type Name	Description	Value
SUSI_THERMAL_EVENT_SHUTDOWN	Shutdown event	0x00
SUSI_THERMAL_EVENT_THROTTLE	Throttle event	0x01
SUSI_THERMAL_EVENT_POWEROFF	Power off event	0x02
SUSI_THERMAL_EVENT_NONE	No event	0xFF

**Table 23 Thermal Protection ID**

Id	Description
SUSI_ID_THERMAL_PROTECT_1	Thermal protection zone 1
SUSI_ID_THERMAL_PROTECT_2	Thermal protection zone 2
SUSI_ID_THERMAL_PROTECT_3	Thermal protection zone 3
SUSI_ID_THERMAL_PROTECT_4	Thermal protection zone 4



### 3.8.1 SusiThermalProtectionGetCaps

```
uint32_t SUSI_API SusiThermalProtectionGetCaps(uint32_t Id, uint32_t ItemId,
uint32_t *pValue)
```

**Description:**

Gets Thermal Protection capabilities.

**Parameters:**

**Id**

Selects target device. See **Table 23**.

**ItemId**

Selects target capability. See **Table 24**. This parameter can also input temperature ID (**Table 3**) to get is it supports in *SusiThermalProtection* function.

**pValue**

Pointer to a buffer that receives the target capability.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pValue==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id or ItemId	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

**Table 24 Thermal Protection capabilities item Id**

Item Id	Description
SUSI_ID_TP_EVENT_SUPPORT_FLAGS	Event support flags (See <b>Table 25</b> )
SUSI_ID_TP_EVENT_TRIGGER_MAXIMUM	The maximum temperature to trigger event
SUSI_ID_TP_EVENT_TRIGGER_MINIMUM	The minimum temperature to trigger event
SUSI_ID_TP_EVENT_CLEAR_MAXIMUM	The maximum temperature to clear event
SUSI_ID_TP_EVENT_CLEAR_MINIMUM	The minimum temperature to clear event

**Table 25 Thermal Protection Support Flags**

Flag Name	Description	Value
SUSI_THERMAL_FLAG_SUPPORT_SHUTDOWN	Support shutdown event	0x01
SUSI_THERMAL_FLAG_SUPPORT_THROTTLE	Support throttle event	0x02
SUSI_THERMAL_FLAG_SUPPORT_POWEROFF	Support power off event	0x04

### 3.8.2 SusiThermalProtectionSetConfig

```
uint32_t SUSI_API SusiThermalProtectionSetConfig(uint32_t Id,
SusiThermalProtect *pConfig)
```

**Description:**

Set Thermal Protection configuration.

**Parameters:**

**Id**

Selects target device. See **Table 23**.

**pConfig**

A data package for thermal protection.

**Return Status Code:**

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pConfig==NULL	SUSI_STATUS_INVALID_PARAMETER
Config invalid	SUSI_STATUS_INVALID_PARAMETER
Source Id or event type not support	SUSI_STATUS_UNSUPPORTED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

### 3.8.3 SusiThermalProtectionGetConfig

```
uint32_t SUSI_API SusiThermalProtectionGetConfig(uint32_t Id,
SusiThermalProtect *pConfig)
```

**Description:**

Get Thermal Protection configuration.

**Parameters:**

**Id**

Selects target device. See **Table 23**.

**pConfig**

A data package for thermal protection.

**Return Status Code:**

<b>Condition</b>	<b>Return Value</b>
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pConfig==NULL	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS