

Software API Library

User's Manual

V1.3

Pro-face

Pro-face

Introduction

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

Pro-face 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 Pro-face platforms.

Pro-face Software API plays the role of catalyst between developer and solution, and makes Pro-face embedded platforms easier and simpler to adopt and operate with customer applications.

\\Program Files\\Pro-face\\SDK\\SUSI4\\lib\\x86\\Susi4.dll

\\Program Files\\Pro-face\\SDK\\SUSI4\\lib\\x86\\Susi4.lib

\\Program Files\\Pro-face\\SDK\\SUSI4\\include\\OsDeclarations.h

\\Program Files\\Pro-face\\SDK\\SUSI4\\include\\Susi4.h

Software API Functions

Initialization Functions

1. SusiLibInitialize

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

Description:

General initialization of the SUSI API. Prior to calling any SUSI API function the library needs to be initialized by calling this function. The status code for all SUSI 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

Library initial fail	SUSI_STATUS_NOT_INITIALIZED
Success	SUSI_STATUS_SUCCESS

2. SusiLibUninitialize

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

Description:

General function to uninitialized the SUSI 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:

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Success	SUSI_STATUS_SUCCESS

GPIO Functions (Support only N2600 model)

Programmable GPIO allows developers to dynamically set the GPIO input or output status

Table 1 GPIO ID

Id	Description
SUSI_ID_GPIO(X)	X is GPIO pin number, definition as below: #define SUSI_ID_GPIO(x) (0x0000 x) This ID control single pin only.
SUSI_ID_GPIO_BANK(Y)	Y is GPIO bank number, definition as below: #define SUSI_ID_GPIO_BANK(Y) (0x10000 Y) This ID control maximum 32 pins per bank. N2600 support only BANK 0(0x10000).

1. SusiGPIOGetCaps

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

Description:

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

Parameters:

- Id**
Selects target device. See **Table 1**.
- ItemId**
Selects target capability. See **Table 2**.
- 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
Success	SUSI_STATUS_SUCCESS

Table 2 GPIO capabilities item Id

Item Id	Description
SUSI_ID_GPIO_INPUT_SUPPORT	Get GPIO input support state
SUSI_ID_GPIO_OUTPUT_SUPPORT	Get GPIO output support state

2. SusiGPIOGetDirection

```
uint32_t Susi_API SusiGPIOGetDirection(uint32_t Id, uint32_t Bitmask,
uint32_t *pDirection)
```

Description:

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

Parameters:

- Id**
Selects target device. See **Table 1**.
- Bitmask**
Value for a bit mask. Only selected bits are changed, unselected bits remain unchanged. This parameter will be ignored when single pin mode..
- pDirection**
Pointer to a buffer that receives the direction of the selected GPIO ports.

Return Status Code:

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pDirection==NULL	SUSI_STATUS_INVALID_PARAMETER
Bitmask==0 when bank mode	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

3. SusiGPIOSetDirection

```
uint32_t SUSI_API SusiGPIOSetDirection(uint32_t Id, uint32_t Bitmask, uint32_t Direction)
```

Description:

Sets the configuration for the selected GPIO ports.

Parameters:

Id

Selects target device. See **Table 1**.

Bitmask

Value for a bit mask. Only selected bits are changed, unselected bits remain unchanged. This parameter will be ignored when single pin mode.

Direction

Sets the direction of the selected GPIO ports.

Return Status Code:

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Bitmask==0 when bank mode	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

4. SusiGPIOGetLevel

```
uint32_t SUSI_API SusiGPIOGetLevel(uint32_t Id, uint32_t Bitmask, uint32_t *pLevel)
```

Description:

Read level the from GPIO ports.

Parameters:

Id

Selects target device. See **Table 1**.

Bitmask

Value for a bit mask. Only selected bits are changed, unselected bits remain unchanged. This parameter will be ignored when single pin mode.

pLevel

Pointer to a buffer that receives the GPIO level.

Return Status Code:

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
pLevel==NULL	SUSI_STATUS_INVALID_PARAMETER
Bitmask==0 when bank mode	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

5. SusiGPIOSetLevel

```
uint32_t SUSI_API SusiGPIOSetLevel(uint32_t Id, uint32_t Bitmask, uint32_t Level)
```

Description:

Write level to GPIO ports. Depending on the hardware implementation writing multiple GPIO ports with the bit mask option does not guarantee a time synchronous change of the output levels..

Parameters:

Id

Selects target device. See **Table 1**.

Bitmask

Value for a bit mask. Only selected bits are changed, unselected bits remain unchanged. This parameter will be ignored when single pin mode.

Level

Input level of the selected GPIO port.

Return Status Code:

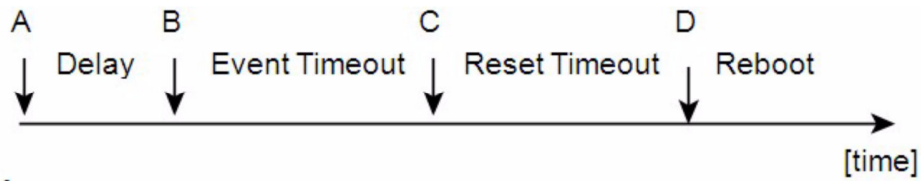
Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Bitmask==0 when bank mode	SUSI_STATUS_INVALID_PARAMETER
Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

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 SUSI 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:

Pro-face



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 3 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

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 3**.
- ItemId**
Selects target capability. See **Table 4**.
- 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 4 Watchdog capabilities item Id

Item Id	Description
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 (Table 5)

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 SUSI API selects the next possible longer timing.

Parameters:

- Id**
Selects target device. See **Table 3**.
- 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 5**.

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 5 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. SusiWDogStop

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

Description:

Stops the operation of the watchdog timer.

Parameters:

Id
Selects target device. See **Table 3**.

Return Status Code:

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

4. SusiWDogTrigger

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

Description:

Trigger the watchdog timer.

Parameters:

Id
Selects target device. See **Table 3**.

Return Status Code:

Condition	Return Value
Library uninitialized	SUSI_STATUS_NOT_INITIALIZED
Unknown Id	SUSI_STATUS_UNSUPPORTED
Success	SUSI_STATUS_SUCCESS

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 3**.

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